



UNIVERSIDADE ESTADUAL PAULISTA  
“JÚLIO DE MESQUITA FILHO”

MARCOS HENRIQUE DE PAULA DIAS DA SILVA

# **Handles – a trajetória de desenvolvimento de um jogo digital para ensino de Matemática**

**Bauru**

**2018**

MARCOS HENRIQUE DE PAULA DIAS DA SILVA

## **Handles – a trajetória de desenvolvimento de um jogo digital para ensino de Matemática**

Dissertação apresentada à Faculdade de Ciências da Universidade Estadual Paulista “Júlio de Mesquita Filho” – UNESP, Campus de Bauru, Programa de Pós-Graduação em Educação para a Ciência, como um dos requisitos para a obtenção do título de Mestre em Educação para a Ciência (Área de Concentração: Ensino de Ciências e Matemática), sob a orientação do Prof. Dr. Wilson Massashiro Yonezawa.

Universidade Estadual Paulista “Júlio de Mesquita Filho” - UNESP  
Faculdade de Ciências  
Programa de Pós-Graduação em Educação para a Ciência

Orientador: Prof. Dr. Wilson Massashiro Yonezawa

Bauru  
2018

Silva, Marcos Henrique de Paula Dias da.

Handles - a trajetória de desenvolvimento de um jogo digital para ensino de Matemática / Marcos Henrique de Paula Dias da Silva, 2018  
182 f.

Orientador: Wilson Massashiro Yonezawa

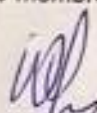
Dissertação (Mestrado)-Universidade Estadual Paulista. Faculdade de Ciências, Bauru, 2018

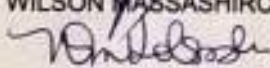
1. Ensino de Matemática. 2. Jogo digital. 3. Game design. I. Universidade Estadual Paulista. Faculdade de Ciências. II. Título.

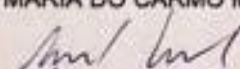
**ATA DA DEFESA PÚBLICA DA DISSERTAÇÃO DE Mestrado DE MARCOS HENRIQUE DE PAULA DIAS DA SILVA, DISCENTE DO PROGRAMA DE PÓS-GRADUAÇÃO EM EDUCAÇÃO PARA A CIÊNCIA, DA FACULDADE DE CIÊNCIAS - CÂMPUS DE BAURU.**

Aos 02 dias do mês de março do ano de 2018, às 14:00 horas, no(a) Sala de Videoconferência da Faculdade de Ciências - UNESP/Bauru, reuniu-se a Comissão Examinadora da Defesa Pública, composta pelos seguintes membros: Prof. Dr. WILSON MASSASHIRO YONEZAWA - Orientador(a) do(a) Departamento de Computação / Faculdade de Ciências - UNESP/Bauru, Profa. Dra. MARIA DO CARMO MONTEIRO KOBAYASHI do(a) Departamento de Educação / Faculdade de Ciências - UNESP/Bauru, Prof. Dr. ANDRÉ KOSCIANSKI do(a) Câmpus Ponta Grossa / Universidade Tecnológica Federal do Paraná - UTFPR, sob a presidência do primeiro, a fim de proceder a arguição pública da DISSERTAÇÃO DE Mestrado de MARCOS HENRIQUE DE PAULA DIAS DA SILVA, intitulada "Handles - um jogo digital para Ensino de Matemática". Após a exposição, o discente foi arguido oralmente pelos membros da Comissão Examinadora, tendo recebido o conceito final: A.P. 10/10 37

----- Nada mais havendo, foi lavrada a presente ata, que após lida e aprovada, foi assinada pelos membros da Comissão Examinadora.

  
Prof. Dr. WILSON MASSASHIRO YONEZAWA

  
Profa. Dra. MARIA DO CARMO MONTEIRO KOBAYASHI

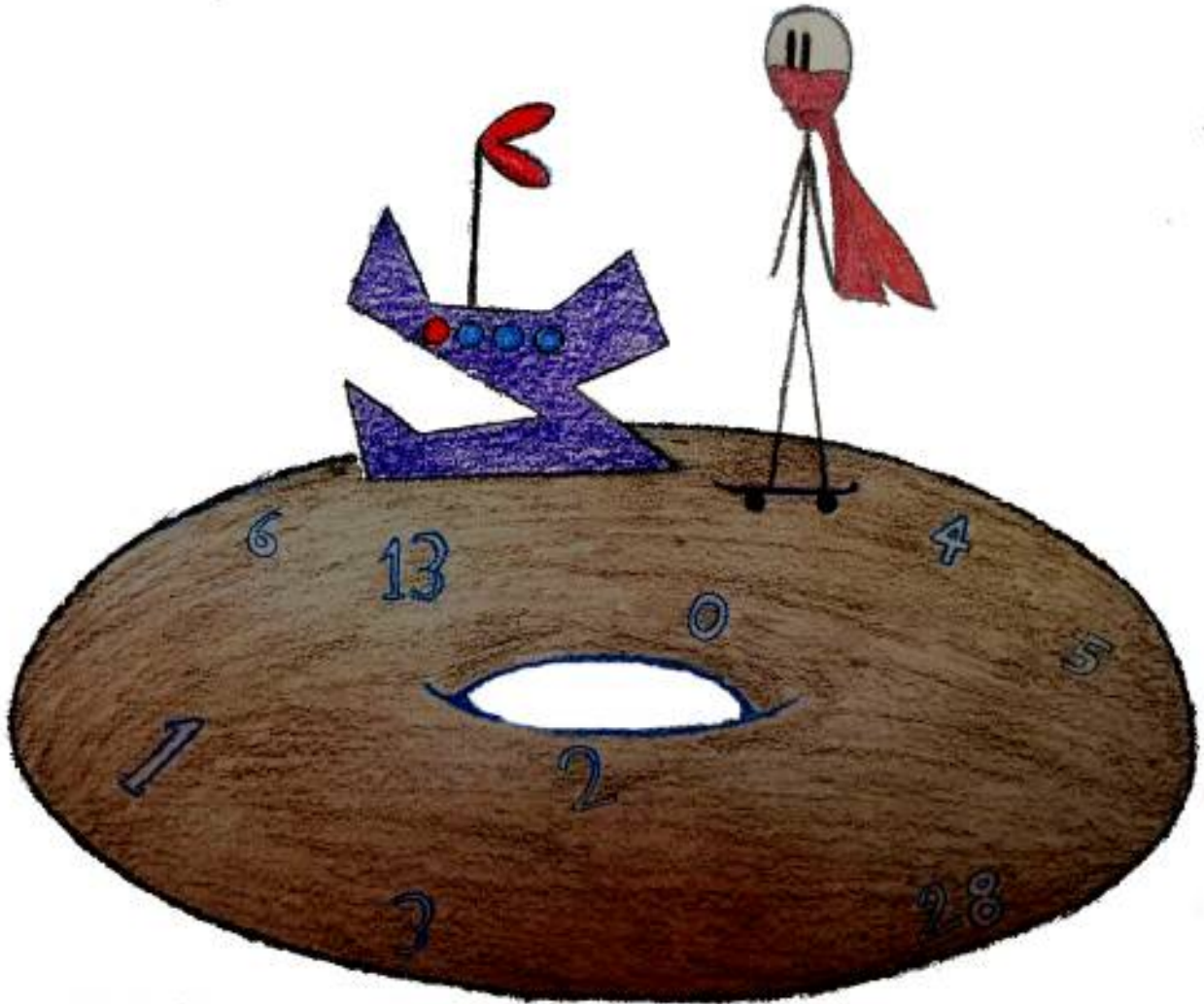
  
Prof. Dr. ANDRÉ KOSCIANSKI

Alteração de título:

Handles - a trajetória de desenvolvimento de um jogo digital para ensino de Matemática.

## Mensagem de motivação:

O essencial é invisível aos olhos<sup>1</sup>.



Referência ao Pequeno Príncipe para o esquema do jogo Handles (fonte própria).

---

<sup>1</sup> Frase do livro O Pequeno Príncipe (SAINT-EXUPÉRY, Antoine de. *O pequeno príncipe*. Rio de Janeiro, Editora Agir, 2009).

SILVA, M. H. P. D. **Handles – a trajetória de desenvolvimento de um jogo digital para ensino de Matemática**. Dissertação (Mestrado em Educação para a Ciência) – Faculdade de Ciências, Universidade Estadual Paulista “Júlio de Mesquita Filho”, Bauru, 2018.

## RESUMO

A experiência em participar de dois grupos pouco associados e perceber que juntos teriam potencial de construir excelentes jogos digitais para ensino de Matemática, foi o motor desta pesquisa. Este trabalho focou nas dificuldades de um licenciado em Matemática, com poucos recursos e baixa experiência em computação, imerso de forma amadora e individual no meio do *game design* digital. Iniciou-se com uma revisão da literatura explicando sobre diversas teorias conceituadas relacionadas com jogos, sendo estas, abordagens que variavam desde a Psicologia, o Jornalismo, a Matemática, o Ensino, o *Game design* entre outras. Então, apresentamos o processo de documentação utilizado para registro das experiências, recursos e caracterização do sujeito, como seus computadores, uma breve biografia, a estruturação do seu diário, os objetivos dos questionários e as versões do jogo digital Handles produzido no período de estudo. Por fim, reunimos todos os desafios de Handles dentro de problemas Matemáticos generalizados, e contextualizados dentro da instância de jogo com soluções sugeridas. Também analisamos os dados de 27 crianças do Ensino Fundamental I e 142 estudantes do Ensino Médio Técnico, que jogaram, e em ambos os grupos houve uma visível aprendizagem dos conceitos Matemáticos envolvidos. Então associamos a experiência narrada no diário com as diversas teorias relacionadas a jogos, revisadas anteriormente, indicando os pontos onde cada uma melhor contribui nesta prática. Encerra-se o texto com a construção de um percurso, baseado neste estudo, por onde licenciados em Matemática em condições semelhantes ao nosso sujeito, possam criar com maior facilidade seus próprios jogos digitais para ensino de Matemática em um nível de sofisticação similar ao Handles.

**Palavras-chaves:** Ensino de Matemática. Jogo digital. Game design.

SILVA, M. H. P. D. **Handles - the design trajectory of a digital game for teaching Mathematics.** Dissertation (Master of Education Science) – Faculty of Sciences, Universidade Estadual Paulista “Júlio de Mesquita Filho”, Bauru, 2018.

## ABSTRACT

The experience of participating in two little associated groups and realizing that, combined, they would have the potential to construct excellent digital games for teaching mathematics was the driver of this research. This study focused on the difficulties of a mathematical education graduate, with few resources and little computing experience, engaged in an amateur and individual fashion in the digital game design milieu. It began with a review of the literature explaining the many conceptual theories related to games, with approaches including psychology, journalism, mathematics, education, and game design. We then present the documentation process used to record the subject's experiences, resources and characterization, such as his computer, a brief biography, the structure of his diary, the questionnaire goals, and the versions of the digital game Handles produced during the study period. Finally, we combine all the challenges related to Handles within the generalized math problems, put into context within the game's instance with the suggested solutions. We also analyzed the data of 27 Elementary School I children and 142 Technical High School students who played the game, with each group showing a visible learning of the math concepts involved. We then associate the experience presented in the diary with the different theories related to games, which have been previously revised, indicating at which points each theory best contributes to this practice. The text ends with the construction of a path, based on this study, according to which mathematics education graduates, in conditions similar to our subject's, may more easily create their own digital games for teaching mathematics on a sophistication level similar to Handles.

**Key-words:** Teaching math. Digital game. Game design.

## LISTA DE ILUSTRAÇÕES

<b>Figura 1</b> -	Jogo Exploração Halloween (SILVA, CÂMARA, 2013)	15
<b>Figura 2</b> -	Jogo Math Madness	16
<b>Figura 3</b> -	Jogo Geometric Quiz	17
<b>Figura 4</b> -	Esquema de <i>puzzle dinâmico</i>	18
<b>Figura 5</b> -	Esquema do <i>designer de problemas</i>	19
<b>Figura 6</b> -	Esquema dos subníveis de jogos até <i>puzzles dinâmicos</i>	21
<b>Figura 7</b> -	Jogo dos 15 (PETKOVIC, 2009, p. 8)	23
<b>Figura 8</b> -	Jogos presentes nos livros, distribuídos por Ano do Ensino Básico	25
<b>Figura 9</b> -	Esquema de objetivos da pesquisa	33
<b>Figura 10</b> -	Tempos necessários para soluções (GAREY, JOHNSON, 1979, p. 7, traduzido)	36
<b>Figura 11</b> -	Exemplos de problemas com funções soluções das ordens $n$ , $n^2$ , NP	37
<b>Figura 12</b> -	Situação de antecipação no Pedra-papel-tesoura (No Game No Life, episódio 2)	39
<b>Figura 13</b> -	Modelo de pesquisa iterativo (ZIMMERMAN, 2003, p. 177, traduzido)	41
<b>Figura 14</b> -	Modelo MDA (adaptado de HUNICKE, LEBLANC, ZUBEK, 2004)	45
<b>Figura 15</b> -	Modelo simplificado do fluxo (NAKAMURA, CSIKSZENTMIHALYI, 2009, 50 traduzido)	50
<b>Figura 16</b> -	Modelo geral do fluxo (NAKAMURA, CSIKSZENTMIHALYI, 2009, traduzido)	51
<b>Figura 17</b> -	Jogo Revolta da Cabanagem (portaldoprofessor.mec.gov.br)	52
<b>Figura 18</b> -	Jogo de arremessar o lixo orgânico/Tetris (Adaptado de KOSTER, 2014)	54
<b>Figura 19</b> -	Computadores do <i>designer de problemas</i>	62
<b>Figura 20</b> -	Modelo de produção para jogos digitais (Adaptado de CHANDLER, 2012)	63
<b>Figura 21</b> -	Jogo Pega e Larga	64
<b>Figura 22</b> -	Jogo Handles	64
<b>Figura 23</b> -	Jogo Handles 2.0	65
<b>Figura 24</b> -	Jogo Handles in Scratch 2.0	66



<b>Figura 25</b> - Jogo Saúvas in Scratch	66
<b>Figura 26</b> - Tela dos jogos dentro do site Scratch	67
<b>Figura 27</b> - Situação do problema 1	71
<b>Figura 28</b> - Situação do problema 2	71
<b>Figura 29</b> - Situação do problema 3	72
<b>Figura 30</b> - Situação do problema 4	73
<b>Figura 31</b> - Situação do problema 5	73
<b>Figura 32</b> - Situação do problema 6	74
<b>Figura 33</b> - Situação do problema 7	75
<b>Figura 34</b> - Situação do problema 8	76
<b>Figura 35</b> - Situação do problema 9	76
<b>Figura 36</b> - Situação do problema 10	77
<b>Figura 37</b> - Frequência de notas no Ensino Fundamental I	79
<b>Figura 38</b> - Frequência de notas no Ensino Médio Técnico	79
<b>Figura 39</b> - Modelo iterativo deste <i>puzzle dinâmico</i> (Adaptado de ZIMMERMAN, 2003)	82
<b>Figura 40</b> - Modelo MDA deste <i>puzzle dinâmico</i> (Adaptado de HUNICKE, LEBLANC, ZUBEK, 2004)	83
<b>Figura 41</b> - Exemplo de interação com ferramentas inteligentes do <i>puzzle dinâmico</i>	84
<b>Figura 42</b> - Esquema de trajetória das referências analisadas	93

## **LISTA DE TABELAS**

**TABELA 1** Distribuição das etapas de produção dos jogos digitais 63

## LISTA DE QUADROS

<b>QUADRO 1</b>	Autores e obras da revisão da literatura	34
-----------------	--	----

## Sumário

1. Introdução.....	13
2. Fundamentação teórica.....	21
2.1. Jogo.....	21
2.2. Jogo digital.....	22
2.3. Jogo para ensino da Matemática.....	23
2.4. Jogo educativo.....	26
2.5. <i>Game design</i> e <i>game designer</i> .....	28
3. Metodologia.....	30
4. Revisão da literatura.....	34
4.1. Jogos como problemas.....	34
4.2. Modelo de pesquisa em <i>game design</i> .....	40
4.3. Aprendizagem nos jogos.....	46
4.4. Guia para <i>game designers</i> .....	55
4.5. Gerenciamento de projetos em <i>game design</i> .....	57
4.6. Fundamentos do <i>game design</i> .....	57
5. Documentos.....	61
5.1. Designer de problemas.....	61
5.2. Diário de <i>game design</i> .....	62
5.2. Jogos digitais.....	63
5.4. Questionários.....	67
6. Educação Matemática.....	70
6.1. Matemática implícita.....	70
6.2. Aplicação e resultados.....	77
6.3. Teoria e prática.....	80
7. Considerações finais.....	88
Referências.....	96
Apêndices.....	100
1. Diário de <i>game design</i> .....	100
Pega e Larga.....	107
Handles.....	115
Handles 2.0.....	127
Handles in Scratch 2.0.....	147
Saúvas in Scratch.....	166
2. Questionário anterior ao jogo.....	172
3. Questionário posterior ao jogo.....	173
4. Artigo não-publicado / releitura do método de Fellows (1996).....	174

## 1. Introdução

Durante minha graduação em Matemática convivi por anos com dois grupos acadêmicos. Os bolsistas do PIBID (Programa Institucional de Bolsas de Iniciação à Docência) de Matemática e os membros do FoG (*Fellowship of the Game*), um grupo sobre jogos digitais.

Parte das ações exigidas pelo PIBID era levar semanalmente jogos que trabalhassem a Matemática junto aos alunos do Ensino Fundamental II de uma escola estadual de período integral. O procedimento recomendado para isto envolvia visitar o Laboratório de Educação Matemática, escolher jogos prontos e aplicá-los na escola. Um ponto que considerava negativo neste método era o tamanho e a simplicidade do repertório disponível no laboratório. Em resposta a esta situação, comecei de forma semanal a desenvolver e adaptar novos jogos para o ensino de Matemática.

Entre as ações exigidas no FoG, havia o estudo e desenvolvimento de recursos na área de jogos digitais. O posicionamento do grupo, composto quase que inteiramente por graduandos em Ciências da Computação, costumava ser bastante prático, as preocupações principais dos projetos giravam em torno da construção do *software*, arte digital e som. Por não acompanhar o grupo em recursos e conhecimentos, além de ajudar nos projetos de jogos, também me responsabilizava por elaborar e ministrar para crianças, cursos de *Scratch 2.0*<sup>2</sup> voltados para construção de jogos digitais.

Nesta época fazia somente um uso superficial das teorias do *game design* (este termo pode ser entendido como o processo de construir um jogo, será discutido novamente na seção 2.5). Todo o processo real de desenvolvimento e construção dos jogos, fossem digitais ou não, ocorria de forma intuitiva. Dentro de ambos os grupos, aspectos teóricos não recebiam atenção, exceto quando a produção acadêmica estava envolvida.

Contudo, hoje considero até mais negativo do que a falta de fundamentação, a ausência de comunicação e de trabalhos em parceria com os dois grupos. No PIBID o interesse estava na produção de jogos para o ensino de Matemática. No FoG o objetivo eram os jogos digitais. Porém, quando surgia um dilema envolvendo jogos digitais para ensino de Matemática, era oportunidade para ambos os grupos trabalharem juntos, o que não ocorria. Cada qual construía o projeto a sua própria maneira. Os bolsistas do PIBID tendo como base do currículo, diversas disciplinas da licenciatura e poucas da computação, produziam jogos digitais onde a Matemática aparecia na forma de questões aritméticas objetivas, os cenários eram somente decorativos e havia pouca interação com o usuário, que geralmente podia clicar em uma das alternativas ou digitar a resposta. De modo análogo, os membros do FoG tendo como base do currículo, diversas disciplinas da

---

<sup>2</sup> Scratch 2.0 é uma linguagem de programação com uma interface gráfica que permite construir seus códigos a partir do encaixe de blocos de comandos.

computação e poucas (senão nenhuma) ligadas à licenciatura, produziam jogos digitais para o ensino da Matemática com estruturas similares aqueles criados pelos bolsistas do PIBID.

Assim, em um ambiente no qual o ensino da Matemática e a computação poderiam trabalhar juntos, a ausência de comunicação afetava seus produtos. De modo similar, temos uma crítica feita por Michael R. Fellows (1996) na *Computer Game Developers Conference (CGDC)* de 1994:

I complained about the dismal state of affairs in computer games for mathematics education, much of which is little more than arithmetic drill decorated with “cute” animation. Boring! And what an ironic situation, given the wealth of mathematical topics associated with computing and nowhere to be seen in these products *based on computing*. (FELLOWS, 1996, p. 3)

Justificando e concordando com a crítica acima, podemos analisar estes jogos a partir da incerteza provocada por seus desafios, do aproveitamento de seus recursos computacionais para auxiliar o jogador e na interatividade disponível durante a resolução dos problemas matemáticos.

Costikyan (2013), interpreta o jogo como um subgrupo do ato de jogar. Ou seja, o jogo por si só não existe, mas qualquer situação que dê ao sujeito a sensação de jogar, passa a ser um jogo. Por exemplo arremessar uma bola em uma piscina é uma simples ação, não um jogo. Mas quando em vez da piscina, o alvo é um balde, isto gera uma incerteza de sucesso e a pessoa sente-se jogando com aquele arremesso um jogo de destreza, que somente então passou a existir. Nessa perspectiva, a incerteza do jogador é o motor que movimenta o jogo (este tópico será discutido com mais detalhes no capítulo 4). Logo, nos jogos digitais para o ensino de Matemática comumente produzidos pelos bolsistas do PIBID e membros do FoG, os jogadores podem lidar somente com a “incerteza de solução”. A dúvida permanece sobre a solução correta ter sido encontrada. No caso, a resolução objetiva das questões aritméticas. Porém, se os jogadores sabem realizar os cálculos aritméticos do jogo, esta incerteza deixa de existir e o jogo torna-se somente um exercício de repetição.

Gee (2005) considera que os bons jogos compartilham de princípios da aprendizagem voltados a ensinar os jogadores como jogar o próprio jogo. Um destes princípios envolve a distribuição de conhecimentos entre jogador e jogo, dispondo também de ferramentas “inteligentes” que os auxiliem na resolução de problemas (este tópico será discutido com mais detalhes no capítulo 4). Porém, nos jogos digitais para ensino de Matemática de ambos os grupos (PIBID e FoG), o cenário decorativo concentrava o conhecimento somente no jogador. Assim, não apresentava este princípio de aprendizagem que intermedeia o desafio matemático e sua respectiva resolução.

Nakamura e Csikszentmihalyi (2009) consideram o *feedback* imediato sobre o progresso realizado, como um requisito (necessário, mas não suficiente) para quem realiza uma atividade entrar em fluxo (também referido como estado de fluxo), um estado no qual o sujeito encontra-se imerso na atividade e exerce o máximo de suas capacidades (este tópico será discutido com mais detalhes no capítulo 4). Fornecer condições para que os jogadores entrem em estado de fluxo, é considerado por vários *game designers* (SCHUYTEMA, 2008, ROGERS, 2012, SALEN, ZIMMERMAN, 2012a) como um aspecto presente nos jogos com boa interação dos jogadores. Contudo, os jogos digitais para ensino de Matemática de ambos os grupos (PIBID e FoG) interagem com o jogador de forma direta e mental (o jogador resolve mentalmente emitindo uma resposta objetiva e absoluta), não havendo canais de *feedback* com jogador sobre seu progresso na resolução do desafio. Uma condição necessária para entrar em fluxo.

A fim de ilustrarmos estes jogos digitais para ensino de Matemática, tomaremos como exemplo um de própria autoria. E para representar a facilidade com que estes jogos são encontrados, apresentaremos também dois dos primeiros resultados na busca por imagens do site Google com os descritores “*game*”, “*online*” e “*math*”.

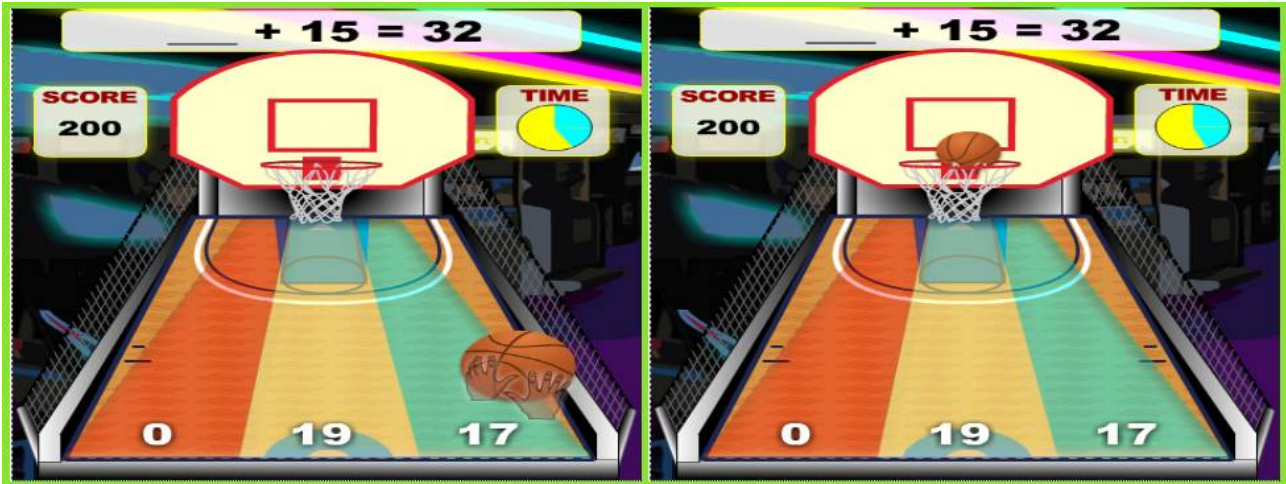
No jogo *Exploração Halloween* (SILVA, CÂMARA, 2013), o jogador precisa chegar até uma das 3 casas avançando por um dos caminhos de pegadas. Com exceção do primeiro par de pegadas em cada caminho, os demais possuem senhas para serem acessados. Cada par de pegadas, quando aberto apresenta ao jogador um problema matemático contextualizado e cuja resposta é a senha para o próximo par de pegadas naquele mesmo caminho.



**Figura 1:** Jogo *Exploração Halloween* (fonte própria).

No jogo *Math Madness* (<<http://www.fuelthebrain.com/games/math-madness/>> acesso em 12/01/2018) o jogador escolhe uma opção de desafios (subtrações, adições, multiplicações) em uma intensidade (fácil, difícil). Em seguida precisa selecionar as alternativas das contas que aparecem

acima da cesta de basquete. Se correta, ganha pontos e avança para a próxima conta, senão permanece na mesma e perde alguns pontos. O jogo termina quando o tempo localizado na lateral da cesta se esgota.



**Figura 2:** Jogo *Math Madness*.

No jogo *Geometric Quiz* (<<http://www.mathgametime.com/games/geometry-quiz>> acesso em 12/01/2018) o jogador precisa dentro de um tempo estipulado digitar em um teclado numérico, localizado na lateral direita da tela, a solução dos problemas de geometria apresentados. Contudo a representação dos problemas é errônea. Na questão 4 (figura 3 à esquerda) por exemplo, as medidas aproximadas dos ângulos  $57^\circ$ ,  $61^\circ$  e  $62^\circ$  seriam respectivamente  $49^\circ$ ,  $73^\circ$  e  $58^\circ$ . Assim como na questão 5 (figura 3 à direita) a medida dos ângulos  $68^\circ$ ,  $56^\circ$ ,  $32^\circ$  e  $92^\circ$  seriam respectivamente  $65^\circ$ ,  $47^\circ$ ,  $61^\circ$  e  $50^\circ$ . Interessante neste caso, é notar quão discrepantes são as diferenças, pois na questão 5 por exemplo, a solução algébrica é superior a um ângulo reto. Contudo, visualmente podemos identificar na representação geométrica, que a solução deveria ser um ângulo agudo.



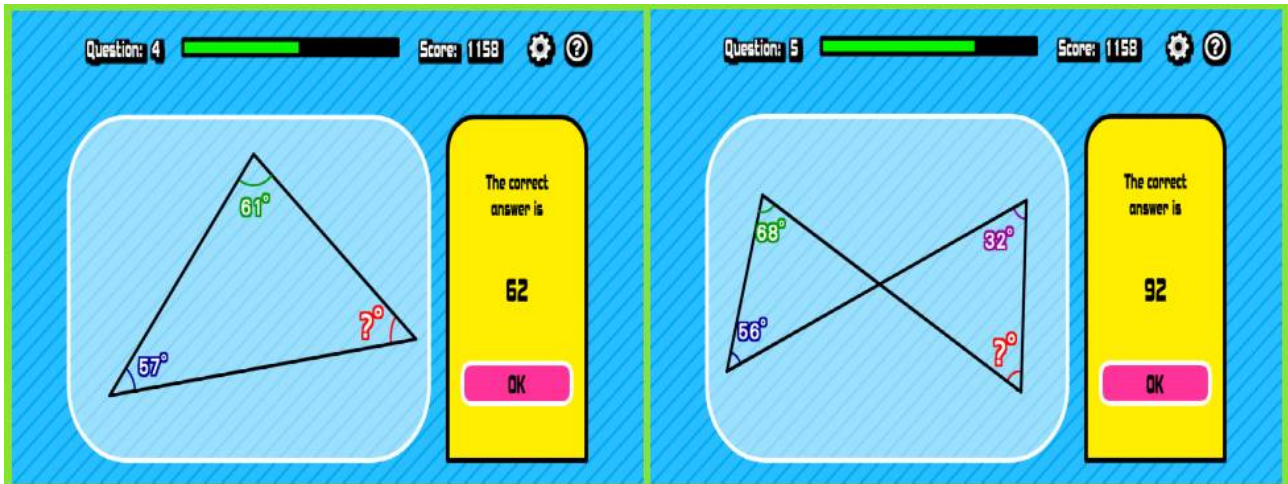


Figura 3: Jogo *Geometric Quiz*.

Assim, nestes três jogos o jogador lida principalmente com uma incerteza de solução. No jogo *Math Madness* e *Geometric Quiz* temos também uma incerteza de performance, relacionada ao tempo disponível para realizar os respectivos cálculos mentais. Nos três produtos não há distribuição de conhecimentos e ferramentas do jogo para jogador. Embora *Geometric Quiz* pareça distribuir conhecimentos com o jogador mediante uma representação geométrica, suas soluções se embasam somente em uma resolução algébrica ao problema. Em todos os casos, seu modo de interação com o usuário é direto e mental, sem *feedbacks* sobre o progresso da resolução.

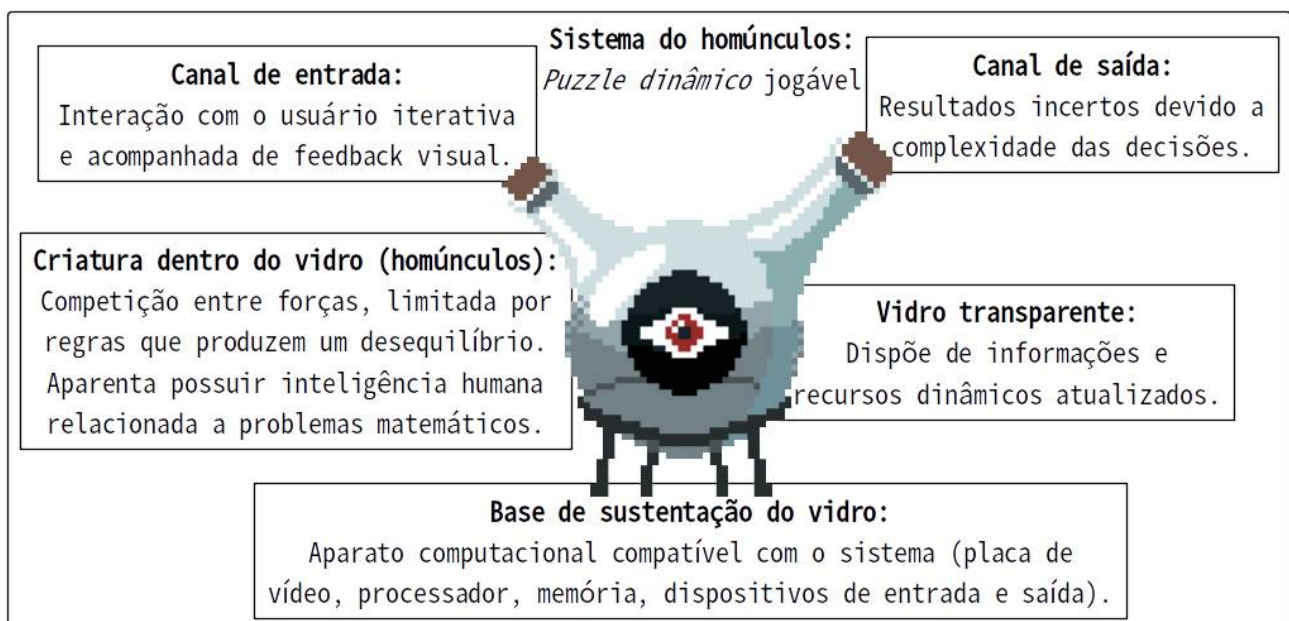
Como resposta a nossa própria crítica justificada e exemplificada nos parágrafos anteriores, e refletindo na experiência em ambos os grupos (PIBID e FoG). Um aspecto positivo seria se os jogos desenvolvidos:

- a) Lidassem com incertezas que Costikyan (2013) classifica como “complexidade analítica” (que envolvem a tomada de decisões dentro de inúmeros caminhos com várias soluções equivalentes levando o jogador a incerteza sobre a decisão certa ter sido tomada).
- b) Compartilhassem do princípio de aprendizagem classificado por Gee (2005) como “conhecimento distribuído e ferramentas inteligentes” (que envolvem o jogo dispor de informações e recursos dinâmicos atualizados que podem ser usados pelo jogador na resolução de problemas).
- c) Interagissem com o usuário de modo iterativo e acompanhado de um *feedback* sobre o progresso atual na resolução do desafio, um requisito considerado por Nakamura e Csikszentmihalyi (2009) para que uma atividade permita o indivíduo entrar em fluxo.

Apenas para facilitar a continuidade do texto, nos referiremos a jogos digitais para o ensino de Matemática com estas três características como *puzzles dinâmicos*.

Apoiando-se na visão de Brougère (1998) de que o jogo é natural a sociedade humana e aprendido nos grupos sociais onde as pessoas se encontram. Temos na direção oposta o papel do *game designer* como um profissional que desenvolve jogos (este tema será discutido melhor na seção 2.5) sem a necessidade de uma constante e natural socialização humana, criando-os em um ambiente artificial a forma “natural” da qual os jogos nascem. Logo, podemos imaginar duas espécies de jogo, aquele que surge impulsionado por condições naturais da própria sociedade e aquele desenvolvido artificialmente a partir de métricas e requisitos preestabelecidos. Assim, um *puzzle dinâmico* como o proposto, não é natural ao seu meio (grupos PIBID e FoG), senão já os fariam.

Neste contexto, temos o *puzzle dinâmico* como uma criatura artificial complexa com funcionamento autônomo reconhecido, que precisa demonstrar uma inteligência humana, cuja existência ocorre somente em um ambiente de suporte devidamente preparado e ajustado, com canais reguladores para entradas e saídas de informações. Características que se assemelham as criaturas humanoides das quais os cientistas do período do renascimento tentavam construir, homúnculos (ALMEIDA, MAGALHÃES, 2010). Dessa forma, para melhor representar o que trataremos por um *puzzle dinâmico*, temos a seguir uma ilustração dele em analogia a um homúnculo dentro do vidro. A definição de jogo digital para ensino de Matemática, a qual fomenta parte das explicações desta ilustração será discutida no capítulo 2, Fundamentação teórica.

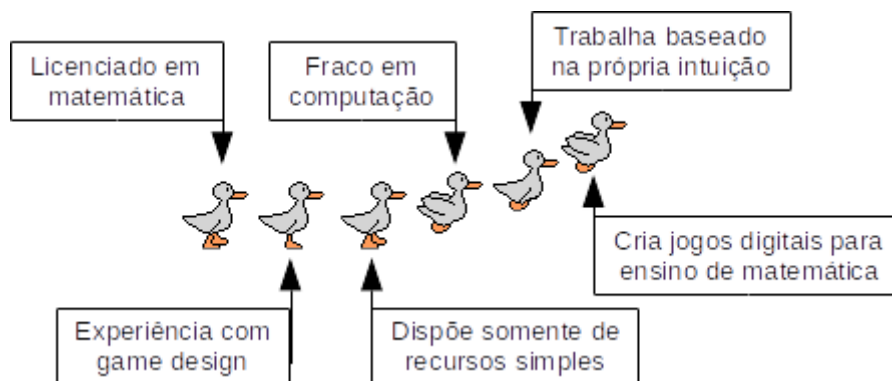


**Figura 4:** Esquema de *puzzle dinâmico* (fonte própria).

Considerando que a prática e as discussões nestes grupos pouco utilizavam a literatura do *game design*, podemos encarar como um benefício de cunho prático a grupos similares, se o

progresso na construção de um *puzzle dinâmico* fosse conectada com as principais teorias da área. Assim, licenciados em Matemática, experientes em desenvolver jogos, com recursos simplórios e poucos conhecimentos em computação, podem se valer destas associações no uso apropriado e direcionado das pesquisas relacionadas ao *game design*. Apenas para facilitar a continuidade do texto, nos referiremos a pessoas com estas características como *designers de problemas*.

De forma a melhorar a caracterização deste sujeito, considero adequado a analogia dele como um pato. Para entender esta comparação, imaginemos se os profissionais fossem aves especializadas em algumas funções. Os estatísticos por exemplo, seriam aves que correm rápido (como avestruzes) ou os cientistas da computação aves que nadam bem (como pinguins). Nesse contexto, o pato (*designer de problemas*) é uma ave que corre e nada de forma satisfatória, mas consegue realizar ambas as ações. Nessa mesma linha, podemos imaginar o *game designer* profissional (indivíduo que cria jogos) como uma ave que voa, e embora o pato não seja munido da mesma estrutura de uma águia, demonstra capacidade de voar, ou seja, consegue criar jogos digitais para ensino de Matemática em nível suficiente para uso.



**Figura 5:** Esquema do *designer de problemas* (fonte própria).

Em cima do que foi apresentado, definimos a seguinte questão de pesquisa: como ocorre a prática de um *designer de problemas* no desenvolvimento de um *puzzle dinâmico*? Entre os benefícios de uma pesquisa como esta, temos:

- a) A descrição do *designer de problemas* cobre muitos licenciados em Matemática, que podem se valer das associações construídas entre a teoria do *game design* e a prática em um *puzzle dinâmico*.
- b) O desenvolvimento por *designers de problemas* ou *game designers* de novos *puzzles dinâmicos*.
- c) Mais tópicos da Matemática, comum ao currículo, explorados através de *puzzles dinâmicos*.

Visando responder ao problema desta pesquisa determinamos como objetivo geral estudar o processo de um *designer de problemas* durante a construção de um *puzzle matemático* eficiente.

Para satisfazer este objetivo geral, definimos os seguintes objetivos específicos que serão explorados no corpo do texto:

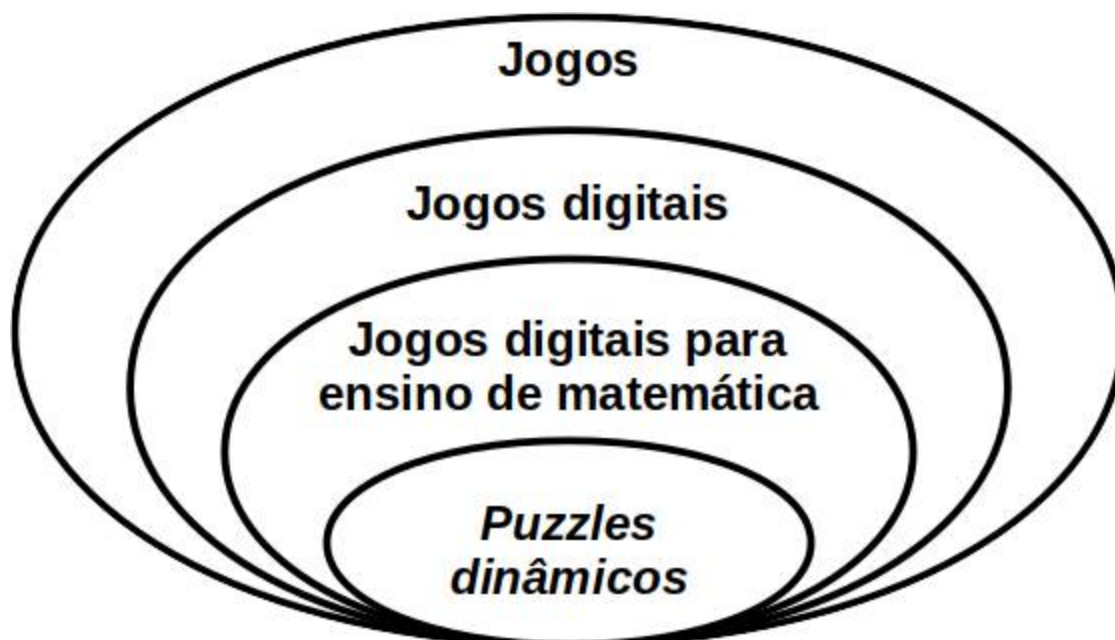
- a) Revisar a literatura acerca das teorias de *game design* e dos guias para *game designers*.
- b) Documentar o processo de construção de um *puzzle dinâmico* por um *designer de problemas*.
- c) Conectar com a educação Matemática a construção do *puzzle dinâmico* e a literatura revisada.

Segundo Bailey (2008), caso o pesquisador não possua hipóteses para começar, ele pode tentar explicar suas descobertas com base em alguma teoria que resulte em perguntas para futuras pesquisas. Assim, apoiado nesta perspectiva, tratamos aqui de um estudo com caráter exploratório voltado para formulação de hipóteses sobre o tema as dificuldades e caminhos da prática de um *designer de problemas* na construção de *puzzles dinâmicos*.

Este texto se encontra distribuído em 7 capítulos da seguinte maneira. O Capítulo 1 expõe nosso tema, aborda quais pensamentos nos trouxeram até o problema de pesquisa, os objetivos e benefícios que esta pesquisa pode proporcionar. O Capítulo 2 apresenta os termos fundamentais para a compreensão do nosso texto como jogo, jogo digital, jogo digital para ensino de Matemática, as diferenças entre jogo e jogo educativo, e o significado de *game design* e *game designer*. O Capítulo 3 discute os aspectos metodológicos do trabalho. A pesquisa é caracterizada segundo seus procedimentos e objetivos. As etapas da pesquisa mostram de forma sucinta cada fase e como todas se conectam na construção da resposta para o nosso problema. O Capítulo 4 apresenta uma revisão da literatura a cerca de livros e artigos relacionados ao *game design* distribuídos por temas comuns aos interesses deste trabalho. No Capítulo 5 explicamos como os documentos analisados neste trabalho foram desenvolvidos e organizados. Fazemos também breves explicações para amparar o leitor sobre suas finalidades. O Capítulo 6 relaciona a revisão da literatura com a experiência registrada no desenvolvimento dos jogos. Também associa o potencial matemático dos problemas que o jogador enfrenta em cada um dos jogos. E faz uma releitura sobre dados coletados da aprendizagem de Matemática dos jogadores antes e depois de jogarem. O Capítulo 7 reúne as experiências e relações levantadas neste estudo para determinar uma trajetória onde *designers de problemas* possam construir *puzzles dinâmicos*.

## 2. Fundamentação teórica

O objeto de investigação deste trabalho é o processo prático de construção de um *puzzle dinâmico*, que pertence ao conjunto dos jogos digitais para ensino de Matemática. Assim, consideramos adequado discutir nesta etapa da pesquisa o que são jogos, o que difere quando lidamos com jogos digitais, o que entendemos como jogo para ensino de Matemática e jogo educativo. E o significado dos termos *game design* e *game designer*.



**Figura 6:** Esquema dos subníveis de jogos até *puzzles dinâmicos* (fonte própria).

### 2.1. Jogo

Em nosso idioma a palavra jogo com o sentido de ludicidade (em vez de jogo no sentido de arremessar algo) possui três sentidos comumente utilizados: o objeto que geralmente é utilizado para jogar um jogo (tabuleiro 8x8 com 12 pares de peças idênticas de cores distintas é entendido como um jogo de *Damas*); a ação que realizamos (jogar *Damas*, cartas, videogame, *Basquete*); o conceito que independe de existirem jogadores ou objetos (as regras do jogo de *Damas*, de *Xadrez*, *Basquete*, *Futebol*). Isto nos traz o dilema da definição objetiva de jogo, pois embora o jogo seja naturalmente reconhecido pelos seres humanos, é de difícil definição até pelos principais autores da área (BROUGÈRE, 1998).

Assim, escolhendo nos opor a ideia de jogo como uma atividade imposta e séria, tomamos como base neste trabalho as opiniões de Avedon e Sutton-Smith (1971), Caillois (2001) e Huizinga (1955) que definem o jogo como:

O jogo é uma atividade livre, ficando conscientemente tomada como “não séria” e exterior à vida habitual, mas, ao mesmo tempo, capaz de absorver o jogador de maneira intensa e total. É uma atividade desligada de todo e qualquer interesse material, com a qual não se pode obter qualquer lucro. Ela é praticada dentro de seus próprios limites de tempo e espaço de acordo com regras fixas e de uma maneira ordenada. Promove a formação de agrupamentos sociais, que tendem a se cercar de sigilo e sublinhar a sua diferença em relação ao mundo comum, por disfarce ou outros meios. (HUIZINGA, 1955, p. 13)

De acordo com Avedon e Sutton-Smith (1971): “Jogos são um exercício de sistemas de controle voluntário, em que há uma competição entre forças, limitadas por regras para produzir um desequilíbrio”.

Jogos são atividades: livres, a interação lúdica não é obrigatória; se fosse, perderia de uma só vez sua qualidade atrativa e alegre como diversão. Separadas, circunscrita dentro dos limites de espaço e tempo, definida e fixada antecipadamente. Incertas, o curso da qual não pode ser determinado, nem o resultado obtido previamente e alguma margem para as inovações deixadas para a iniciativa do jogador. Improdutivas, não cria bens, riqueza, nem elementos novos de qualquer espécie; e, com exceção da troca de bens entre os jogadores, termina em uma situação idêntica à que prevalecia no início do jogo. Regidas por regras, sob convenções que suspendem as leis ordinárias e no momento, estabelecem uma nova legislação, que conta sozinha. Que “faz de conta”, acompanhado por uma consciência especial de uma segunda realidade ou fantasia livre, como em oposição à vida real. (CAILLOIS, 2001, p. 9-10)

## 2.2. Jogo digital

O tabuleiro, as peças e o conhecimento das regras do jogo compõe o sistema do jogo de *Xadrez*, assim como o jogo de *Xadrez* de computador, no qual também temos o tabuleiro e as peças projetadas na tela, e as regras que determinam suas interações. A complexidade dos jogos digitais em relação aos jogos não digitais está ligada a algumas qualidades providas de sua estrutura computacional. Logo o que define um jogo serve tanto aos jogos digitais quanto não digitais. Porém, há quatro características, segundo Salen e Zimmerman (2012a), que resumem as qualidades especiais dos jogos digitais. Estas características também estão presentes nos jogos não digitais, mas nos jogos digitais, em geral, estas são incorporadas de forma mais robusta:

Interatividade imediata, mas restrita: um *feedback* imediato, interativo. Projetando sistemas de ações e resultados, onde o jogo responde perfeitamente à entrada de um jogador. Porém restrita ao espaço de possibilidades suportado pelo modelo do jogo.

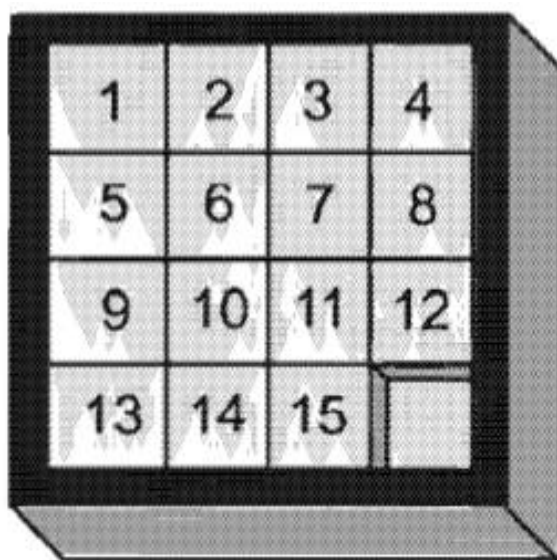
Manipulação da informação: Os gráficos e o áudio não são o único tipo de informação que um jogo digital manipula. Cada aspecto de seu *software* pode ser considerado como informação.

Sistemas complexos e automatizados: Na maioria dos jogos não digitais, os jogadores têm de fazer avançar a partida a cada passo, por meio da manipulação das peças ou comportando-se de acordo com as instruções explícitas descritas pelas regras. Em um jogo digital, o *software* pode automatizar esses procedimentos e fazer o jogo avançar sem a entrada direta de um jogador.

Comunicação em rede: Os jogos digitais oferecem a capacidade de se comunicar por longas distâncias e partilhar uma variedade de espaços sociais com muitos outros participantes (SALEN, ZIMMERMAN, 2012a, p. 103 – 107)

## 2.3. Jogo para ensino da Matemática

Matemáticos desde muitos séculos atrás utilizam conflitos presentes nos jogos como formas de explicar propriedades e trabalhar problemas próprios. Como no famoso *Jogo dos 15*, um quebra-cabeças de quinze peças, composto por uma placa oca de metal com quinze quadrados que trocam de lugar, todos gravados com números, letras ou figuras. Seu desafio é arranjar as peças ordenadamente da esquerda para a direita, de cima a baixo, estas ações no contexto deste jogo, também são usadas para tratar tópicos da álgebra de grupos (PETKOVIC, 2009).



**Figura 7:** *Jogo dos 15* (PETKOVIC, 2009, p. 8).

De forma geral, o jogo (matemático ou não) na infância trabalha habilidades que favorecem a aprendizagem de Matemática, como ilustrado na fala inicial de Zaslavsky (2009) no livro **Mais jogos e atividades Matemáticas do mundo inteiro**.

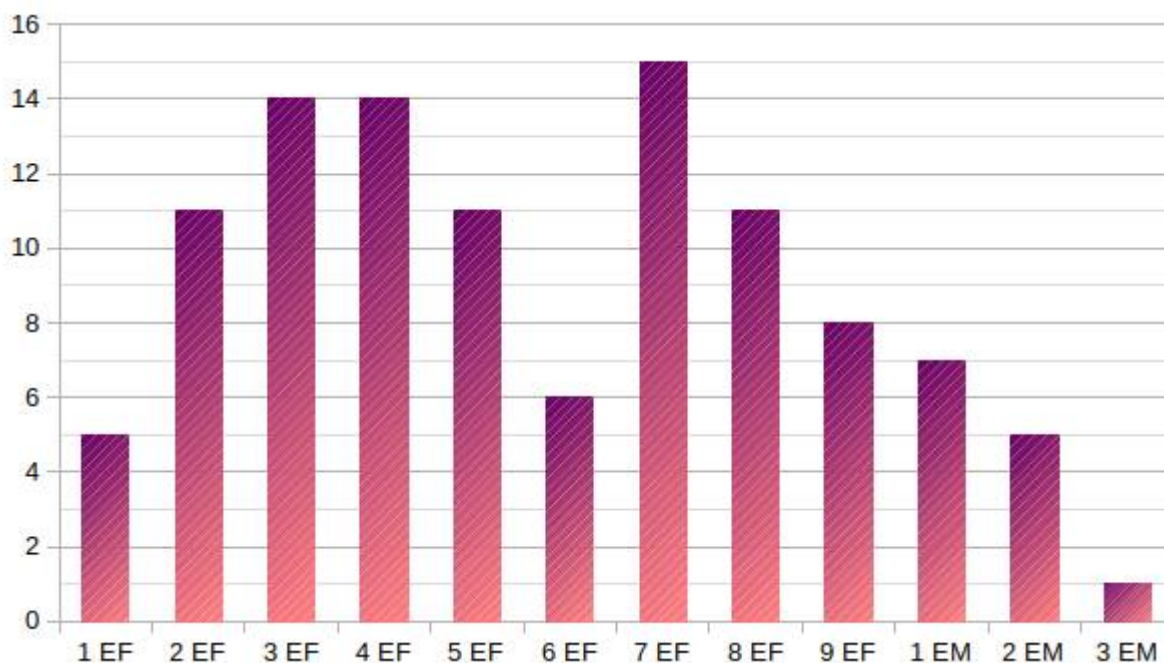
Por que encorajar as crianças a jogar quando há tanta Matemática a ser aprendida? Exatamente por isso! Ao se engajar com essas atividades divertidas, as crianças utilizam muitas habilidades importantes. Elas calculam, medem e resolvem problemas. Elas aguçam suas habilidades em geometria e no reconhecimento de padrões. Acima de tudo, elas aprendem a pensar criticamente (ZASLAVSKY, 2009, p.11).

De acordo com o documento *Principles and standards for school mathematics*, publicado pelo NCTM (National Council of Teachers of Mathematics) em 2000, a Matemática é uma das grandes realizações culturais e intelectuais da humanidade, os cidadãos devem desenvolver uma apreciação e compreensão desta grande conquista, incluindo aspectos estéticos e recreativos. Na perspectiva de mesclar a Matemática com o prazer proporcionado pelo jogo, Smole, Diniz e Cândido afirmam que:

Todo jogo por natureza desafia, encanta, traz movimento, barulho e uma certa alegria para o espaço no qual normalmente entram apenas o livro, o caderno e o lápis. Essa dimensão não pode ser perdida apenas porque os jogos envolvem conceitos de Matemática. Ao contrário, ela é determinante para que os alunos se sintam chamados a participar das atividades com interesse (SMOLE, DINIZ, CÂNDIDO, 2007, p. 12).

Os três volumes dos livros **Cadernos do Mathema Jogos de Matemática** de Smole, Diniz e Cândido (2007) são referências em exemplos de jogos de Matemática para Ensino Básico. Ao analisarmos a quantidade de jogos propostos pelas autoras com relação a seus respectivos públicos-alvo, teremos a seguinte distribuição que vai desde o 1º Ano do Ensino Fundamental I (no gráfico como 1 EF), até o 3º Ano do Ensino Médio (no gráfico como 3 EM) (Figura 8):





**Figura 8:** Jogos presentes nos livros, distribuídos por Ano do Ensino Básico (fonte própria).

Neste contexto, as autoras (Smole, Diniz e Cândido) constroem sua definição de jogo visando atender as necessidades de aprendizagem pelo jogo em aulas de Matemática regulares a partir dos referenciais de Kamii (1991) e Krulik (1983), assim as autoras definem de forma combinada às suas bases teóricas, várias características que o jogo para ensino de Matemática na sala de aula regular deve ter. Sua abordagem não inclui jogos digitais, visto que suas interações estão sempre associadas a outros jogadores e não a um computador. Porém, destas características, uma que pode ser aplicada aos jogos digitais diz que “no jogo deve haver a possibilidade de usar estratégias, estabelecer planos, executar jogadas e avaliar a eficácia desses elementos nos resultados obtidos, isto é, o jogo não deve ser mecânico e desprovido de significado para os jogadores” (SMOLE, DINIZ, CÂNDIDO, 2007, p. 14).

Smole, Diniz e Cândido (2007) propõe a utilização de jogos na perspectiva do enfrentamento de situações-problemas. Amplia o conceito de problema, como situações que não possuem soluções evidentes e que exigem do jogador, a combinação de seus conhecimentos e decisão pela maneira de usá-los em busca das soluções, exige também a adoção de uma atitude de investigação em relação àquilo que está proposto como obstáculo, permite o aparecimento de diferentes soluções, comparando-as entre si e levando a discussão a outros jogadores, sobre suas hipóteses e conjecturas acerca do problema no jogo, traz também a postura de inconformismo frente aos obstáculos ou aquilo que foi estabelecido pelos colegas, estimulando tanto o senso crítico quanto a criatividade.

Não se opondo aquilo que consideramos jogo (BROUGÈRE, 1996, HUIZINGA, 1955, CAILLOIS, 2001, AVEDON, SUTTON-SMITH, 1971) e jogo digital (SALEN, ZIMMERMAN 2012a), podemos somar a definição de Smole, Diniz e Cândido (2007) naquilo que forma um jogo digital para ensino de Matemática. Dessa forma, sintetizamos a discussão desta e das suas seções anteriores para formular aquilo que trataremos neste trabalho como jogo digital para ensino de Matemática: Uma atividade livre, conscientemente tomada como não obrigatória, desprovida de ganhos e reconhecida como exterior à vida habitual. Envolvendo uma competição entre forças, limitada por regras que produzem um desequilíbrio onde seu resultado é incerto aos jogadores. Exigem mídias eletrônicas ou digitais para ser jogado. Envolve conceitos de Matemática durante as decisões tomadas no decorrer do jogo.

## 2.4. Jogo educativo

Por fim, discutimos como a definição de jogo que utilizamos conflua com o significado histórico de jogo educativo. Retornando ao cenário das Casas de Asilo francesas, onde antes de 1834 elas tinham como finalidade somente a caridade, permitindo que a recreação fosse estimulada, ou seja, um espaço destinado ao brincar voltado, ainda que de maneira implícita, ao desenvolvimento infantil. Mas em 1883, estes locais se transformam nas Escolas Maternais Francesas, e a caridade perdeu destaque para a educação, o que trouxe um forte conflito sobre a prática de jogos, mesmo com caráter recreativo, nestes locais, visto que desde a Idade Média até o século XIX, o jogo era relacionado fortemente ao “jogo de azar”. Assim, a educação com jogos influenciaria no aumento do vício em apostas (BROUGÈRE, 1998).

O brincar era aceito como parte do desenvolvimento da criança entre 2 e 6 anos. Mas o mesmo era pouco reconhecido socialmente como uma prática educativa, diferente de uma caridade. Nesse meio, tanto a arte quanto o jogo eram vítimas do julgamento moral de exercitar o ócio. Contudo, as formas de expressão artísticas rapidamente foram aceitas, estando associadas a suas produções. Enquanto o jogo continuava a ser evitado, visto que ao fim de todo um esforço físico, mental e tempo gastos, não se produzia nada. Uma exceção aos jogos quando usados pelos matemáticos para contextualizar situações com árvores de decisões complexas (BROUGÈRE, 1998, PETKOVIC, 2009).

Assim, inserir o jogo na escola somente foi possível mediante o auxílio da palavra “educativo”, já que o jogo educativo não teria o papel de ser jogado, mas sim de enganar a criança, fazendo-a realizar uma tarefa de trabalho (oposta ao ócio) mascarada com ludicidade (jogo). Neste cenário o jogo passou a ser aceito, porém extremamente vigiado e com cobranças excessivas, o

professor ou tutor precisava acompanhar todo o processo, seu uso era em períodos limitados e com o único objetivo de facilitar a aprendizagem (BROUGÈRE, 1998).

Os jogos precisaram ser mascarados como jogos educativos para serem permitidos na educação ocidental. Como efeito, sua definição gradativamente se alterou para algo positivo e saudável. Contudo sua aceitação aparece vinculada com a ideia de um trabalho, mascarado com a ludicidade (BROUGÈRE, 1998). Um exemplo disto, são as instruções passo a passo para o uso de jogos nas escolas, que removem o caráter de atividade livre e voluntária ao jogo. Tais instruções estão presentes nas Diretrizes Curriculares Municipais de Bauru, que regulamentam o uso dos jogos na escola onde um dos testes do jogo digital para ensino de Matemática desta pesquisa foi realizado:

Planejamento: escolha do jogo adequado ao conceito que será trabalhado; exploração do jogo pelo docente e análise minuciosa das regras; inserção adequada do mesmo na sequência didática que será desenvolvida em sala de aula; distribuição do tempo que será utilizado para a execução do jogo.

Execução: apresentação do jogo e de seus objetivos aos alunos; discussão das regras e adaptações caso sejam necessárias; organização dos alunos em grupos; monitoramento durante a execução do jogo problematizando-o; definição de como serão realizados os registros advindos das jogadas.

Avaliação: discussão detalhada sobre o jogo e suas regras para a validação destas regras; análise das situações registradas pelos alunos participantes e problematização das mesmas; elaboração de textos e desenhos explorando o que os alunos aprenderam a partir da experiência de jogar (BAURU, 2013, p. 223).

Embora estas descrições digam respeito a jogos educativos no ponto de vista histórico, na prática as palavras jogo e jogo educativo se misturam no discurso, mas continuam diferindo em sentidos. Assim, embora todo jogo possa ter um conteúdo associado, o que o torna educativo aos olhos do docente, é sua relação a uma parte do currículo comum. Para melhor entender esta relação, trabalharemos a seguir com alguns exemplos de jogos para ensino e jogos educativos.

**a)** Jogo digital para ensino de Entomologia: Um jogo sobre tirar fotos de borboletas representadas de modo proporcional e condizentes com suas respectivas espécies, embora seu conteúdo seja ligado à Biologia, isto representa uma pequena parcela do currículo. Assim, conhecer os nomes, espécies, descrever borboletas entre outros aspectos, seria pouco aplicável em avaliações curriculares onde a dimensão do assunto abordado é bastante ampla.

**b)** Jogo digital para ensino de Automobilismo: Um jogo sobre corrida de carros embora trabalhe vários conceitos sobre cilindradas, potência do motor, qualidade dos pneus aerofólios, entre outros, isto pouco afetaria o desempenho escolar de alguém que entenda destes assuntos. Visto que estes não são conhecimentos cobrados no currículo.

- c) **Jogo digital para ensino de Matemática:** Um jogo sobre as derivadas e integrais de funções de variáveis complexas. Muito embora este seja um conteúdo matemático, e a Matemática está presente no currículo, o grau que este tópico aparece, é elevado demais para o Ensino Básico.
- d) **Jogo digital educativo de Matemática:** Um jogo sobre as derivadas e integrais de funções de variáveis complexas. Dentro de disciplinas do Ensino Superior envolvendo o Cálculo com funções de variáveis complexas, aos olhos dos docentes este seria um jogo educativo, visto que proporcionaria um reflexo no desempenho dos alunos por meio das avaliações.
- e) **Jogo digital para ensino de Geografia/História:** Um típico jogo de estratégia militar envolvendo países ou regiões em conflito. Fora da disciplina de geografia ou história, este jogo pouco afetaria nos seus jogadores o desempenho nas avaliações.
- e) **Jogo digital educativo de Geografia/História:** Um típico jogo de estratégia militar envolvendo países ou regiões em conflito. No contexto de estudar sobre conflitos entre países ou regiões, ou até mesmo suas respectivas posições no mapa, entender quais são vizinhos e a distância entre cada um, o fato de jogar poderia afetar o desempenho destes jogadores nas avaliações destas disciplinas.

Dessa forma, esperamos expressar nossa visão sobre a principal diferença prática entre criar um jogo ou um jogo educativo, é sua finalidade. Então, embora este texto discuta um jogo que utilize tópicos da Matemática, e por isto, possua uma preocupação em ensiná-los, a forma como este trabalho ocorre esta desprendida de uma aplicação dentro de currículos ou faixas etárias. Sendo seu objetivo, proporcionar um jogo capaz de ensinar a Matemática necessária para jogá-lo, do mesmo jeito que um jogo de automobilismo ensinaria sobre carros para que o jogador o jogue. Razão esta, que trataremos do *puzzle dinâmico*, neste caso, como um jogo digital para ensino de Matemática.

## **2.5. Game design e game designer**

Rogers (2012) embora não defina o *game design*, trabalha no simples pressuposto de que *game design* é criar jogos, e isto envolve todas as etapas de sua produção. Neste sentido específico o *game designer* tem a função de criar ideias e regras que compreendem o jogo, sendo capaz de identificar e comunicar adequadamente as diferenças entre jogos bons e ruins.

Schuyttema (2008) explica o *game design* como a planta baixa de uma casa, onde não bastando ter a mão de obra e os materiais, é preciso saber como a casa deve ser para manejar os recursos e transformá-lo em uma estrutura completa (neste caso, em um jogo). O *game designer* neste contexto é a pessoa designada para criar o plano diretor que decide como os elementos serão combinados para criar o jogo.

Chandler (2012) aborda o *game design* de forma implícita como o processo de produção que visa construir jogos digitais. Neste contexto o *game designer* dentro da equipe de produção fica responsável por projetar os sistemas de controle do jogo, criar histórias e desenvolver perfis de personagem, levando ao jogador uma experiência atraente e imersiva.

Salen e Zimmerman (2012a) definem *game design* a partir da ótica do *game designer*. O *game designer* projeta sistemas dos quais emergem interações significativas com os jogadores que interagem com sistemas sociais e culturais maiores. O *game design* é o processo de construção destes sistemas.

De modo simplista, *game design* é o processo de fazer jogos e *game designer* é o sujeito quem planeja o jogo. Porém, o que é um jogo e o propósito para construí-lo, divergem entre grupos. Para o currículo comum de Bauru (2013), jogo é o que apresentamos aqui como jogo educativo. Na visão comercial de Chandler (2012), os jogos precisam levar aos jogadores experiências atraentes e imersivas. Já para Salen e Zimmerman (2012a), jogos são sistemas que interagem significativamente com os jogadores. De qualquer modo, para entendermos o sentido de *game design* e *game designer* dentro de um grupo, é essencial saber o que eles particularmente entendem por jogo. E isto não pode ser generalizado, pois conforme Avedon e Sutton-Smith (1971), há uma personalidade dos jogos perante quem os estuda, afetando também o significado de *game design* e *game designer*:

Cada pessoa define jogos a sua própria maneira – os antropólogos e especialistas em folclore, em termos de origens históricas; os militares, empresários e educadores em termos de usos; os sociólogos em termos de funções psicológicas e sociais. Em tudo isso há provas contundentes de que o significado dos jogos é, em parte, uma função das ideias daqueles que pensam a respeito deles. (AVEDON, SUTTON-SMITH, 1971, p. 438)

### 3. Metodologia

Com base em seus objetivos gerais, este trabalho é um estudo exploratório. Entretanto, de acordo com os procedimentos técnicos utilizados, temos um estudo tanto bibliográfico como documental. A seguir justificaremos estas classificações.

A pesquisa exploratória visa tornar um problema mais explícito e construir hipóteses sobre ele. Esta forma de pesquisa apresenta uma estrutura pouco rígida, permitindo a inclusão dos mais variados aspectos relacionados ao objeto estudado. As principais ferramentas utilizadas nestas pesquisas e presentes neste trabalho, são os estudos bibliográficos e documentais (SELLTIZ, 1967).

Um estudo bibliográfico se desenvolve, principalmente, com base em livros e publicações periódicas. Neste trabalho, os textos de teorias do *game design* e guias para *game designers* que utilizamos na revisão do tema, constituem fontes bibliográficas da categoria livro de leitura corrente – obras de divulgação. Pois, se objetivam em proporcionar conhecimentos científicos ou técnicos sobre *game design* (GIL, 2002).

Por outro lado um estudo documental, embora seja semelhante ao bibliográfico, se vale, principalmente, da documentação por fontes que não receberam um tratamento analítico. Isto permite a inclusão de diversas mídias de registro, como cartas pessoais, relatórios, fotografias, gravações, vídeos ou no caso deste trabalho, diários de *game design* e jogos no formato digital (GIL, 2002).

Revisar a literatura é uma forma de relatar o que já foi publicado sobre um tópico por acadêmicos e pesquisadores conceituados, com o objetivo de transmitir aos leitores os conhecimentos e conceitos estabelecidos a respeito de um tópico junto de seus pontos fortes e fracos. Sua construção não deve ser somente descritiva ou catalográfica, ela precisa seguir um conceito norteador para um tema, no caso deste trabalho são revisadas teorias abordando o *game design* por diversos vieses (problemas matemáticos, ensino, gerenciamento de projetos...) a fim de trazer aos leitores uma visão ampla sobre como o *game design* pode ser pesquisado em diferentes campos do conhecimento (TAYLOR, PROCTER <<http://advice.writing.utoronto.ca/types-of-writing/literature-review/>> acesso em 18/04/2018).

Os materiais da literatura de *game design* revisados, foram escolhidos por sua abrangência de tópicos, variando de Linguística, Psicologia, Matemática e tantos outros temas que são necessários tratar para a construção de jogos, como Jornalismo ou Metodologia de pesquisa. Além de livros sobre fundamentos do *game design* e de guias para *game designers*. Um critério por conveniência para seleção de livros foi sua disponibilidade na coleção da biblioteca UNESP – Campus de Bauru.

Foram ignorados nesta revisão os livros que: exigissem altos conhecimentos em computação para entendê-los; seu tema envolvesse a capitalização de recursos a partir dos jogos; os aspectos tratados nele fossem de natureza técnica, destinada a profissionais específicos (arte, som, software) da indústria de jogos. Foram considerados os idiomas inglês e português para a escolha dos materiais. O total de trabalhos para a revisão foi determinado considerando o tempo disponível para a elaboração desta pesquisa. A revisão foi estruturada a partir da leitura de cada obra e no desenvolvimento de uma síntese explicando seus principais aspectos teóricos ou como suas ideias podem ser desenvolvidas neste trabalho.

Foram considerados como documentos nesta pesquisa os materiais produzidos durante o processo de desenvolvimento do *puzzle dinâmico*, o que consta dentro de um intervalo de 24 meses (março/2016 até fevereiro/2018). Deste período, usamos os seguintes documentos: um diário de *game design*; cinco jogos digitais; dois modelos de questionários avaliativos e suas respectivas planilhas de resultados computados, dos quais, com exceção destes resultados, os demais itens podem ser localizados a partir deste texto. Também incluímos uma breve biografia explicando ao leitor os conhecimentos técnicos e recursos disponíveis pelo autor dos documentos no início do período investigado. Uma descrição sucinta de como o diário de *game design* foi construído e estruturado em um modelo de produção proposto por Chandler (2012). Também são apresentadas sínteses sobre os cinco jogos digitais, além de uma breve explicação a cerca dos interesses e objetivos considerados para a elaboração destes questionários, justificando que não fossem utilizados questionários padronizados.

Embora o trabalho com registros feitos em diários tenha por vezes sua credibilidade colocada a prova por falta de uma triangularização que valide seus acontecimentos, Zabalza (1994) discute este como um documento de caráter único capaz de revelar a tomada de posição de uma pessoa que não poderia ter seu comportamento estudado de outras maneiras. Assim, o trabalho com diários visa encontrar nas redundâncias de um texto longitudinal, evidências que creditem os fatos pela frequência com que ocorrem.

No caso deste trabalho, temos um recurso a mais para credibilizar o registro do diário em questão, o produto. Pois diferente dos diários de aula, onde a observação da prática docente poderia afetar a própria realização da aula, no caso do diário de *game design*, seu registro ocorre pela produção de suas partes, as quais podem ser acompanhadas de forma datada e registrada no computador em que se trabalhou ou nos seus rascunhos, esquemas e protótipos, os quais embasaram sua elaboração. Materiais que resgatam mais facilmente a evolução dos acontecimentos e a maneira com que seu autor pensou durante o processo. Outro modo de reforçar os fatos do estudo deste

diário, está nos versionamentos do jogo, que acompanham o processo narrado de acordo com sua descrição.

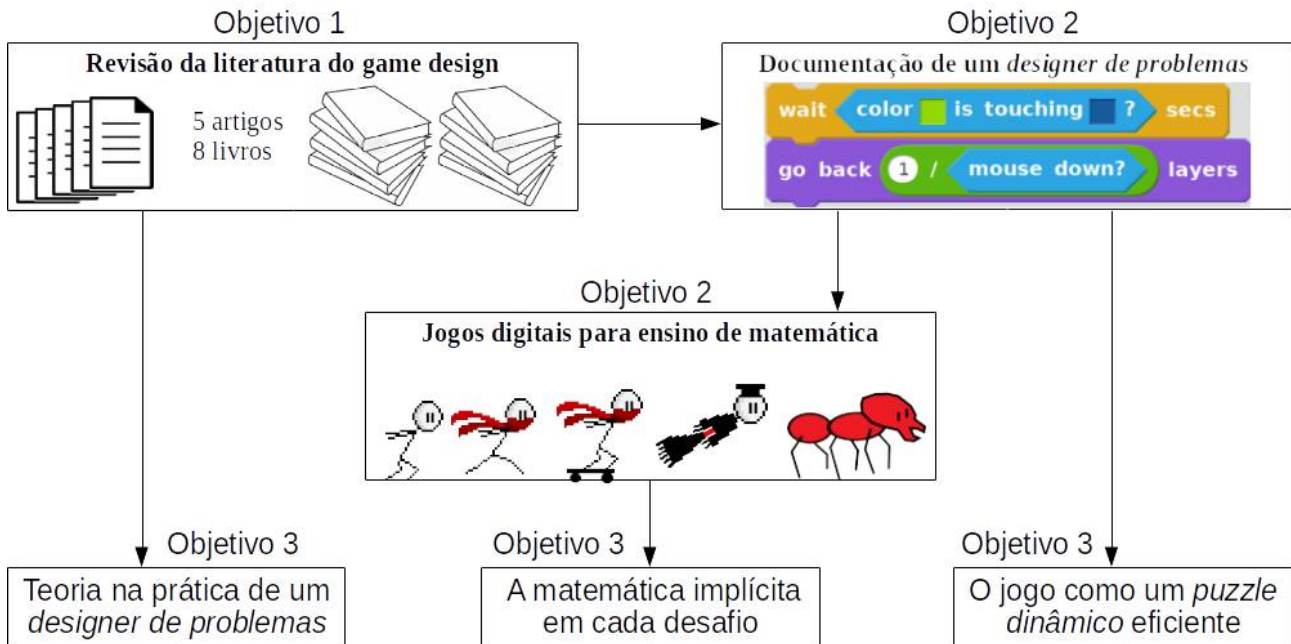
A conexão entre a educação Matemática com a construção do *puzzle dinâmico* e a literatura revisada ocorreu em três vias. Na primeira, analisamos os conteúdos matemáticos que podem ser explorados a partir dos desafios propostos nos 5 jogos digitais produzidos, relacionando o conteúdo a: problemas matemáticos generalizados; problemas matemáticos contextualizados ao jogo e sua respectiva solução em linguagem Matemática; uma ilustração representando a primeira aparição do problema contextualizado dentro do jogo.

Na segunda, exploramos o efeito do jogo na aprendizagem de Matemática a partir dos resultados dos dois questionários documentados. Para isto foram analisados os conceitos matemáticos envolvidos em cada questão dos testes utilizados para considerar quais aspectos foram melhor aprendidos pelos jogadores. Por fim, interpretamos os gráficos do desempenho de cada grupo de jogadores.

Na terceira, relacionamos a narrativa do diário de *game design* com os temas explorados na revisão da literatura, indicando onde esta prática divergiu ou coincidiu com a teoria discutida. Distribuímos cada obra tratada na mesma ordem como aparecem na revisão da literatura. Procuramos construir associações simétricas, no sentido da narrativa se relacionar diretamente com uma teoria. Mas por vezes utilizamos de relações assimétricas, no sentido da teoria se relacionar com o diário de *game design*, ainda que sua narrativa não discuta aspectos da teoria.

A Figura 9 mostra um diagrama das relações entre os objetivos desta pesquisa. As setas indicam a direção na qual o respectivo bloco mais contribuiu. Para evitar confusões na interpretação, retomamos conforme foi discutido na introdução e na fundamentação teórica, que um *puzzle dinâmico* pertence ao conjunto dos jogos digitais para ensino de Matemática. Logo, sua eficiência esta ligada não só a seu papel como jogo, mas também na sua função como recurso para ensino de Matemática.





**Figura 9:** Esquema de objetivos da pesquisa (fonte própria).

No caso, esta também pode ser considerada uma pesquisa auto-etnográfica, dado que parte do material de estudo é a análise dos documentos gerados pelo próprio pesquisador, como o diário de *game design* construído na forma de narrativa em primeira pessoa ou seus jogos digitais produzidos e avaliados durante o processo de construção por critérios pessoais. Ambos refletem uma ação que já ocorria desde antes do período estudado. Dessa forma, como o pesquisador experimentou o próprio contexto do perfil de sujeito estudado, passando pelos dramas vivenciados por um *designer de problemas* na ação de produzir um *puzzle dinâmico*, estas refletem tanto um estudo sobre um material construído na experiência, quanto uma autocrítica baseada em seu próprio comportamento, visto que o faz tratando-o como documentos objetivos e inter-relacionando-os com diversas teorias do *game design*, apresentando de forma descritiva e conectada com aspectos comuns e pertinentes à educação Matemática, como na transcrição dos desafios do jogo para problemas matemáticos generalizados e na análise dos dados dos testes (HAMMERSLEY, ATKINSOM, 1983).

Os dados desta dissertação foram coletados mediante a aprovação do projeto de pesquisa “Pensamento Computacional no Ensino da Matemática” pelo Comitê de Ética e Pesquisa da Faculdade de Ciências vinculada à Universidade Estadual Paulista “Júlio de Mesquita Filho” (CAAE: 67075617.0.0000.5398).

## 4. Revisão da literatura

Foram revisados 13 trabalhos (entre artigos e livros), seus anos, autores, temas comuns a esta discussão e o tipo de documento estão listados no quadro abaixo. Neste capítulo, apresentaremos uma breve síntese desenvolvida em cima dos principais conteúdos observados, relacionáveis com esta pesquisa e como estes são discutidos em cada obra. Sua distribuição segue a ordem em que o quadro foi organizado (ano, nome dos autores em ordem alfabética, tema).

**QUADRO 1:** Autores e obras da revisão da literatura (fonte própria).

Ano	Autores	Tema	Tipo de documento
1996	Fellows	Jogos como problemas	Artigo
2013	Costikyan	Jogos como problemas	Livro
2003	Zimmerman	Modelo de pesquisa em <i>game design</i>	Artigo
2004	Hunicke, LeBlanc, Zubek	Modelo de pesquisa em <i>game design</i>	Artigo
2005	Gee	Aprendizagem nos jogos	Artigo
2009	Nakamura, Csikszentmihalyi	Aprendizagem nos jogos	Artigo
2010	Bogost, Ferrari, Schweizer	Aprendizagem nos jogos	Livro
2013	Juul	Aprendizagem nos jogos	Livro
2014	Koster	Aprendizagem nos jogos	Livro
2008	Schuytema	Guia para <i>game designers</i>	Livro
2012	Rogers	Guia para <i>game designers</i>	Livro
2012	Chandler	Gerenciamento de projetos em <i>game design</i>	Livro
2012	Salen, Zimmerman	Fundamentos do <i>game design</i>	Livro

### 4.1. Jogos como problemas

O artigo de Fellows (1996) “**The Heart of Puzzling: Mathematics and Computer Games**”, descreve como as ideias Matemáticas modernas podem ser exploradas não apenas no aumento do número de jogos para ensino de Matemática, mas também incorporando enigmas inovadores e estruturas de interação sem precedentes aos jogos digitais em geral. A discussão começa definindo os conteúdos dos enigmas e das estruturas de interação como espinhas dorsais da maioria dos jogos digitais. Por sua vez, elas podem ser reduzidas a problemas de natureza Matemática. Com isso, existem muitas oportunidades inexploradas para aproveitar pesquisas atuais em Matemática quando se trata de construir jogos digitais.

O autor introduz seu método argumentando que os jogos digitais são rígidos, pois quando o jogador descobre a receita para derrotar o inimigo ou superar um obstáculo, isto torna-se trivial. Ou seja, uma fórmula confiável e replicável para resolver um problema específico foi encontrada. A

natureza computacional do jogo digital por mais que se assemelhem a realidade, tornam a regra inaplicável a problemas da vida real, cujas formas de superá-lo não foram anteriormente definidas ou governadas por um código entendível. Assim, se a conclusão do desafio no jogo fosse um teorema, seu percurso seria composto por diversos lemas. Pessoas com muita experiência em jogos digitais aprenderam a fazer suposições razoáveis e se comportar (embora inconscientemente) como pesquisadores matemáticos. Retornando ao enigma como o coração do problema, que na sua ausência dificilmente será criado o fascínio viciante do jogo digital. Isso é notavelmente semelhante às experiências mentais subjetivas da pesquisa Matemática.

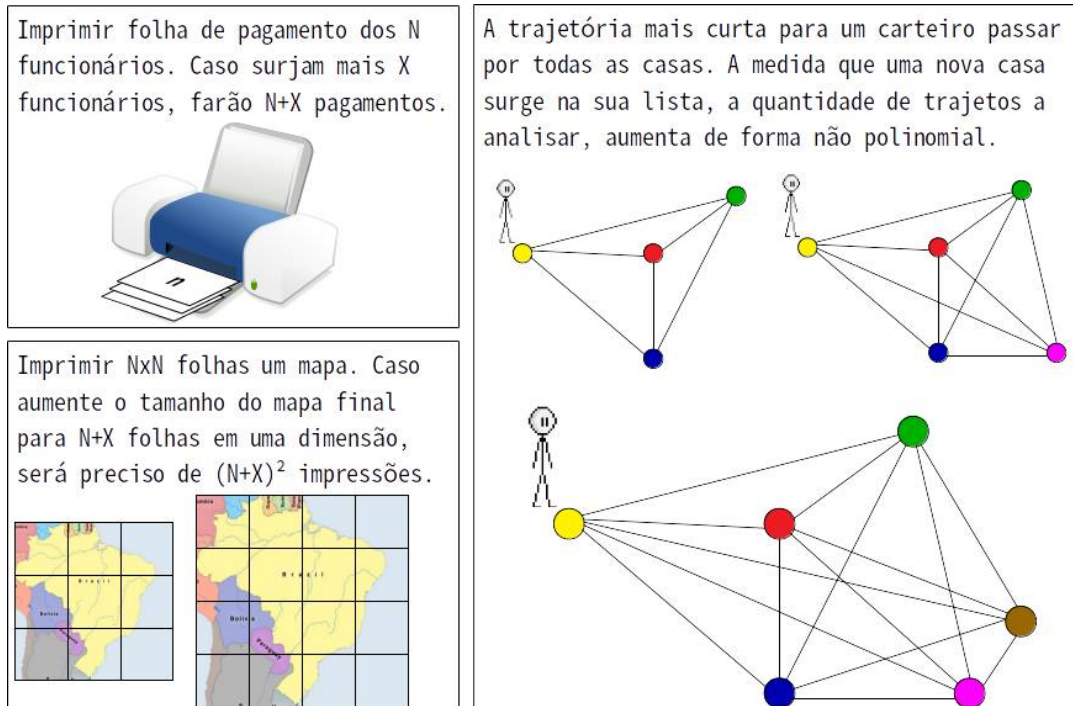
Seu método é baseado no trabalho de referência **Computers and Intractability: A Guide to the Theory of NP-completeness** de Garey e Johnson (1979). Este livro cataloga os dramas computacionais das mais variadas situações e resume suas respectivas essências Matemáticas. Nestes catálogos, temos os problemas da classe NP, que serão a matéria-prima deste método. Estes são problemas cujo tempo de complexidade da função para resolvê-los é não-polinomial (daí o termo NP). Embora não seja intenção neste trabalho aprofundarmos a ideia de problemas NP, para melhor compreendermos o seu efeito no método proposto por Fellows (1996), apresentamos um quadro traduzido de Garey e Johnson (1979) que compara o tempo necessário para resolver problemas com soluções polinomiais e não-polinomiais, junto ao seu comportamento conforme o número de elementos ( $n$ ) do problema aumenta.

Tempo de complexidade da função	Tamanho de n					
	10	20	30	40	50	60
$n$	.00001 segundos	.00002 segundos	.00003 segundos	.00004 segundos	.00005 segundos	.00006 segundos
$n^2$	.0001 segundos	.0004 segundos	.0009 segundos	.0016 segundos	.0025 segundos	.0036 segundos
$n^3$	.001 segundos	.008 segundos	.027 segundos	.064 segundos	.125 segundos	.216 segundos
$n^5$	.1 segundos	3.2 segundos	24.3 segundos	1.7 minutos	5.2 minutos	13.0 minutos
$2^n$	.001 segundos	1.0 segundos	17.9 minutos	12.7 dias	35.7 anos	366 séculos
$3^n$	.059 segundos	58 minutos	6.5 anos	3855 séculos	$2 \times 10^8$ séculos	$1.3 \times 10^{13}$ séculos

**Figura 10:** Tempos necessários para soluções (GAREY, JOHNSON, 1979, p. 7, traduzido)

Exemplificando as funções da figura 10 dentro de situações mais concretas, propomos três situações comuns. Uma função de complexidade  $n$  poderia ser a impressão da folha de pagamento de  $X$  funcionários, assim, são necessários  $X$  processos para isto, e se um dia surgirem mais  $Y$  funcionários, o trabalho aumentará  $Y$  processos em relação ao serviço anterior. Uma função de complexidade  $n^2$  poderia ser a impressão de um mapa quadrado distribuído por segmentos a partir de folhas a serem coladas umas nas outras, assim, a impressão do mapa final é  $N \times N$  onde  $N$  é o número de folhas de cada lado do quadrado exigindo  $N^2$  impressões. No caso deste mapa ter que ser aumentado para mais  $M$  folhas em cada lado, o total de folhas a serem impressas aumentará  $NM + M^2$  em relação ao mapa anterior. Já uma função não-polinomial pode ser pensada como a análise das trajetórias de um carteiro para passar por todas as  $X$  casas na menor rota possível. Cada casa acrescentada ao seu itinerário exige uma nova busca por trajetórias a serem percorridas, dado que elas estão diretamente relacionadas umas com as outras e a árvore de caminhos possíveis aumenta na mesma ordem que a quantidade de casas. Se considerarmos trajetórias que não passam

duas vezes pela mesma casa, teríamos então o fatorial de  $(X-1)$  possibilidades para analisar. Onde o fatorial de um número Natural significa este número vezes todos os seus antecessores até chegar em 1.



**Figura 11:** Exemplos de problemas com funções soluções das ordens  $n$ ,  $n^2$ , NP (fonte própria).

Outra propriedade chave deste método é o conceito próprio do autor das manivelas para manipulação de informações. Estes são mecanismos com os quais podemos tratar a informação de um problema de Matemática, como se fosse uma caixa de ferramentas disponível para enfrentar uma adversidade, se elas são ideias ao desafio, temos um caso típico dos problemas reais de Matemática, porém quando mesclamos ferramentas não-convencionais a estes problemas, lidamos com as famosas “gambiaras” que são muito aproveitáveis no quesito de criar estruturas para jogos digitais inovadores. Para facilitar a compreensão, o autor exemplifica as manivelas de reparação discreta e de embalagem bidimensional em linha.

- a) “A manivela de reparações discretas é modelada até certo ponto no drama de informações do jogo Campo Minado. A essência do problema é que existe uma solução errada escondida e que precisa ser reparada, mas sem expor as falhas” (FELLOWS, 1996, p.5, tradução própria).
- b) A manivela de embalagem bidimensional em linha é o coração do jogo Tetris, onde a solução deve ser construída à medida que a instância é revelada.

Assim o autor propõe que inúmeros enigmas divertidos e desafiantes podem ser construídos de forma sistemática usando duas etapas simples:

- a) Identificar uma manivela para manipulação de informação;
- b) “Anexar” esta manivela ao catálogo de problemas NP e “girá-la” quantas vezes desejar.

Em seu método, podemos interpretar o “anexar” da manivela no catálogo, ao ato de modelar um problema NP nesses mecanismos de manipulação de informações. E podemos entender o “girar” da manivela, como o processo de trabalhar valores nestes problemas de diversas maneiras até obter uma estrutura jogável. Ou seja, tentar resolver o problema com uma ferramenta, até que calibremos dificuldade do problema para o quanto a ferramenta é capaz de facilitar sua resolução. Veja por exemplo Tetris, este jogo seria inviável se o quadro fosse uma matriz de 100x50 espaços para peças, ou se as peças fossem formadas por permutações de grandes quantidades de bloquinhos adjacentes. Mas após “girar algumas vezes a manivela neste problema” podemos perceber por exemplo, que em uma matriz de 15x10 espaços para peças e com peças formadas de quatro bloquinhos adjacentes permutados, temos um certo equilíbrio no qual é possível enxergar intuitivamente algumas boas soluções para o problema de otimização combinatória utilizando a manivela de embalagem bidimensional como ferramenta de resolução. Ao fechar o método, o autor comenta que os jogos digitais não são apenas enigmas isolados, e precisam também de envolvimento com uma história. Para isso, sugere anexá-los a fonte do próprio problema NP, já que a maioria deles foi formulado por uma necessidade de solução real.

O livro de Costikyan (2013) **“Uncertainty in Games”**, descreve como os fatores que produzem incertezas nos jogadores são as características primárias a todos os tipos de interações dos jogadores, mas não dos jogos. Pois analisando apenas os jogos, a vitória no *Pedra-papel-tesoura* deveria ser arbitrária se os jogadores não tentassem antecipar as decisões de seus adversários, enquanto o *Xadrez* seria determinístico se os jogadores conhecessem todas as jogadas possíveis. Neste sentido, o autor coloca o jogo como um subgrupo do ato de jogar, e com isto, herda do jogador a característica de incerteza. Nesta direção temos por exemplo o episódio 2 do anime *“No Game No Life”*, no qual o protagonista (à esquerda na figura 11) atribui desvantagens às suas próprias ações no jogo *Pedra-papel-tesoura* com o objetivo de provocar um comportamento mais previsível na sua adversária.



**Figura 12:** Situação de antecipação no *Pedra-papel-tesoura* (No Game No Life, episódio 2).

O autor discute as incertezas emergentes de diversos jogos bem difundidos e de como estas se distanciam da própria incerteza do jogo sem o jogador, com o intuito de equipar o leitor de exemplos acerca das explicações mais detalhadas sobre os princípios da incerteza. Assim os jogos exemplificados são respectivamente: *Super Mario Bros*; *The Curse of Monkey Island*; *FPS Deathmach Play*; *Pedra-Papel-Tesoura*; *Diplomacy*; *Monopoly*; *Xadrez*; *Roller Coaster Tycoon*; *Poker*; *Rogue-Likes*; *CityVille*; *Memoir '44*; *Civilization V*; *Magic: The Gathering*; *Gardens of Time*. Uma vez que o leitor compreenda como cada um dos jogos apresentados trata a incerteza ou a incerteza aparente dos jogadores, o autor explicita 11 princípios de incerteza ligados ao ato de jogar. Considerando um maior interesse desta seção sobre jogos como problemas, trataremos a seguir das 11 incertezas segundo a visão do autor, dando destaque à incerteza do tipo complexidade analítica, considerada um requisito ao *puzzle* dinâmico descrito na introdução deste trabalho.

- a) Incerteza de performance: O sucesso depende do desempenho do jogador na realização de ações. Como nos jogos de tiro em primeira pessoa, corrida, ação, aventura e outros semelhantes.
- b) Incerteza de solução: Envolve a resolução de um problema importante sob certas condições. Como em jogos de investigação onde cada jogador coleciona pistas e desenvolve hipóteses.
- c) Imprevisibilidade do jogador: Em contextos com muitos jogadores, nos quais as intenções individuais não são claras. Como em jogos que permitem resolver conflitos de interesses atacando os outros.

- d) Aleatoriedade:** Contextos nos quais as variáveis aleatórias compõe ou estruturam o jogo. Isto pode ser usado tanto para quebras de simetria e padrões, ou para dinâmicas emergentes como o blefe.
- e) Complexidade analítica:** Em jogos nos quais os jogadores precisam pensar sobre o que fazer dentro de uma árvore de decisão complexa, a incerteza permanece em torno da decisão certa ter sido tomada. Um exemplo simples e conhecido com esta incerteza é o *Xadrez*. O autor também apresenta o que ele considera um caso extremo relacionado a complexidade analítica, o jogo *The Campaign for North Africa*, com duração prevista para 1.500 horas e 90 páginas de regras. Jogos que permitem emergir nos jogadores a incerteza de complexidade analítica geralmente possuem camadas em muitos sistemas e mecânicas que interagem umas com as outras em forma de redes, tornando difícil aos jogadores entender o sistema como um todo a medida que o número de jogadas aumenta (semelhante a dificuldade em resolver problemas de categoria NP). Sua complexidade de regras muitas vezes não pode ser coberta por quem as desenvolve, permitindo que os jogadores encontrem estratégias degeneradas que interrompem o desenrolar do jogo. Como em *Magic*, que diversas combinações de cartas capazes de derrotar o oponente na primeira jogada precisaram ser proibidas entre as comunidades de jogadores.
- f) Informações ocultas:** Presente em jogos nos quais o objetivo é a exploração dos cenários, descobrindo sobre o próprio contexto, e em jogos de cartas onde a informação escondida favorece o jogador.
- g) Antecipação narrativa:** Presente em jogos onde os jogadores têm especulações sobre o desenvolvimento da história, sendo incerto a eles se a sequência esperada será consolidada.
- h) Antecipação de desenvolvimento:** Quando o contexto de incerteza se refere as alterações que o jogo sofrerá, onde os jogadores especulam as possíveis mudanças que sofrerão.
- i) Incerteza de cronograma:** A incerteza surge da tomada de muitas decisões em um tempo reduzido para jogar. Levando o jogador a uma especulação de como jogará antes de começar a jogar.
- j) Incerteza de percepção:** Esta incerteza emerge na dificuldade em perceber o que existe no entorno. Um exemplo seria encontrar os itens escondidos nos jogos *Onde está o Wally?*
- k) Contingência Semiótica de Malaby:** Uma incerteza da contextualização no período posterior ao jogo, onde após os jogar, a história é então definida com base nos resultados obtidos durante o jogo.

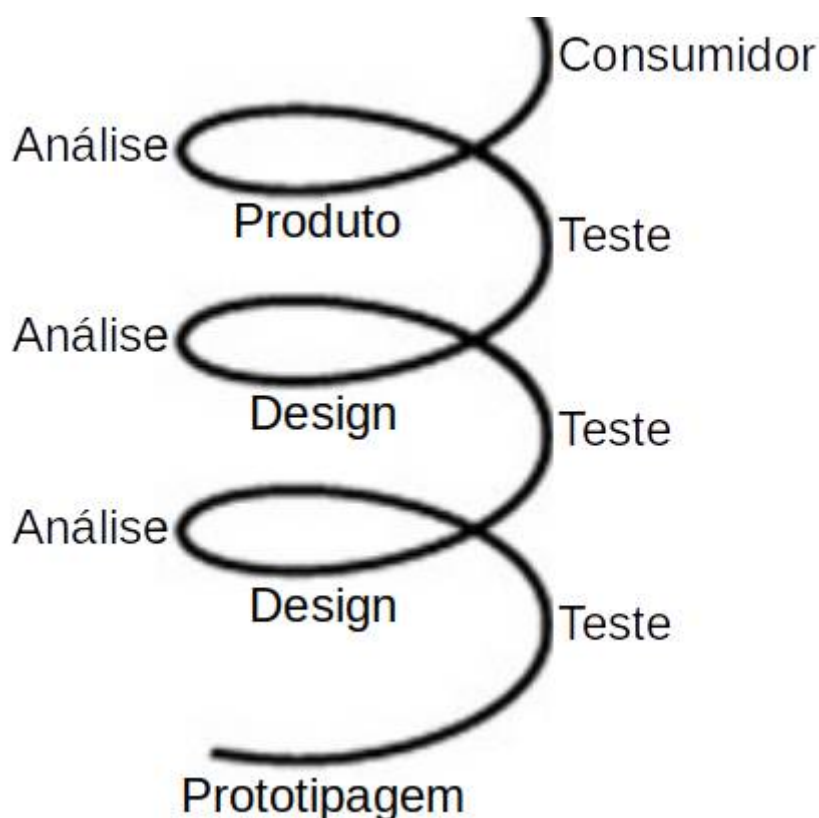
## 4.2. Modelo de pesquisa em *game design*

O artigo de Zimmerman (2003), intitulado **Play as research: the iterative design process**, aborda a pesquisa no campo do design, mais especificamente no *game design*. Como o design é



uma forma de fazer perguntas, a pesquisa de design é uma maneira de fazer perguntas amplas sobre problemas específicos de design. Neste caso, o autor leva o jogo como um produto que pode ser criado e consumido para fins de exercício, interação humana ou aprendizagem de tópicos específicos, mas na maioria dos casos seus usos visam o prazer intrínseco provocado. Assim, a pesquisa em *game design* investiga as formas que um jogador pode desfrutar de um jogo.

O método iterativo é uma metodologia do design baseada no processo cíclico de prototipagem, teste, análise e melhoria dos trabalhos em andamento. No processo iterativo, a interação com o sistema é usada como uma forma de pesquisa para informar e desenvolver um projeto em versões sucessivas ou iterações implementadas. Quando tratamos de jogos, ser iterativo significa testar a jogabilidade.



**Figura 13:** Modelo de pesquisa iterativo (ZIMMERMAN, 2003, p. 177, traduzido).

Zimmerman (2003, p. 177) afirma que esta é uma estratégia diferente do modelo típico em que os jogos são desenvolvidos, onde o *game designer* no início, planeja um conceito acabado, então escreve um documento de design exaustivo que descreve todos os aspectos possíveis do jogo em detalhes, mas quando chegam os testes de interação com os jogadores, todo esse trabalho precisa ser refeito, já que o jogo final dificilmente se parece com o projetado originalmente na cabeça do *game designer*. O *game design* iterativo trabalha com essa variabilidade que o jogo

sofrerá, economizando tempo e recursos do desenvolvimento inicial, permitindo que sejam redirecionados para o produto final, que será mais robusto e bem-sucedido.

A delicada interação entre regras e jogadores é algo muito complexo e emergente para descrever antecipadamente, exigindo o equilíbrio instável que somente o teste do protótipo pode fornecer, sugerindo que o próprio desenvolvedor jogue, sua equipe de desenvolvimento jogue, seus colegas joguem, quem estiver visitando o estúdio jogue, sejam organizados grupos de voluntários que desconheçam o jogo para jogar. Em todos estes casos, com o desenvolvedor observando, fazendo questões aos jogadores, ajustando seu *game design* e buscando colocá-lo para ser testado novamente.

O autor faz a análise do processo de *game design* iterativo em três jogos dos quais trabalhou, enunciando o protótipo e explicando ao leitor como a visão inicial dos desenvolvedores se alterou mediante os primeiros testes de jogabilidade. Os jogos analisados foram: *SiSSYFiGHT 2000*; *LOOP*; *LEGO junkbot*.

De forma a complementar o modelo de Zimmerman (2003), temos o artigo de Hunicke, LeBlanc e Zubek (2004), denominado “**Mda: A formal approach to game design and game research**” que apresenta a estrutura MDA (*Mechanics Dynamics Aesthetics*), uma abordagem formal para a compreensão dos jogos, cujo foco está na conexão entre *game design*, desenvolvimento, crítica e pesquisa técnica em jogos. Esta metodologia procura esclarecer e fortalecer o processo iterativo de desenvolvedores, estudiosos e pesquisadores, facilitando a todas as partes a decomposição, o estudo e a estruturação de uma ampla gama de projetos e artefatos para jogos.

O modelo pressupõe que os jogos são produtos criados por desenvolvedores e consumidos pelos jogadores. Mas, diferentes de livros, músicas ou filmes que possuem uma forma previsível de uso, onde ler o final de uma página antes do começo, ouvir a música em velocidade acelerada ou ver o filme pausando-o não parecem preocupações dos autores sobre o comportamento de seus consumidores, contudo, a forma como os jogadores consomem os jogos é imprevisível. A sequência de eventos que ocorrem e como eles afetam o jogador são um mistério até que o produto seja finalizado, uma consequência que aproxima o processo de *game design* ao que pode ser entendido como um *Wicked Problem*.

Rittel e Webber (1973) tratam o tema *Wicked Problem*, como uma classe de problemas que não pode ser bem definida até que se saiba sua respectiva solução. Muitos dos problemas sociais, políticos e educacionais não podem ser colocados em um ambiente de testes ou ter suas condições controladas sem consequências. Por exemplo, não é possível testar abordagens de prevenção a crimes sem colocar em risco a segurança dos cidadãos, ou experimentar métodos de ensino na

Educação Infantil sem ameaçar o desenvolvimento a longo prazo das crianças envolvidas. Nesta categoria, os autores definem estes problemas com dez características comuns, das quais expomos a seguir:

- 1) Não há uma formulação definitiva para um *Wicked problem*.
- 2) *Wicked Problems* não possuem condição de parada.
- 3) Soluções de *Wicked Problems* não são do tipo verdadeiro ou falso, mas sim, “melhor que” ou “pior que”.
- 4) Não há testes imediatos ou finais de uma solução para um *Wicked Problem*.
- 5) Toda solução de um *Wicked Problem* é uma operação de “tiro-único”; pois não há oportunidade para aprender a partir da tentativa e erro, toda ação conta significativamente.
- 6) *Wicked Problems* não possuem um enumerável conjunto de potenciais soluções (ou exaustiva descrição), nem um grupo bem descrito de operações permissíveis que podem ser incorporadas na realização do plano de solução.
- 7) Cada *Wicked Problem* é essencialmente único.
- 8) Cada *Wicked Problem* pode ser considerado como um sintoma de outro problema.
- 9) A existência de uma representação discrepante para um mesmo *Wicked Problem* pode ser explicada de inúmeras maneiras; a escolha de uma explicação determina a natureza da resolução do problema.
- 10) O planejador não tem o direito de estar errado. (RITTEL, WEBBER, 1973, p. 161-166, tradução própria).

Mateas e Stern (2005) a partir do estudo sobre a construção de um jogo digital com um design experimental, delineiam uma relação de semelhança entre o processo de *game design* com a categoria de *Wicked Problem*, associando-o diretamente com as características 1, 2, 3, 4 e 7 da lista de Rittel e Webber (1973, p. 161-166) recém-apresentada e corroborando com a visão de Hunicke, LeBlanc e Zubek (2004) sobre a dificuldade do *game designer* perceber como jogo chegará ao seu cliente. Abaixo de forma sucinta, apresentamos as características listadas de um *Wicked Problem* segundo Rittel e Webber (1973, p. 161-166), na forma como Mateas e Stern (2005) as associaram:

- 1) Não há uma formulação definitiva para um *Wicked problem*: Se soubesse em detalhes os requisitos do jogo a ser produzido, o próprio já estaria concluído a menos de um processo braçal. (MATEAS, STERN, 2005)
- 2) *Wicked Problems* não possuem condição de parada: O *game designer* produz algo misterioso até que alcance as mãos do cliente, não havendo uma condição de parada senão a escassez de recursos, tempo ou energia. (MATEAS, STERN, 2005)

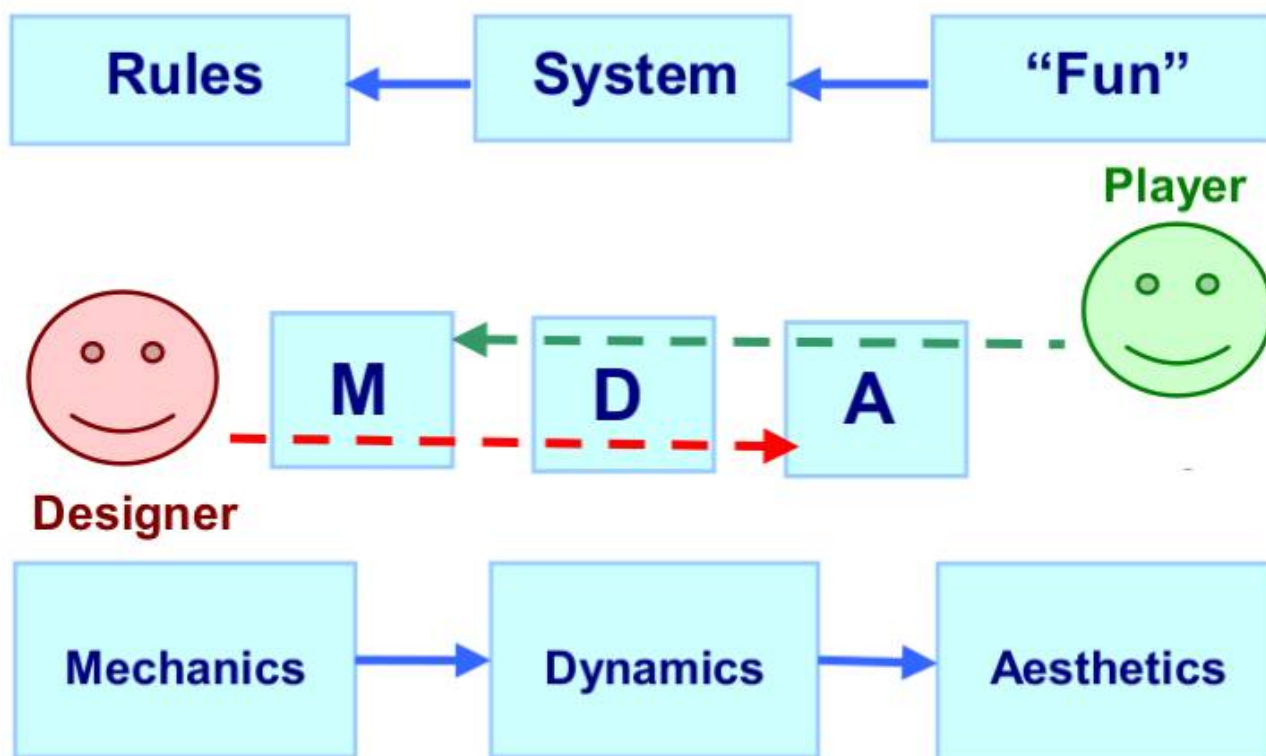
3) Soluções de *Wicked Problems* não são do tipo verdadeiro ou falso, mas sim, “melhor que” ou “pior que”: Um jogo só pode ser comparado de acordo com suas próprias versões, definindo assim se está melhor ou pior. (MATEAS, STERN, 2005)

4) Não há testes imediatos ou finais de uma solução para um *Wicked Problem*: Relacionando-se diretamente a Hunicke, LeBlanc e Zubek (2004) e a Zimmerman (2003), não há testes suficientes para produzir um jogo, dado que sempre que ele for testado, novas considerações podem ser feitas e seu problema pode ser trabalhado para uma solução melhor. (MATEAS, STERN, 2005)

7) Cada *Wicked Problem* é essencialmente único: A solução encontrada para construir um jogo só poderia ser reaproveitada por completo se envolvesse a construção de um jogo idêntico. (MATEAS, STERN, 2005)

Retornando ao modelo MDA proposto por Hunicke, LeBlanc e Zubek (2004), ele fragmenta o consumo de jogos, dividindo-o para estudo, em componentes distintos (Diversão → Sistema → Regras) e estabelecendo seus contrapontos no sentido oposto ao *game designer* (Mecânica → Dinâmica → Estética).

Nesta análise, a mecânica descreve os componentes específicos do jogo, ao nível da representação de dados e dos algoritmos. A dinâmica lida com o comportamento durante o jogo através das entradas dos jogadores e dos elementos de saída do jogo. Estéticas tratam das respostas emocionais desejadas para serem evocadas nos jogadores quando interagem com o sistema do jogo. Do ponto de vista do *game designer*, a mecânica dá origem a comportamentos dinâmicos do sistema, o que, por sua vez, leva a experiências estéticas particulares. Do ponto de vista do jogador, a estética define o tom, de onde nasce uma dinâmica observável e eventualmente a mecânica fica operável.



**Figura 14:** Modelo MDA (adaptado de HUNICKE, LEBLANC, ZUBEK, 2004).

A essência desta estrutura baseia-se na percepção do jogo como um artefato e não como uma mídia. Portanto, o conteúdo do jogo é o comportamento que evoca no jogador e não a mídia em que é transmitida. Pensar no *game design* como desenvolvimento de artefatos ajuda a estruturá-los como sistemas que das interações geram comportamentos. Isso esclarece a tomada de decisões em todos os níveis de estudo e desenvolvimento de jogos. Entre os benefícios desta abordagem para o desenvolvimento do jogo temos que: o *game designer* vê como as pequenas mudanças em uma camada podem afetar outras camadas distantes; o *game designer* é incentivado a construir dinâmicas que encorajem os jogadores a tomarem decisões.

Exemplificando este modelo, uma mecânica comum com cartas seria: embaralhar, mover as cartas na mesa, virar e desvirar cartas, pegar de um monte, descartar para um monte.

Assim, tomando a dinâmica do jogo paciência, temos que as cartas são distribuídas em  $N + 1$  montes, destes o 1º monte tem uma carta virada para baixo, o 2º monte tem duas cartas viradas para baixo, ..., o  $N^{\circ}$  monte tem  $N$  cartas viradas para baixo. Já o  $[N+1]^{\circ}$  monte, tem o restante das cartas. O jogador poderá desvirar as cartas do topo dos montes 1 até  $N$ . Sacar um número  $M$  de cartas do monte  $[N+1]$ , decidir se usará alguma delas e se quiser sacar mais  $M$  cartas deste monte, precisará colocar aquelas já sacadas no final deste monte.

De forma simples, estas são partes da dinâmica do jogo de paciência, não especificando aqui a quantidade de montes, cartas ou naipes. O papel da estética aparece a este respeito, pois existem estruturas prazerosas (como 52 cartas, 4 naipes e 7 montes) e outras que seriam triviais (2 cartas, 2 montes 1 naipe) ou estressantes (1000 cartas, 20 naipes e 50 montes) para serem jogadas.

Do mesmo modo, o jogador percebe o jogo pela sua estética (a composição dos montes, número de cartas e naipes), interage com sua dinâmica (tentando resolver o problema do jogo) e compreende assim as mecânicas que o compõe (embaralhar, distribuir, virar e desvirar cartas).

A completude do modela MDA com o processo cíclico proposto por Zimmerman (2003) aparece na relação de análise dos testes para a nova iteração do produto. Dessa forma, aplicando este modelo é possível conhecer melhor as consequências que uma modificação implicaria tanto para sua construção pelo *game designer*, quanto na sensação provocada aos jogadores.

### 4.3. Aprendizagem nos jogos

O artigo de Gee (2005), intitulado **Good video games and good learning**, aborda que os bons jogos digitais têm como semelhança o fato de incorporarem bons princípios de aprendizagem respaldados por pesquisas recentes da psicologia cognitiva, justificando isto por uma lógica simples, pois se ninguém pudesse aprender a jogá-los esses jogos não seriam comprados. No entanto, os jogadores precisam de jogos complexos onde o desafio e a aprendizagem são muito daquilo que faz um jogo divertido.

Para iniciar a discussão o autor ataca o seguinte jargão, *a única coisa que se aprende ao jogar um jogo digital, é jogar esse jogo digital*, defendendo que apesar da ironia, um jogo ensinar como ser jogado é um excelente princípio de aprendizagem, mas por vezes ignorado no ambiente escolar, onde aprender os “fatos” substituem o “brincar com os fatos”. Ou seja, a prioridade do conhecimento fica restrita à reprodução escrita das verdades, e não à sua investigação, o que Gardner (1991) já provou como um método defeituoso, pois alunos quando ensinados desta forma não conseguem aplicar conhecimento para resolver problemas ou para entender os fundamentos conceituais de seu campo de aprendizagem.

O artigo segue listando e explicitando os princípios de aprendizagem mais relevantes encontrados em bons jogos digitais. Trataremos a seguir os 16 princípios segundo a visão do autor, dando maior destaque ao princípio das ferramentas inteligentes e conhecimento distribuído, considerado um requisito ao *puzzle dinâmico* descrito na introdução deste trabalho.

- a) Identidade:** Os jogadores herdam um personagem atraente e fortemente formado ou o criam desde o início. Em ambos os casos, estão comprometidos com o novo mundo virtual em que vivem, aprendem e atuam através do seu compromisso com sua nova identificação.
- b) Interação:** Nada acontece até o jogador agir e tomar decisões. A partir de então, o jogo reage, oferecendo *feedback* e novos problemas. Palavras e atos são colocados no contexto de uma relação interativa entre o jogador e o mundo.
- c) Produção:** Os jogadores projetam os jogos digitais pelas ações e decisões tomadas. Muitos destes já vêm com versões do *software* que são feitas para que os jogadores possam modificá-las.
- d) Riscos:** Bons jogos digitais reduzem as consequências das falhas dos jogadores, permitindo que voltem para a última instância antes da derrota. Incentivando-os a correr riscos, explorar e tentar coisas novas.
- e) Customização:** Os jogadores personalizam um jogo para que ele se adapte aos seus estilos de aprendizagem e modo de jogar. Os jogos, muitas vezes, têm diferentes níveis de dificuldade e permitem aos jogadores resolver problemas de diferentes maneiras.
- f) Agência:** Os jogadores têm um senso real de gerenciamento e controle perante as situações mais inusitadas que alguns jogos digitais podem proporcionar.
- g) Boa ordenação dos problemas:** Nos jogos digitais, os problemas enfrentados pelos jogadores são organizados para que os primeiros sejam bem construídos e levem os jogadores a formular hipóteses que funcionem bem na resolução de problemas mais difíceis.
- h) Desafio e consolidação:** Os jogos digitais oferecem aos jogadores um conjunto de problemas desafiadores e em seguida, permitam que resolvem esses problemas até que dominem ou automatizem suas soluções. Só então o jogo lança uma nova classe de problemas.
- i) “Na hora certa” e “a pedido”:** Os jogos digitais quase sempre dão informações verbais exatamente quando os jogadores precisam e podem usá-las, ou seja, quando o jogador sente a necessidade disso, quer e está pronto para fazer bom uso dela.
- j) Sentidos contextualizados:** O jogo digital quase sempre contextualiza os significados das palavras em termos de ações, facilitando a aprendizagem do significado e seu uso no contexto, se comparada com uma definição que explica a palavra em termos de outras palavras.
- k) Frustração prazerosa:** Os jogos digitais são, muitas vezes, percebidos como desafiadores, mas possíveis.
- l) Pensamento sistemático:** Os jogadores são encorajados a pensar sobre a relação entre eventos, fatos e habilidades. Analisando, por exemplo, como cada decisão pode afetar suas ações futuras.

- m)** Explorar, pensar lateralmente, repensar os objetivos: Os jogadores são incentivados a explorar os detalhes antes de ir muito rápido ao prêmio, pensando não apenas de forma linear, e usando essa exploração e pensamento lateral para repensar seus próprios objetivos de tempos em tempos.
- n)** Ferramentas inteligentes e conhecimento distribuído: Os personagens e muitos aspectos do mundo do jogo são de fato “ferramentas inteligentes”, com habilidades e conhecimentos emprestáveis ao jogador. Isto permite um aumento na complexidade dos desafios, visto que o jogo e o jogador atuam juntos no seu enfrentamento. Por exemplo, um jogo de tiro em primeira pessoa com a opção de mira automática permite que o jogador enfrente um número maior de alvos, desviando sua preocupação da mira para outros aspectos como munição, armas, posicionamento. Ou, também, no jogo *Full Spectrum Warrior*, no qual os soldados que o jogador controla sabem como se mover em diversas formações de batalha. Cabe ao jogador as decisões sobre quando e onde chamar cada formação para que os soldados se movam com segurança entre as zonas.
- o)** Equipes transfuncionais: Os jogos digitais, às vezes, favorecem jogar em time, onde cada jogador domina um personagem com diferentes habilidades e que desempenha um papel específico dentro do grupo. Todos realizando funções próprias em busca do bem comum para sua equipe.
- p)** Performance anterior à competência: Os jogadores podem atuar antes de serem competentes, apoiados pelo design do jogo, as “ferramentas inteligentes” oferecidas pelo jogo e muitas vezes também pelo suporte de outros jogadores mais avançados.

O artigo de Nakamura e Csikszentmihalyi (2009), intitulado **The concept of flow**, retrata os resultados sobre uma pesquisa de 25 anos acerca da natureza e condições da felicidade. Para isto foram entrevistados desde jogadores de *Xadrez*, montanhistas, dançarinos e tantos outros que justificam a autossatisfação como principal motivo para continuarem essas atividades. A pesquisa nesta área virou-se principalmente para os jogos, nos quais as observações desta autossatisfação eram evidentes, enquanto em outras áreas onde dinheiro e prestígio podiam ser conquistados, traziam divergências se eram estes os motivos implícitos para a satisfação na atividade.

Chamando *flow* (traduzido como fluxo ou estado de fluxo) um efeito observado naturalmente em muitas situações. Como quando um artista trabalha em uma pintura de forma solitária por muitas horas, ignorando fatores como fome, desconforto e fadiga, até que finalmente complete seu trabalho e rapidamente perca o interesse no produto. Neste contexto, a pesquisa expõe os seguintes elementos observados como necessários para que uma atividade promova o estado de fluxo para aqueles que a realizam:

- a)** Percepção de desafios ou oportunidades de ação em um nível que acompanha as habilidades atuais de quem executa a ação.



- b)** Objetivos claros e próximos de serem alcançados;
- c)** *Feedback* imediato sobre o progresso que está sendo feito.

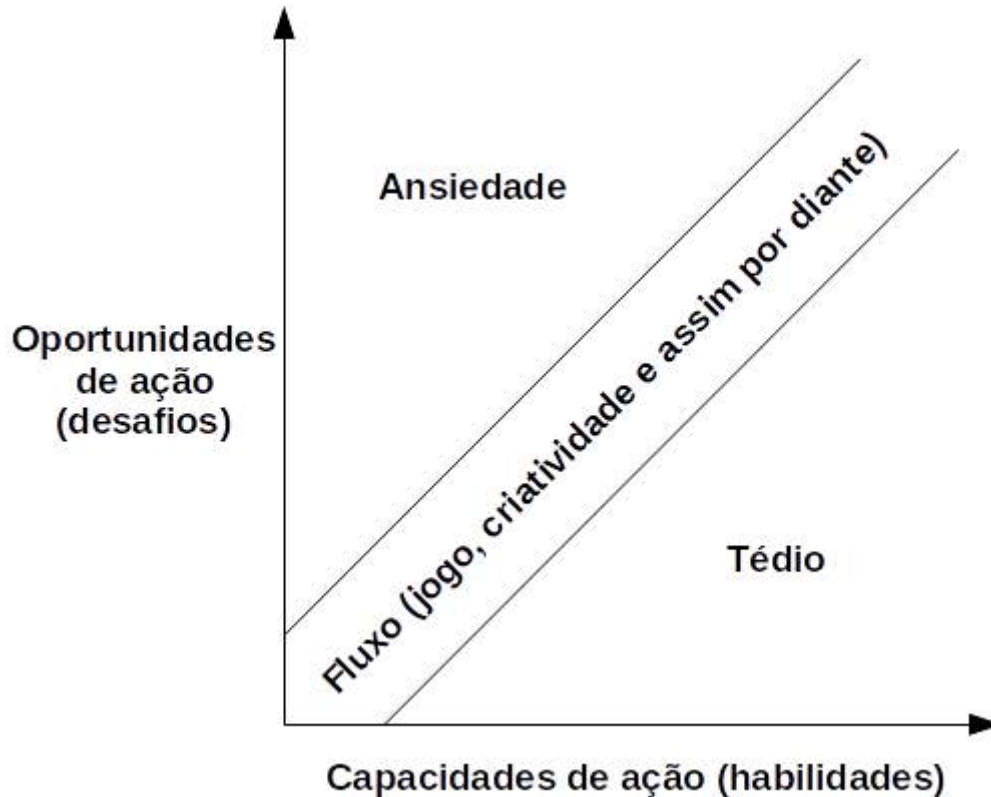
Vários dos entrevistados que experimentaram estados de fluxo confirmam sua experiência subjetiva como uma série de realizações, com um processo de *feedback* contínuo da situação através de ajustes rápidos nas ações mediante os resultados percebidos. Analisando também as características comuns para o estado de fluxo, observou-se que ocorre no sujeito:

- a)** Uma concentração intensa e focada sobre o que está sendo feito naquele momento;
- b)** Suas ações e sua consciência estão fortemente conectadas;
- c)** Perde-se autoconsciência reflexiva (ou seja, a perda de consciência do sujeito como ator social);
- d)** A sensação de que se é capaz de controlar a ação, já que aparentemente se sabe como responder a qualquer situação subsequente;
- e)** A distorção da experiência temporal (geralmente o tempo passa muito mais rápido do que o normal);
- f)** A experiência desta atividade como sendo intrinsecamente gratificante, embora a conquista final seja apenas uma desculpa para continuar a realizá-la.

O estado de fluxo é um equilíbrio frágil, se os desafios começam a exceder suas capacidades, leva o sujeito a apatia ou ansiedade. Mas por outro lado, se suas habilidades começam a superar os desafios, o indivíduo fica apático ou perde a motivação. Um fator de grande interesse neste estudo diz respeito ao estado do fluxo fornecer à pessoa uma experiência otimizada que conseqüentemente reage com uma série de respostas utilizando o máximo de suas capacidades. Seguindo em rápido e progressivo avanço nas habilidades necessárias para enfrentar os desafios propostos. Assim, jogos com a capacidade de levar os jogadores rapidamente a um estado de fluxo, colocam suas faculdades mentais ao máximo para aprenderem a jogar sem que se sintam cansados por isto. Uma característica semelhante que Gee (2005) relaciona presente nos bons jogos digitais, o fato deles serem desafiadores ao mesmo tempo que viciantes, colocando o jogador em uma busca incessante por aprender a jogá-lo sem realmente sentir-se cansado pelo esforço despendido.

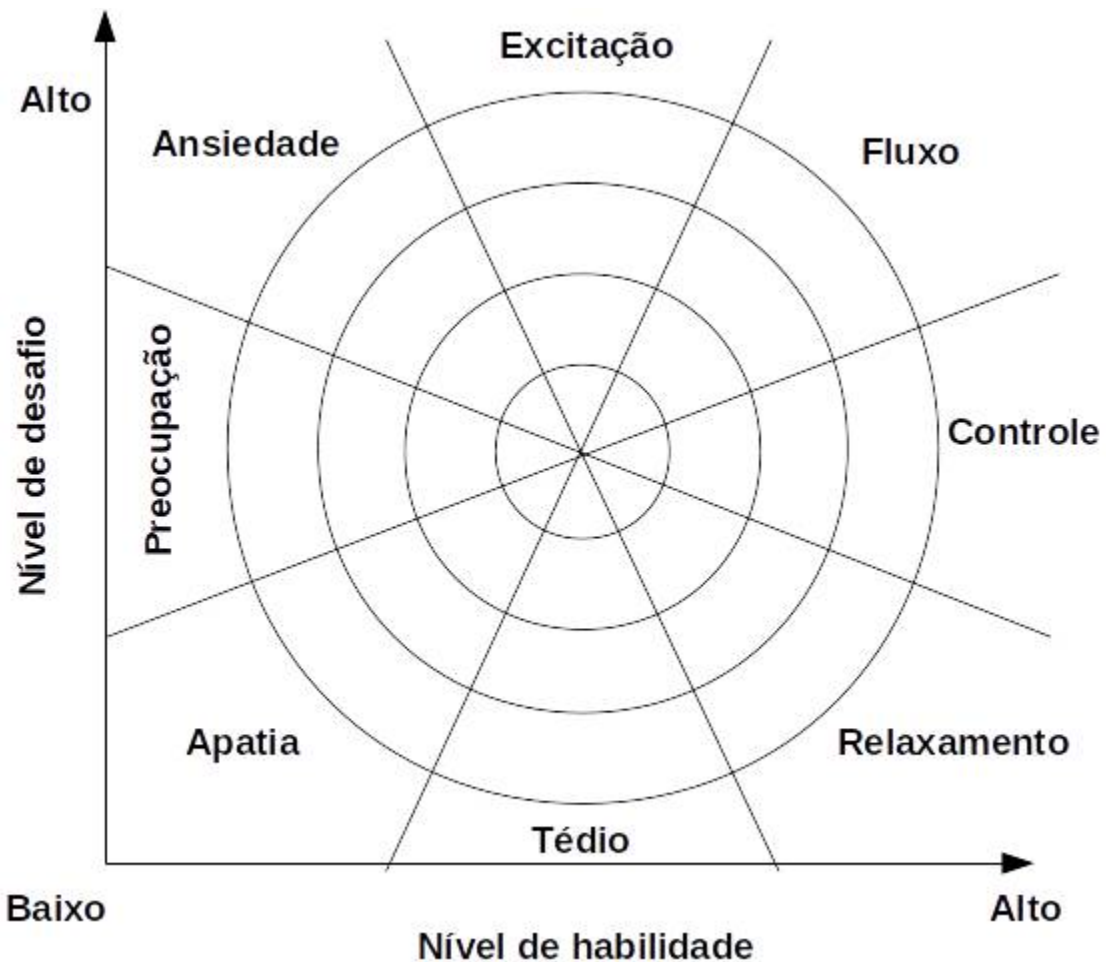
O gráfico a seguir representa uma maneira como este estado de equilíbrio aparece em diversos livros de *game design*, onde o setor superior se refere a ações que vão além das habilidades do jogador, e o setor inferior aos desafios que a habilidade do jogador excede. O estado de fluxo seria a faixa central. Em jogos bem-sucedidos, o jogador enquanto desenvolve suas habilidades, o jogo aumenta na mesma medida os desafios, mantendo-o durante a maior parte de sua interação, neste estado intermediário entre ansiedade e apatia, trazendo assim os benefícios da experiência do

estado de fluxo. Neste gráfico, o jogador inicia o jogo na posição mínima de habilidade e dificuldade. Assim, a medida que o jogo coloca desafios e fornece mecanismos de aprendizagem, as habilidades do jogador se desenvolvem na mesma medida que os desafios a serem enfrentados, permitindo uma aprendizagem de “como jogar” com os benefícios experimentados por quem vivencia um estado de fluxo.



**Figura 15:** Modelo simplificado do fluxo (NAKAMURA, CSIKSZENTMIHALYI, 2009, traduzido)

O gráfico a seguir expõe uma estrutura mais desenvolvida do estado de fluxo, onde temos mais 5 estados (além do estresse, apatia e fluxo). É uma concepção mais acentuada sobre como eles se aproximam ou distanciam-se de acordo com as habilidades e desafios atuais na atividade. Neste gráfico, o estado de fluxo já é considerado uma relação entre a habilidade elevada junto ao desafio elevado. Em contrapartida ao anterior, onde podia-se entender que desafios fracos com uma baixa habilidade também permitiram adentrar o estado de fluxo. Dessa forma, trabalha melhor as divergências de estados mediante as modificações de desafios com habilidades e como estes se relacionam proximalmente.



**Figura 16:** Modelo geral do fluxo (NAKAMURA, CSIKSZENTMIHALYI, 2009, traduzido)

O livro de Bogost, Ferrari e Schweizer (2010), intitulado **Newgames: Journalism at Play**, discute o papel dos jogos, principalmente dos digitais, como mídias da comunicação. Exemplificando a princípio com o caso de pirataria ocorrido na Somália, cujo contexto histórico e social do incidente, proporcionou em pouco tempo a criação de um jogo digital. O jogo neste caso atuava como uma forma de disseminar informações com um grau de interatividade muito maior do que os vídeos ou textos, pois está mais próximo de um modelo simulado da situação do que de uma representação estática do incidente. Esta interatividade do fato com o incidente, se aproxima daquilo que Gee (2005) sugere faltar nos ambientes escolares, onde o “brincar com os fatos” perde lugar para o “aprender os fatos”.

De forma simplória, a maioria das pessoas aprende que Napoleão perdeu a Batalha de Waterloo. Mas seria possível tê-la vencido? E se houvessem mais ataques pelos flancos? E se Napoleão tivesse mais navios? Ou se ele concentrasse toda sua força de ataque em uma única direção? Perguntas assim não fazem parte do ambiente escolar, visto que os fatos aprendidos são estáticos. Porém para uma mesma situação dentro de um jogo, estas são questões pertinentes e que

motivam o estudo sobre diferentes estratégias para aquela situação. Um aspecto que é discutido por estes autores é o registro de eventos histórico de alguns jogos digitais.

Um outro exemplo apresentado sobre o “brincar com os fatos” ligando o jogo a uma mídia de comunicação diz respeito a dinâmica de um documentário, onde o jogo está preocupado em explicar um incidente em detalhes, colocando o jogador em um papel importante para sua realização, como no caso do jogo digital *Revolta da Cabanagem*, desenvolvido no Laboratório de Realidade Virtual (LaRV) da Universidade Federal do Pará em 2011. Nele o jogador participa de um movimento político histórico no Brasil, no qual o jogador lida com pequenas tropas de “cabanos” com armamentos variados que precisam realizar ações fidedignas com a história brasileira, como atacar o regimento do governo e eliminar o comandante José Maria Nabuco de Araújo (<<http://portaldoprofessor.mec.gov.br/fichaTecnicaAula.html?aula=31348>>, acesso em 15/04/2018).



**Figura 17:** Jogo *Revolta da Cabanagem* (fonte portaldoprofessor.mec.gov.br)

A facilidade computacional com que os jogos digitais podem ser criados hoje, permite que os eventos do dia a dia se tornem jogos antes de sair do centro das atenções. Em distinção com o que acontecia no início dos jogos digitais, por exemplo Pacman, que levou 9 meses para ser construído. Os jogos de hoje são construídos em competições que, após o lançamento do tema, os desenvolvedores possuem 48 horas corridas para enviar o produto (exemplo, a Global Game Jam). Assim é possível compararmos o jornal impresso cujas notícias são estáticas, contendo fatos referentes a incidentes de um dia atrás a serem aprendidos pelo leitor, com os jogos que trazem simulações interativas para se “brincar com os fatos” referentes a incidentes de dois dias atrás.

O livro de Juul (2013), intitulado **The art of failure: An essay on the pain of playing videogames**, discute as formas em que o fracasso é visualizado dentro e fora do jogo, pois, enquanto a falha é desagradável fora do jogo, nos jogos o processo de falha é encorajado na maioria

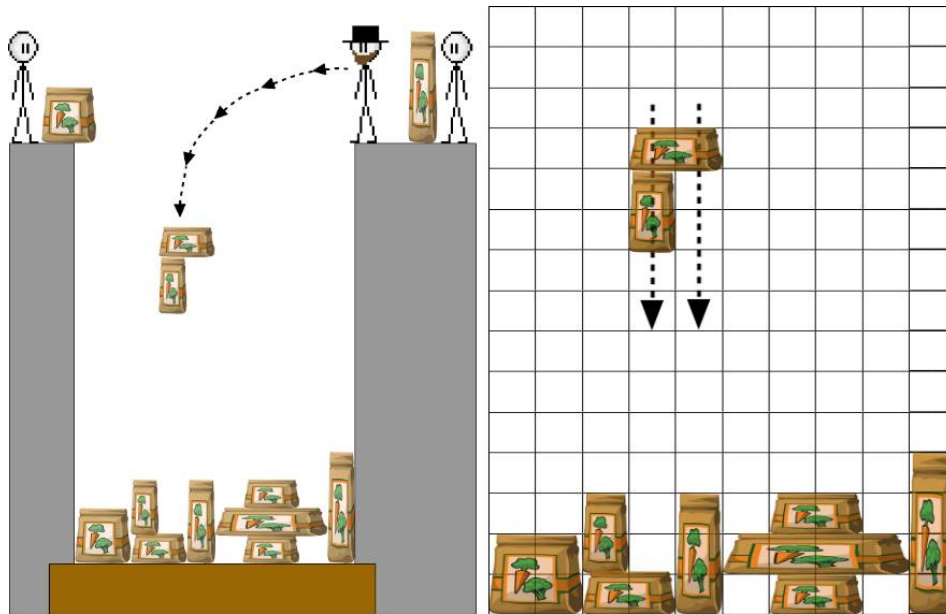
dos contextos relacionados ao aprendizado e até ao desenvolvimento das habilidades nos jogos. De certa forma, o jogador precisa falhar para aprender, ou em outros contextos, a falha no jogo é uma consequência inevitável do ato de jogar.

O autor em muitos dos usos descritos para a falha nos jogos, pode se relacionar com o princípio da aprendizagem descrito por Gee (2005) como riscos. Onde as consequências das falhas são reduzidas, incentivando que os jogadores arrisquem novas experiências e modos de jogar.

Entre os usos do fracasso que segundo o livro podem prejudicar a aprendizagem de como jogar, temos as situações onde a condição de sucesso não está conectada com as habilidades necessárias em seu contexto real. Um exemplo desta situação seria o jogo *Math Madness* (figura 2) apresentado na introdução, onde o jogador somente acerta a bola de basquete na cesta caso resolva as operações Matemáticas. Um indivíduo que goste de jogos digitais de basquete, entenderia seu fracasso ligado a irrelevância do desafio com a ação exigida, visto que nos jogos digitais comuns de basquete, acertar as cestas não tem ligação com resolver operações Matemáticas.

Outro modo abordado sobre como o fracasso pode ser trabalhado nos jogos para melhor imergir os jogadores nas suas ações, é mediante a construção do drama, onde falhar está ligado a formação da história que motivará o personagem do jogador a enfrentar desafios mais sérios e corrigir sua falha. Por exemplo, no começo de *Megaman X* o protagonista enfrenta um rival e é obrigatoriamente derrotado, mas antes de ser destruído, o seu parceiro aparece para salvá-lo. Então próximo ao final do jogo, encontramos o parceiro do personagem preso e este rival novamente aparece. Neste caso temos uma falha ficcional, onde falhar compõe o cenário que o jogo utilizará, apoiando o aumento das habilidades do jogador ao drama da história ligada aos desafios.

O livro de Koster (2014), intitulado **A Theory of Fun for Game Design**, discute que o jogo não é mais do que um presente dado ao cérebro enquanto ele busca padrões difíceis de reconhecer. Em termos gerais, todos os jogos ensinam seus padrões de comportamento. Alguns jogos ensinam relações espaciais como por exemplo, adaptado de uma analogia do próprio autor, um jogo de arremessar sacos de lixo orgânico em um poço ecológico, aqueles que não caem nas fendas começam a se amontoar, demorando mais para se decompor e ameaçando chegar até o topo. Os mecanismos são iguais ao jogo *Tetris*, embora a experiência proposta seja muito diferente. Não é surpreendente que os jogos sejam reduzidos a apenas alguns padrões básicos adaptados de formas criativas.



**Figura 18:** Jogo de arremessar o lixo orgânico/*Tetris* (adaptado de KOSTER, 2014).

Neste sentido, o autor discute que os jogos são ferramentas de aprendizagem, pois os jogadores que procuram avançar em um jogo sempre tentarão otimizar o que estão fazendo, mesmo que isto se pareça com “jogar lixo no poço” ou alinhar peças. A busca pelo caminho ideal independe do contexto e da interação com o jogador pretendida pelo *game designer*. A aprendizagem de como jogar visa tornar as ações tão previsíveis quanto possível. E dependendo de como os jogadores emergirem as incertezas destes, o jogo pode se tornar tão previsível quanto o jogo-da-velha ou tão incerto quanto aqueles de complexidade analítica (COSTYKIAN, 2013). Isto significa que os jogadores tentam obter segurança jogando, mas se esta segurança for realmente atingida, o jogo perderá sua incerteza e ficará desmotivador.

O livro encerra com a discussão sobre como os enigmas nos jogos abordam a complexidade de entender o subjetivo da arte. Porque a arte envolve enigmas a serem desvendados por seus espectadores. Por exemplo, não faz sentido pintar um cachorro se o que queremos que o espectador veja é um cachorro, mais fácil e viável seria levar ao espectador um cachorro. Os jogos assim como as artes não trazem aos seus públicos os sentidos e fatos descritos como se vê na escola. Senão estes não diferenciariam de enciclopédias ou registros estáticos. Assim, os *game designers* de jogos desafiadores enfrentam questões como esta: podemos criar jogos que não têm uma resposta correta? Ou seja, enigmas com desafios reais e complexos para que o cérebro busque seus padrões, da mesma maneira como quando somos instigados por um quadro abstrato repleto de significados.

## 4.4. Guia para game designers

O livro **Design de Games: Uma abordagem prática** de Schuytema (2008), apresenta uma introdução prática e completa ao *game design*, começando por uma exploração do que é um jogo e *game design*. Então coloca o leitor na posição de um *game designer* iniciante, expondo vários papéis e tarefas que são assumidos durante o desenvolvimento de um jogo, a fim de manter a experiência do jogador divertida e viciante, sendo para isto necessário coletar e reter conhecimentos sobre o produto, mover-se de acordo com as tendências e conduzir informações dos desenvolvedores aos jogadores.

O guia dedica parte de seu texto como um tutorial de programação na linguagem *script Lua*. No decorrer das explicações sobre as etapas de produção, exemplifica passo a passo dos elementos de *game design* mediante listas de códigos e *scripts* que compõe um jogo exemplo trabalhado do início ao fim durante o livro. Isso é feito com detalhes práticos de vários problemas de *game design*, como o equilíbrio da mecânica de um jogo, a medida de risco e recompensa, e o escalonamento de desafios. Neste contexto, faz uso do termo átomo de um jogo, para referir-se aos seus menores elementos que quando combinados formam as demais estruturas do sistema.

De modo a lembrar o modelo MDA (HUNICKE, LEBLANC, ZUBEK, 2014), o guia trabalha a percepção e as emoções do jogador por camadas, primeiro o som, depois o movimento e em seguida percepções mais detalhadas/complexas. Também discute como as reações emocionais emergem dessas formas de percepção e de seus objetivos, para então tratar de técnicas e sugestões que mantenham os jogadores ativos/interessados pelo aumento constante de desafios de acordo com suas habilidades. Considerando também períodos de descanso, explorando medidas de riscos e benefícios.

Por fim, aborda a importância do *storytelling* e como isto se relaciona com o significado de jogar, relacionando suas semelhanças, diferindo principalmente no fator passivo (história) e no fator ativo (jogo). Também exemplifica como os átomos podem ajudar na coerência da história, para estar de acordo com o tipo de jogabilidade desejada e reforçando seus objetivos. No geral, o guia apresenta páginas repletas de comentários e experiências dos principais *game designers* da indústria. Além de apresentar ao final de cada capítulo um breve resumo e exercícios ao leitor.

O livro **Level Up! Um guia para o design de grandes jogos** de Rogers (2012), aborda problemas técnicos mas sutis, contudo capazes de comprometer a prática do *game design*. A discussão começa explicando os jogos como atividades com regras e condições de vitória, onde seus objetivos devem ser simples de entender. E embora não se possa garantir que um jogo satisfará

o público, uma estratégia é que o *game designer* com sua experiência de jogador, desenvolva um jogo que o agrade. Assim, provavelmente agradará ao menos aqueles com gostos similares ao seu.

Uma discussão muito particular deste livro se volta ao que o autor chama de três C's, personagem (*character*), câmera, controle. Em relação ao personagem, sua forma deve seguir seus papéis e personalidades no jogo. Também utilizar as medidas do personagem para métrica do jogo, facilita o dimensionamento dos objetos e a interação entre eles. Além de que quando os NPCs (*non-player character*) possuem qualidades em nível de jogadores, estes devem ser aproveitados com ações de jogabilidade. Por exemplo em *Castlevania: Symphony of the Night*, onde o jogador pode ativar a companhia do NPC "Familiar Sword", uma espada flutuante que seguirá seu personagem atacando automaticamente seus inimigos e identificando passagens secretas.

Já em relação à câmera, sua escolha compromete as interações dos jogadores. Usar visão em primeira pessoa permite uma maior imersão para o jogador. Com visão em terceira pessoa, o jogador pode ter uma boa noção do personagem e do mundo. Ver o jogo de topo dá ao jogador o poder de enxergar e planejar as ações do ambiente. Ver o jogo de perfil permite uma maior interação entre os personagens. Assim, a câmera deve ser tratada no jogo como um "detetive" para ajudar o jogador a entender o mundo à maneira como o *game design* quer que percebam.

No que diz respeito aos controles, o autor sugere que o *game designer* mantenha o foco na ergonomia e use estruturas de comandos comuns a jogos do mesmo gênero, pois a familiaridade reduziria a confusão. Seus significados devem estar ligados a simplicidade e ao sentido da ação, por exemplo em um jogo de luta, para golpes com os membros inferiores do corpo temos comandos geralmente localizados na parte inferior do grupo de botões. Assim como o tempo de preparo e posterior a uma ação deve ser bem planejado, pois ainda que sacrifique a lógica da realidade, evita comprometer a jogabilidade de jogos dinâmicos com ações longas ou com reações lentas.

Outros aspectos particulares desta obra, são suas discussões sobre os HUD's (acrônimo de *heads-up display*) que se refere a qualquer elemento visual com a função de comunicar informação ao jogador. Estas informações devem ser intuitivas, facilmente encontradas e rapidamente interpretadas. Por fim, o autor sugere que os *game designers* planejem cada momento do jogo de cima para baixo (mundo → nível → experiência → jogabilidade), que utilizem poucas mecânicas e jogabilidades para o jogo todo, maximizando sua reutilização, e todo o contexto dos níveis deve induzir o jogador ao modo ideal de se jogar, mesmo que na prática isto não ocorra.



## 4.5. Gerenciamento de projetos em game design

O livro **Manual de produção de jogos digitais** de Chandler (2012), apresenta um guia com informações que um produtor, um líder de equipe ou um gerente de estúdio precisa para desenvolver um jogo, do conceito à finalização, pois administrar a produção de um jogo de forma metódica é mais simples, já que o produtor para gerenciar uma equipe de *game design*, se relacionará com os trabalhos de diversos profissionais de arte, engenharia, design e testes. Logo, o autor apresenta e discute sobre os métodos formais para o gerenciamento de projetos de jogos, relacionando-os aqueles já utilizados na indústria de *software*.

O autor divide o processo geral de produção dos jogos em quatro grandes fases. A primeira delas é a pré-produção, que envolve discutir as principais áreas a serem aprovadas pelos investidores antes do projeto iniciar, com elaboração de protótipos, análises de riscos. Nesta fase também se realiza o levantamento dos requisitos do jogo e como afetarão a etapa de construção. A segunda etapa é a produção, nela são implementados os planos da etapa anterior, desenvolvidos *builds* (recursos automáticos para verificar a funcionalidade do jogo) e reavaliação dos requisitos, esta fase dura até a entrega do produto ao cliente. A terceira etapa são os testes, aqui os requisitos são colocados à prova mediante vários critérios, desde o controle de qualidade até a jogabilidade pretendida, esta fase ocorre concomitante à anterior e dura até a entrega do produto ao cliente. A última etapa é a finalização, nela são produzidos os *post-mortems* (termo do latim que significa depois da morte), onde se recapitula sistematicamente aquilo que foi aprendido durante esta experiência, para que os erros não se repitam e os acertos possam ser aproveitados em situações futuras. Também é nesta fase que se faz o arquivamento do material, que quando bem estruturado permite corrigir ou atualizar facilmente a versão atual divulgada.

O livro também aborda aspectos bastante problemáticos ligados a legalização e publicação dos jogos de acordo com seus direitos de uso, tratando também de problemas relacionados a jogos *on-line* multijogador em massa conhecidos como *MMORPG* (vários *MMORPGs* possuem milhares de jogadores interagindo em simultâneo), *voiceovers* (quando os personagens possuem vozes dubladas), música e captura de movimentos (quando os movimentos dos personagens são gerados a partir de movimentos reais realizados), o que não se aplicaria em peso na produção de um *puzzle dinâmico* como proposto na introdução deste trabalho.

## 4.6. Fundamentos do game design

Nos livros de Salen e Zimmerman (2012), **Regras do jogo: fundamentos do design de jogos: volumes 1, 2, 3 e 4**, são discutidos de maneira inusitada os conceitos básicos de *game*

*design*. São detalhados guias para entender os jogos no ponto de vista de estruturas mais simples (sistemas, regras, jogadores, experiências, papel cultural) sendo todos estes baseados e explicitados em referências fortes.

O volume 1 abrange a interação, o design, os sistemas, a interatividade, a definição de jogos e jogos digitais, o círculo mágico (referente ao processo de entrada e saída da realidade proposta por um jogo) e os esquemas primários. Neste volume, são apresentadas sínteses de definições de jogo por oito referências na área (David Parlett, Clark C. Abt, Johann Huizinga, Roger Caillois, Bernard Suits, Chris Crawford, Greg Costikyan, Elliot Avedon e Brian Sutton-Smith). Em seguida, coloca o jogo digital como um subgrupo de jogos, que adiciona nas configurações do jogo outras 4 características especiais de origem computacional: Interatividade imediata, mas restrita; Manipulação da informação; Sistemas complexos e automatizados; Comunicação em rede.

No volume 2, temos a definição de regras. As regras são discutidas então nos três níveis seguintes, expandindo o assunto também para o ambiente de jogo digital:

- a)** Constitutivas: são abstratas, centrais e Matemáticas de um jogo. Contêm a lógica essencial do jogo, elas não indicam explicitamente como os jogadores deveriam pôr essas regras em prática.
- b)** Operacionais: os jogadores as seguem quando estão jogando na prática, dirigindo comportamentos e vistas como necessárias para se jogar.
- c)** Elegantes: regras não explícitas, que se assemelham a uma norma de etiqueta comum a vários jogos, elas permitem que o jogo ocorra sem comportamentos muito adversos mas não regrados.

Continuando o volume, a dinâmica dos jogos é explicada como pertencente aos seguintes sistemas:

- a)** Emergente: jogos que variam a complexidade de seus desafios dentro de quatro níveis: aqueles que as relações dos componentes nunca mudam; mudam de modo periódico; mudam de forma complexa porém dentro de um padrão; mudam através de um comportamento aleatório.
- b)** Incerteza: são sistemas de jogos que utilizam fatores de certeza (resultados predeterminados), risco (resultados dependem de probabilidade conhecida) e incerteza (resultado desconhecido).
- c)** Teoria da informação: jogos onde a medição da incerteza de uma informação é feita a partir da análise de todas as possíveis mensagens e significados de um ato dentro do jogo.
- d)** Informação: refere-se aos conhecimentos ou conteúdos que são manipulados, adquiridos, ocultos e revelados durante o jogo. Em um sistema de informação perfeita nada é escondido de ninguém.
- e)** Cibernético: envolvem jogos com três componentes autorreguladoras, o sensor que mede um aspecto do sistema ou ambiente, o comparador que avalia e decide sua ação e o ativador que reage.

- f)** Teoria dos jogos: um ramo da Matemática que estuda situações de tomada de decisão onde os jogadores são racionais e agem da melhor maneira possível para obter o resultado mais proveitoso.
- g)** Conflito: no sentido de que os jogadores podem ter a eles mesmos como rivais que movimentam a tomada de decisões, ou o sistema que compõe o jogo para ser derrotado coletivamente.

O volume conclui ao discutir a quebra das regras a partir de respostas para as seguintes perguntas retóricas. Quão longe quebrar as regras é permitido? E até onde podemos desrespeitá-los sem quebrá-los?

O volume 3 define a interação dos jogos como experiências de ludicidade e do prazer, perdendo o foco estreito nas regras como um sistema formal para analisar as maneiras pelas quais as regras se tornam uma experiência significativa para os participantes de um jogo, enunciando os seguintes tipos de experiências fornecidas pelos jogos:

- a)** Experiência lúdica: quando um jogador interage com um jogo, o sistema formal se manifesta por meio de efeitos experienciais de realizar uma ação, ter a resposta do jogo e interiorizar as decisões.
- b)** Experiência do prazer: quando proporcionam experiências que são buscadas pela própria causa, ou que afetam a vida de um jogador fora do jogo, ou que só significam dentro do próprio jogo.
- c)** Interação lúdica de significados: os jogos podem representar situações internas na sua estrutura, ou ainda, serem representações no seu próprio formato.
- d)** Interação lúdica narrativa: os jogos contém componentes narrativos gerados previamente ou gerados dinamicamente pelas ações do jogador no jogo gerando sua própria história.
- e)** Experiência da simulação: todos os jogos podem ser entendidos como simulações, até aqueles muito abstratos ou que simulam efeitos não encontrados no mundo real.
- f)** Experiência social: quando as relações entre os elementos da estrutura do jogo são consideradas como relações sociais.

O Volume 4 tem como temas a definição de cultura e jogos. Para isso, as seguintes abordagens culturais utilizadas pelos jogos são discutidas:

- a)** Retórica cultural: todos os jogos refletem a retórica do contexto cultural em que são projetados ou jogados. Tanto em jogos históricos como naqueles da contemporaneidade.
- b)** Cultura aberta: quando apesar da retórica cultural inicial do jogo, os jogadores possuem acesso a seu interior, podendo fazer trocas de significados e adequá-los para sua própria cultura.
- c)** Resistência cultural: quando a tensão entre os jogos e seus contextos culturais são explorados. O foco neste esquema é a interação lúdica para uma transformação cultural.

**d)** Ambiente cultural: focando na ambiguidade entre jogadores e não jogadores, na sobreposição entre jogo e vida real, e no ofuscamento dos espaços de fronteira entre a realidade e o jogo.

Fecha-se o volume com a discussão de que nenhum jogo é uma ilha, pois os jogos são sempre jogados em algum lugar, por alguém, por uma razão ou outra. Eles existem, em outras palavras, dentro de contextos e ambientes culturais. O círculo mágico é o espaço em que as regras assumem um significado especial. Mas o círculo mágico em si existe dentro de uma esfera maior da cultura, onde seus jogadores habitam antes do jogo começar e após seu término.

## 5. Documentos

Distribuímos esta documentação em três seções. Primeiro caracterizamos através de um currículo simplificado o sujeito que formulou os documentos junto com os recursos computacionais utilizados. Então descrevemos de forma sucinta como o diário de *game design* está organizado no apêndice deste trabalho. Também apresentamos uma síntese sobre cada um dos cinco jogos digitais tratados como documentos, junto com o endereço *online* por onde podem ser acessados. Por último, exploramos os objetivos esperados com o desenvolvimento de cada questão Matemática do questionário.

### 5.1. Designer de problemas

Para caracterizar a origem dos documentos desta pesquisa como provenientes a alguém pertencente a população estudada, apresentamos um currículo simplificado que apresenta informações básicas sobre sua formação, experiência em ciências da computação e desenvolvimento de jogos. Também apresentamos os recursos utilizados na construção do *puzzle dinâmico* em estudo.

Minicurriculo referente a fevereiro/2016: Licenciado em Matemática com 3 anos de experiência desenvolvendo e aplicando jogos através de uma metodologia própria. Seu trabalho com jogos digitais para ensino de Matemática ocorria em poucas ocasiões e era sempre individual. Ignorava as teorias do *game design* na prática do desenvolvimento de seus produtos. Possuía conhecimentos medianos na linguagem de programação *Scratch* e pouca experiência nas linguagens de programação C e Matlab, entendendo apenas o básico sobre algoritmos e estruturas de dados.

O sujeito em estudo, dispunha de recursos simplórios para construir um jogo digital, no caso, dois computadores portáteis de pequeno porte (um Toshiba mini NB305-N410 e um ASUS EeePC 1025C Flare, detalhes sobre os aparelhos podem ser encontrados no site <[www.laptopmag.com](http://www.laptopmag.com)> acesso em 12/01/2018) e sem capital para contratar serviços. Todo o processo foi realizado de forma individual, desde a esquematização do jogo, sua arte, programação, divulgação, realização dos testes, organização do material e construção do diário de *game design*. Para entendermos melhor a discrepância entre esta abordagem para aquela comum à indústria de jogos, podemos ver o caso do jogo digital *Stonekeep*, lançado em 1995, que começou com pretensão de gastar 50 mil dólares e sua produção durar 9 meses, mas precisou de 5 milhões de dólares, durou 5 anos e contou com mais de 200 pessoas na equipe (<<http://interplay.wikia.com/wiki/Stonekeep>> acesso em 18/04/2018). Contudo criar um jogo digital é uma atividade cara, mesmo com o atual desenvolvimento tecnológico, os custos necessários para sua construção aumentam cada vez mais,

só em 2007, sua produção já era discutido na faixa de 20 a 100 milhões de dólares, com uma expectativa de retorno entre 250 a 1 bilhão de dólares (ZYDA, 2002). Dessa forma, delineamos o indivíduo como pertencente ao perfil particular que criamos de *designer de problemas*, dos quais, entre suas condições, envolvia a simplicidade dos recursos. Dado que o projeto estudado não contou com financiamentos ou patrocínios de nenhuma natureza, nem mesmo exigiu uma direta aplicação financeira de seu desenvolvedor.

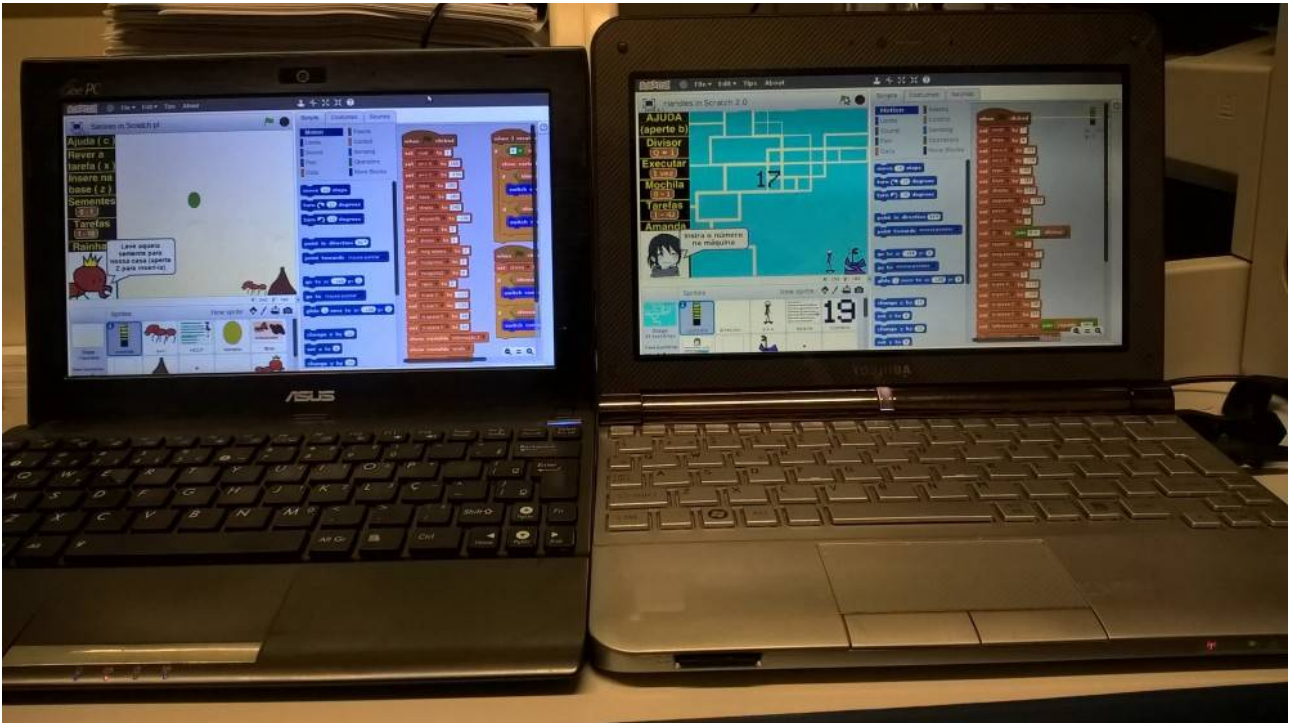
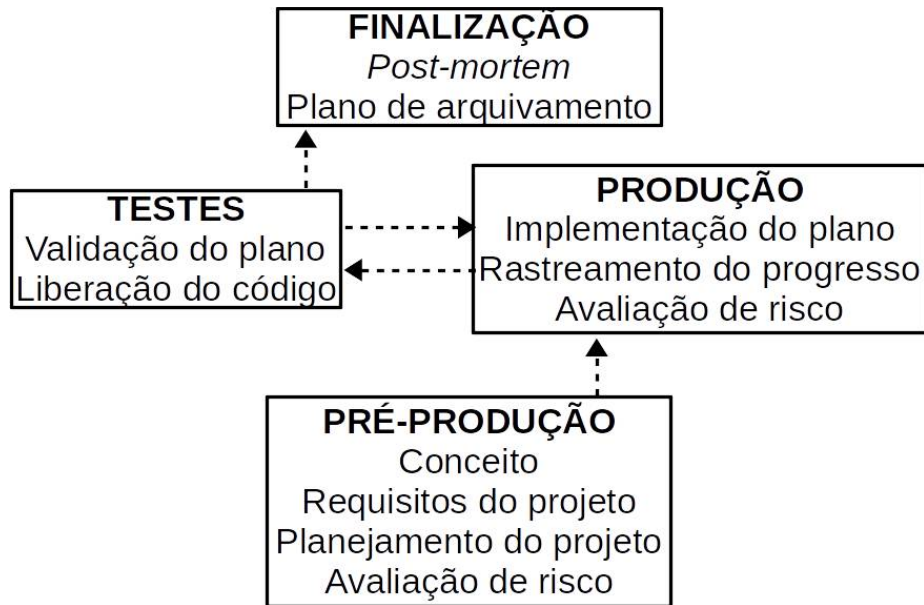


Figura 19: Computadores do *designer de problemas* (fonte própria).

## 5.2. Diário de *game design*

O diário de *game design* narra em 1ª pessoa a história de como cada um dos jogos digitais foi construído a partir dos principais dilemas criativos, técnicos e produtivos vivenciados. Seu texto encontra-se distribuído por jogo, e para cada jogo temos quatro etapas de produção. Antecedendo a primeira etapa o autor procura estabelecer os conflitos e conjecturas que incentivaram seu desenvolvimento. Para isto discute de modo informal o amadurecimento das ideias, o que não pode ser sintetizado em processos datados com início e fim, enquanto que as quatro etapas de produção de cada jogo, se embasam no modelo de organização proposto por Chandler (2012).



**Figura 20:** Modelo de produção para jogos digitais (adaptado de CHANDLER, 2012).

O processo de elaboração deste *puzzle dinâmico* pode ser distribuído de acordo com a versão e a etapa prática de desenvolvimento nos intervalos de mês/ano. Como mostrado na tabela abaixo:

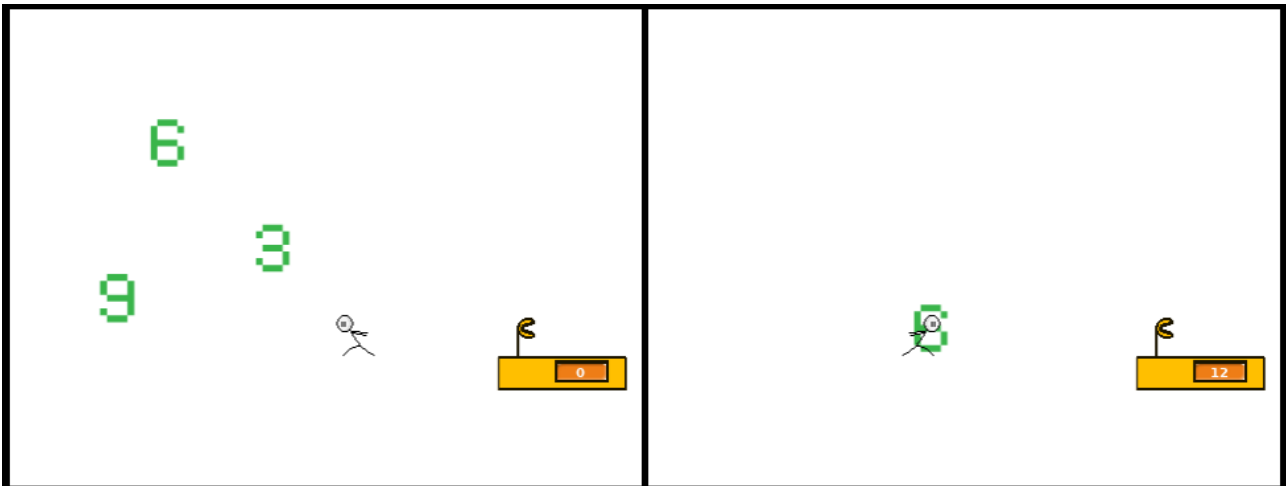
**TABELA 1:** Distribuição das etapas de produção dos jogos digitais (fonte própria).

Nome da versão	Períodos do desenvolvimento (mês/ano)			
	Pré-produção	Produção	Testes	Finalização
Pega e Larga	03/16 – 04/16	04/16	04/16	04/16
Handles	04/16	05/16 – 06/16	05/16 – 06/16	06/16
Handles 2.0	07/16	07/16 – 10/16	10/16 – 11/16	11/16
Handles in Scratch 2.0	12/16 – 01/17	12/16 – 04/17	12/16 – 06/17	07/17
Saúvas in Scratch	12/17	01/18	02/18	-

## 5.2. Jogos digitais

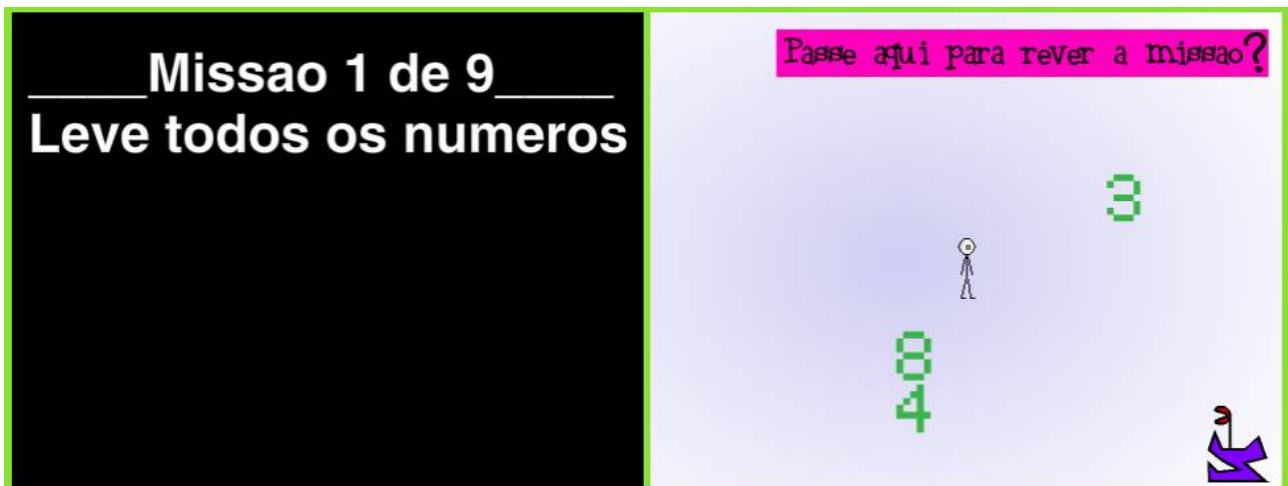
Os jogos digitais considerados como documentos nesta pesquisa, encontram-se hospedados no site <https://scratch.mit.edu> e compartilhados de forma *online* tanto para jogar quanto para ter seu código acessado pelo mesmo site. Abaixo enunciamos os endereços específicos de cada jogo, acessados em 27/01/2016 junto com uma breve explicação sobre seus desafios propostos.

**a) Pega e Larga:** <<https://scratch.mit.edu/projects/200418290/>> acesso em 18/04/2018. Neste jogo, o jogador controla um personagem que precisa buscar números espalhados pela tela e levá-los até uma máquina, que realiza cálculos desconhecidos e mostra apenas seus resultados. Ao final, o jogador é desafiado a descobrir qual cálculo foi realizado.



**Figura 21:** Jogo **Pega e Larga** (fonte própria).

**b) Handles:** <<https://scratch.mit.edu/projects/107763741/>> acesso em 18/04/2018. Neste jogo, o jogador controla um personagem que precisa buscar números espalhados pela tela e levá-los até uma máquina. Se todos e somente os números corretos de acordo com a missão forem inseridos na máquina, o jogador avança ao próximo nível. O jogador pode controlar as condições do jogo para recolher somente os números corretos, que são determinados pelos seus divisores comuns.



**Figura 22:** Jogo **Handles** (fonte própria).

**c) Handles 2.0:** <<https://scratch.mit.edu/projects/132436860/>> acesso em 18/04/2018. Neste jogo, o jogador controla um personagem que precisa buscar números espalhados pela tela e levá-los até uma máquina. Se todos e somente os números corretos de acordo com a missão forem inseridos na máquina dentro de uma ordem apropriada, o jogador avança ao próximo nível. O jogador pode controlar as condições do jogo para recolher somente os números corretos, que são determinados



pelos seus divisores comuns. O jogador também pode construir estruturas para automatizar suas próprias ações no jogo.

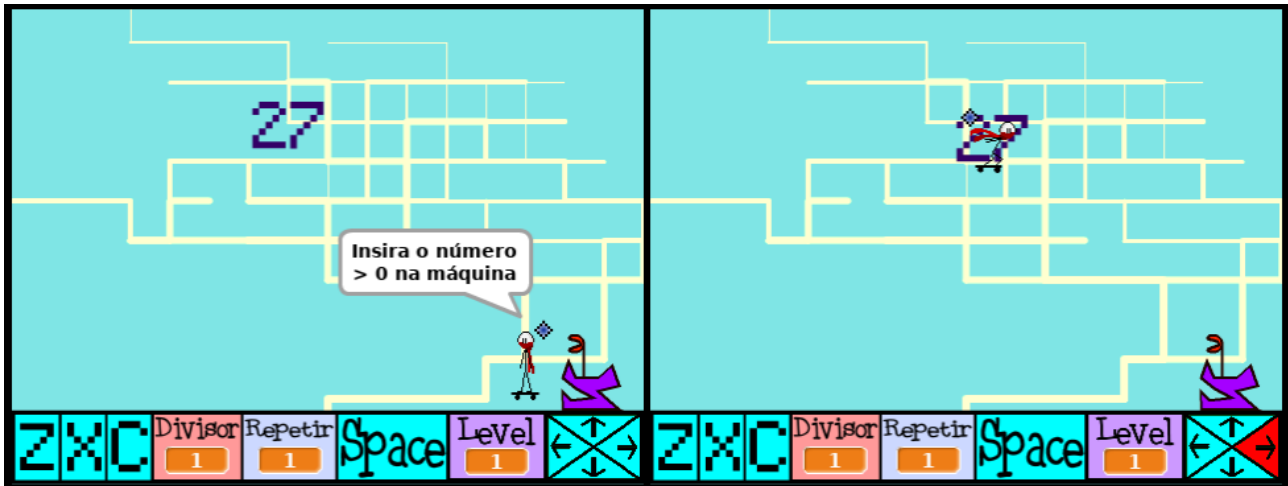


Figura 23: Jogo Handles 2.0 (fonte própria).

**d) Handles in Scratch 2.0:** <<https://scratch.mit.edu/projects/125825356/>> acesso em 18/04/2018. Neste jogo, o jogador controla um personagem que precisa buscar números ou bolinhos espalhados pela tela e levá-los até uma máquina. Se todos e somente os números ou bolinhos corretos de acordo com a missão forem inseridos na máquina dentro de uma ordem apropriada, o jogador avança ao próximo nível. O jogador pode controlar as condições do jogo para recolher somente os números corretos, que são determinados pelos seus divisores comuns. Pode selecionar a quantidade de bolinhos que conseguirá segurar ao mesmo tempo e interagir com ferramentas do jogo que analisam e comparam propriedades destes objetos. O jogador também pode construir estruturas para automatizar suas próprias ações no jogo.

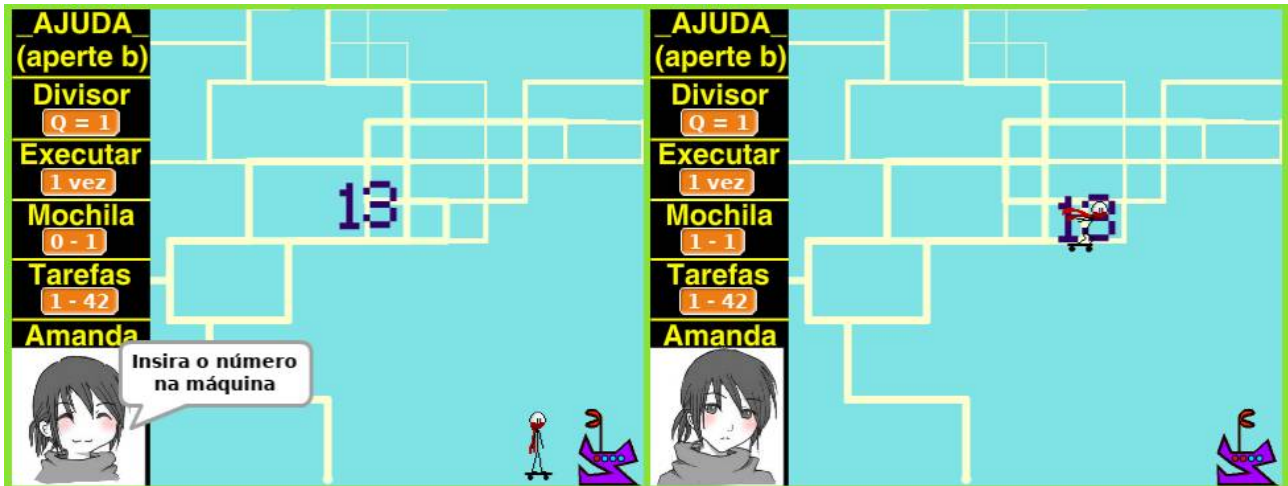


Figura 24: Jogo Handles in Scratch 2.0 (fonte própria).

e) **Saúvas in Scratch:** <<https://scratch.mit.edu/projects/195779695/>> acesso em 18/04/2018. Neste jogo, o jogador controla uma formiga que precisa buscar sementes espalhadas pela tela e levá-las até um formigueiro. Se todas e somente as sementes corretas de acordo com a missão forem inseridas no formigueiro, o jogador avança ao próximo nível. O jogador pode controlar as condições do jogo para recolher somente as sementes corretas, que são determinadas pelos divisores comuns dos dias restantes para sua germinação.

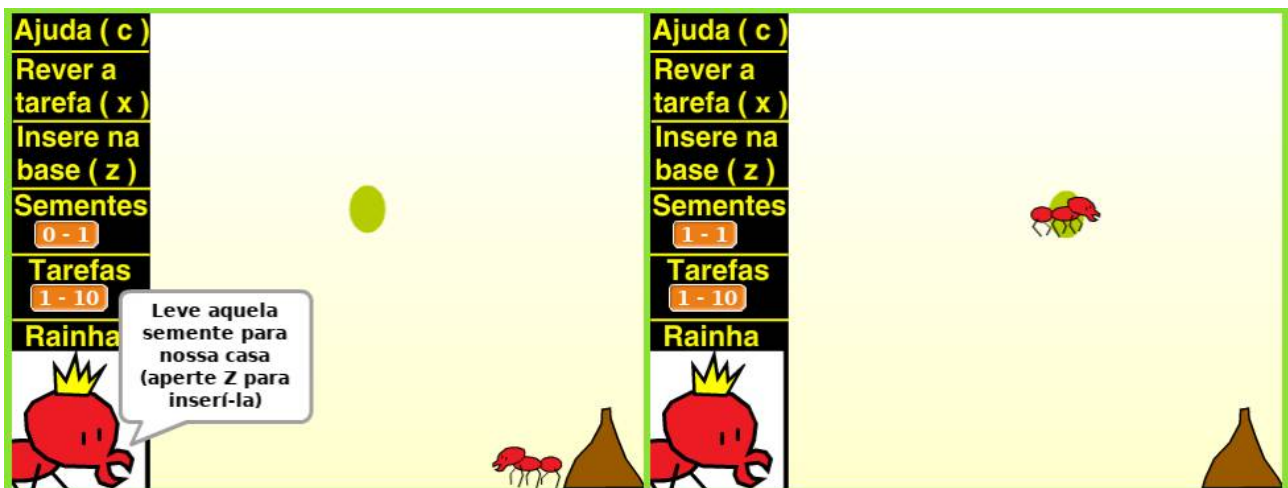
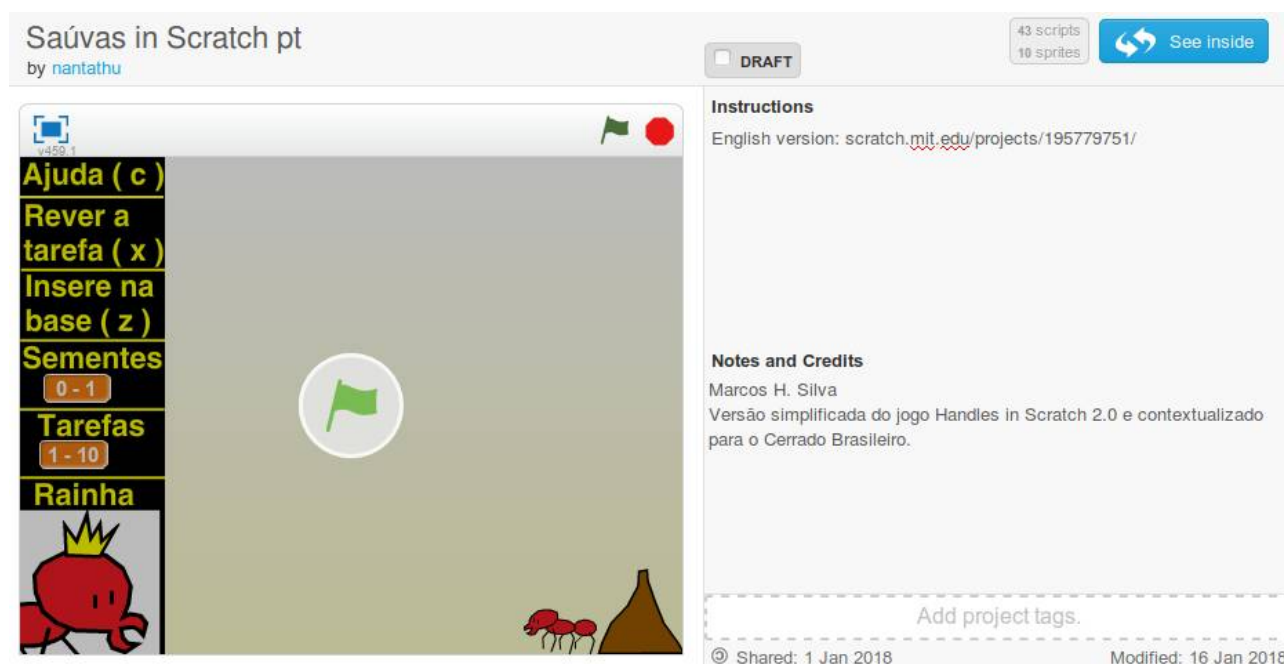


Figura 25: Jogo Saúvas in Scratch (fonte própria).

Para os cinco jogos, após acessar o endereço haverá uma janela cinza com uma bandeira verde ao centro. Clicando nela o jogo inicia. No canto superior direito desta janela, há uma bandeira verde que quando clicada reinicia o jogo e um octógono vermelho que quando clicado interrompe a execução do jogo. Ao lado da janela cinza, tem um quadro de instruções e créditos à produção.

Acima deste quadro há um botão chamado *See inside*. Quando clicado, permite que qualquer pessoa, mesmo sem uma conta no site, acesse e edite uma versão própria do jogo.



**Figura 26:** Tela dos jogos dentro do site *Scratch* (fonte própria).

## 5.4. Questionários

Um dos métodos para coletar dados dos jogadores de *Handles in Scratch 2.0* valeu-se de dois questionários. O primeiro a ser respondido antes de jogar e o outro logo após jogar por um período máximo de 60 minutos. Cada questionário conta com 10 questões objetivas de Matemática, podendo ter várias alternativas corretas ou nenhuma, 6 questões de escala *likert*, 4 questões dissertativas. Elas estão organizadas da seguinte maneira, questões 1 até 20, referentes ao questionário anterior ao jogo, e questões 21 até 40 referentes ao questionário posterior ao jogo. Para efeito de consulta, ambos os questionários encontram-se disponíveis nos apêndices 2 e 3 respectivamente.

As questões de Matemática de um questionário são análogas as do outro, variando poucas informações, mas preservando o raciocínio necessário para sua resolução. Elas foram agrupadas dentro de três grupos explicados seguir.

**a) Divisibilidade prática:** identificação dos números com um determinado divisor comum a partir da aplicação do algoritmo da divisão inteira para cada caso. Correspondendo as questões 1, 2, 3, 4, 21, 22, 23 e 24.

Relação com o jogo: desde o nível 1, o divisor encontra-se em 1 selecionando todos os números, ainda que o jogador desconheça esta propriedade; a partir do nível 8, todos os desafios

exigem manipular a propriedade de seleção dos números, que ocorre escolhendo um valor e agarrando os números divisíveis por este; a partir do nível 10, surgem desafios de levar os múltiplos, incluindo o 0 entre os números corretos; a partir do nível 15, surgem desafios de levar apenas os números 1 da tela, colocando o jogador frente ao problema de que o divisor 1 agarrará tudo sem distinção; os níveis 18 e 19 envolvem a seleção dos pares e ímpares respectivamente;

**b) Divisibilidade conceitual:** reconhecimento da impossibilidade de divisão por zero, e que o zero seja o único número natural divisível por outro maior que ele mesmo. Correspondendo as questões 5, 6, 25 e 26.

Relação com o jogo: o método univalente para soltar números neste jogo é escolher os números cujo divisor comum é 0; a partir do nível 10, surgem desafios de levar os múltiplos, incluindo o 0 entre os números corretos; os níveis 13, 19, 21, 24 e 25 não podem ser resolvidos a menos que o jogador mude o critério do divisor comum para 0 a fim de soltar os números 0; o nível 20 envolve a seleção dos números 0 entre outros números errados.

**c) Resolução iterativa:** conhecer o sistema de medida de tempo (dia, hora e minuto) e realizar operações de adição e subtração dentro de um sistema cíclico. Correspondendo as questões 7, 8, 9, 10, 27, 28, 29 e 30.

Relação com o jogo: resolver os níveis envolve aplicar um algoritmo eficaz, que as vezes pode ser iterativo, como selecionando número por número seja visualmente ou através da propriedade de divisão comum; o cenário bidimensional com os lados opostos (horizontais e verticais) “costurados”, representam a movimentação do personagem sobre a superfície de um toro, um sólido geométrico que permite o personagem se mover de forma cíclica pela tela; a partir do nível 18, os números facilmente se amontoam, sendo conveniente prever seus comportamentos e testá-los na máquina para cada divisor comum escolhido no lugar de tentar separá-los visualmente; dos níveis 22 ao 25, a única forma de selecionar os números é prevendo seus comportamentos e testá-los na máquina para cada divisor comum escolhido, dado que eles são visualmente indistinguíveis.

As questões pessoais de escala *likert* e dissertativas possuíam o intuito de analisar aspectos particulares nos perfis dos jogadores e de suas percepções sobre o jogo. Seu uso tinha duas finalidades, a primeira envolvia contribuir no desenvolvimento do produto, e a segunda envolvia proteger o jogo de uma modificação desnecessária dado por um possível resultado negativo nas questões objetivas de Matemática.

No questionário anterior ao jogo, o interesse era determinar se os participantes: gostam de jogos digitais (questão 11); possuem círculos de amizade relacionados aquilo que jogam (questão 12); gostam de estudar em casa ou frequentar o ambiente escolar (questão 13); possuem aceitação familiar para jogarem em casa (questão 14); gostam de enfrentar desafios novos em jogos (questão 15); gostam de criar modos diferentes de jogar (questão 16). E também reunir dados sobre: suas preferências por jogos digitais (questão 17); as características que mais valorizam nos seus jogos digitais favoritos (questão 18); a disciplina escolar favorita (questão 19); comentários abertos que não puderam ser cobertos por nenhuma questão anterior (questão 20).

No questionário posterior ao jogo, o interesse era determinar se os participantes em relação ao jogo: o acharam divertido (questão 31); gostaram do modo de selecionar os números (questão 32); consideram ter um bom desempenho (questão 33); tiveram facilidade para aprender como jogar (questão 34); sentiram-se incentivados a jogar pelo mecanismo de *feedback* (questão 35); gostaram de quando o mecanismo de *feedback* apresentava um resultado positivo sobre suas ações (questão 36). E também reunir dados relacionados ao jogo sobre: as características que consideraram mais positivas (questão 37); as características que consideraram mais negativas (questão 38); sugestões de mudanças (questão 39); comentários abertos que não puderam ser cobertos por nenhuma questão anterior (questão 40).

## 6. Educação Matemática

Neste capítulo trataremos em um aspecto amplo as possibilidades de uso destes *puzzles dinâmicos* para o ensino de Matemática na Educação Básica ou Ensino Superior. Também discutiremos a eficiência do *puzzle dinâmico* criado mediante os próprios testes realizados pelo *designer de problemas* durante seu processo de construção, e relacionamos aqui os aspectos da literatura sobre *game design* e sua contribuição nas experiências do *designer de problemas* com este *puzzle dinâmico* baseado no diário de *game design*.

### 6.1. Matemática implícita

As cinco versões deste jogo possuem ao todo 83 níveis (Pega e Larga → 1 nível; Handles → 9 níveis; Handles 2.0 → 21 níveis; Handles in Scratch 2.0 → 42 níveis; Saúvas in Scratch → 10 níveis). A partir da experiência em jogar todos os níveis destes jogos, traçamos os aspectos comuns a cada desafio e os generalizamos dentro de dez problemas matemáticos. Para auxiliar o leitor na compreensão dos problemas, apresentamos abaixo cada problema exemplificado também em situações de jogo.

Problema 1: Sejam  $a, b, c, d$  números naturais conhecidos e  $\#$  uma operação Matemática desconhecida dentro destas três opções: multiplicação; subtração; adição. Dado três resultados:  $a\#b$ ;  $(a\#b)\#c$ ;  $((a\#b)\#c)\#d$  como podemos determinar qual operação está sendo realizada?

Problema no jogo: Sejam  $a, b, c, d$  tal que  $a=0, b=0, c=3, d=9$  e  $\#$  uma operação Matemática desconhecida dentro destas três opções: multiplicação; subtração; adição. Dado três resultados:  $a\#b=0$ ;  $(a\#b)\#c=3$ ;  $((a\#b)\#c)\#d=12$ , como podemos determinar qual operação está sendo realizada?

Resolução: De  $a\#b=0$  nada podemos determinar, mas  $(a\#b)\#c=3$  nos indica que  $\#$  não pode ser multiplicação, senão  $(a*b)*c=0$ , também não pode ser subtração, senão  $(a-b)-c=-3$ , mas pode ser adição, pois  $(a+b)+c=3$ . Não havendo outras alternativas,  $\#$  é a operação de adição.

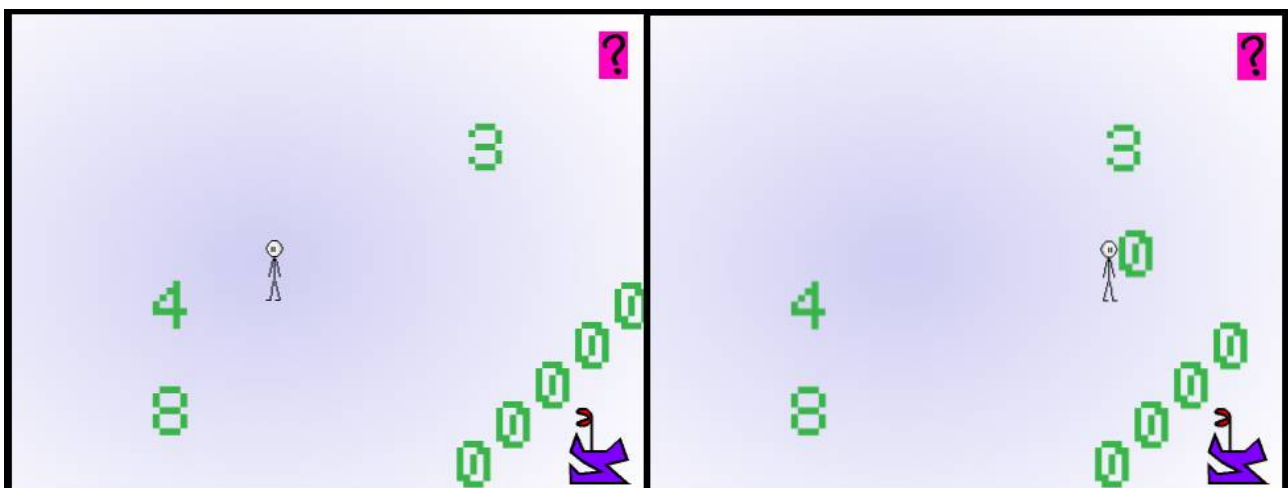


**Figura 27:** Situação do problema 1 (fonte própria).

Problema 2: Suponha que a movimentação no plano cartesiano ocorra a partir de distâncias inteiras e somente em direções horizontais e verticais. Foram espalhados no plano  $N$  bloqueios puntiformes, todos eles ocupando posições inteiras do plano cartesiano. Partimos da posição  $(x, y)$  sendo  $x$  e  $y$  números naturais tais que  $x+y>0$ . Como podemos abrir caminho até a posição  $(0,0)$ ?

Problema no jogo: Considerando no plano cartesiano apenas o eixo negativo de  $x$  e o eixo positivo de  $y$ . O personagem começa na posição  $(-10, 6)$  e temos 5 bloqueios nas posições  $(-4, 0)$ ,  $(-3, 1)$ ,  $(-2, 2)$ ,  $(-1, 3)$ ,  $(0, 4)$ .

Resolução: Removendo o bloqueio da posição  $(0,4)$  temos um caminho até a posição  $(0,0)$ .



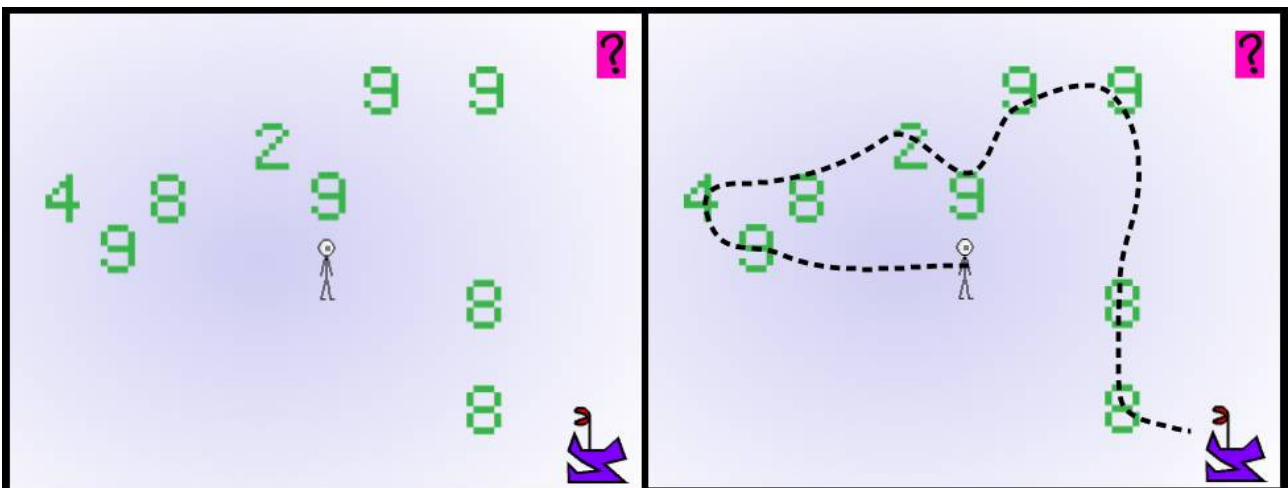
**Figura 28:** Situação do problema 2 (fonte própria).

Problema 3: Seja  $N$  a quantidade de objetos com posições conhecidas espalhados no plano cartesiano. Cada objeto deve ser levado até a posição  $(0,0)$  e não há limite de quantos objetos podem ser carregados ao mesmo tempo. Porém a velocidade de movimentação inicial é  $x$

unidades/s, e de forma acumulativa para cada objeto carregado, a velocidade reduz em  $y$  unidades/s. Sendo  $M$  um número real, como buscar todos os objetos em um tempo menor que  $M$  segundos?

Problema no jogo: Considerando no plano cartesiano apenas o eixo negativo de  $x$  e o eixo positivo de  $y$ . O personagem começa na posição  $(-10, 6)$  e temos objetos nas posições  $(-15, 7)$ ,  $(-14, 6)$ ,  $(-13, 7)$ ,  $(-12, 8)$ ,  $(-10, 7)$ ,  $(-8, 10)$ ,  $(-4, 10)$ ,  $(-4, 4)$ ,  $(-4, 2)$ . Nossa velocidade inicial é de 3 unidades/s e para cada objeto carregado sofremos um atraso de 0,2 unidades/s. Como podemos buscar todos os objetos em um tempo inferior a 100 segundos?

Resolução: Começaremos calculando a distância total a ser percorrida em uma determinada trajetória:  $(-10, 6)$  até  $(-15, 7) = 6$  unidades;  $(-15, 7)$  até  $(-14, 6) = 2$  unidades;  $(-14, 6)$  até  $(-13, 7) = 2$  unidades;  $(-13, 7)$  até  $(-12, 8) = 2$  unidades;  $(-12, 8)$  até  $(-10, 7) = 3$  unidades;  $(-10, 7)$  até  $(-8, 10) = 5$  unidades;  $(-8, 10)$  até  $(-4, 10) = 4$  unidades;  $(-4, 10)$  até  $(-4, 2) = 8$  unidades;  $(-4, 2)$  até  $(0, 0) = 6$  unidades. Total da distância nesta trajetória = 38 unidades. Suponha um caso extremo onde o personagem começa sua trajetória com a todas as penalidades de velocidade, ou seja,  $(3 - 1,8)$  unidades/s. Isto deixaria sua velocidade inicial em 1,2 unidades/s. Então nesta trajetória ele gastaria 31,6 segundos. Como a situação real é melhor do que este caso extremo, na trajetória proposta, gastaríamos menos de 31,6 segundos.



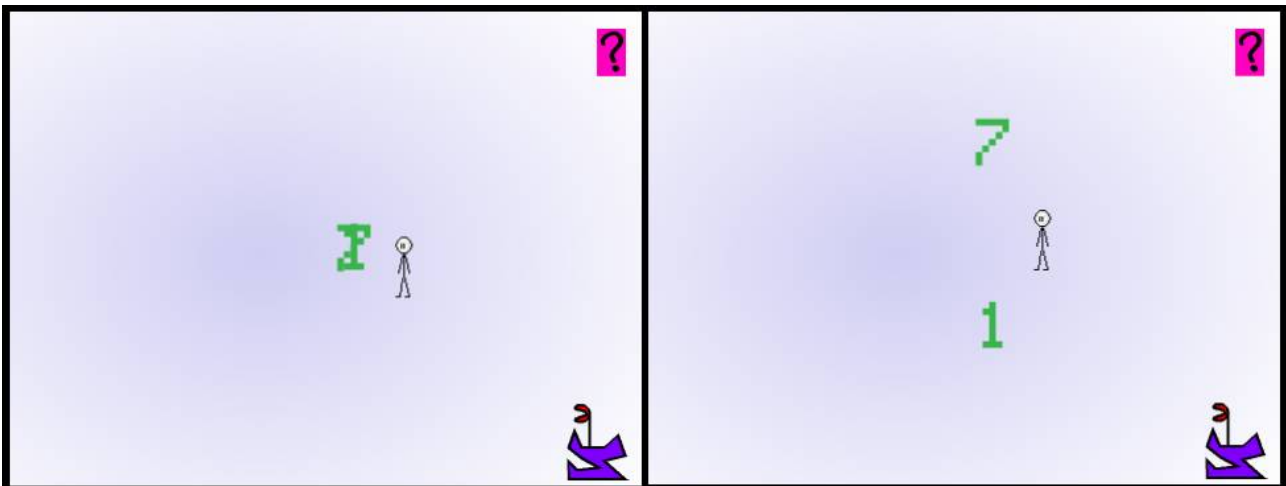
**Figura 29:** Situação do problema 3 (fonte própria).

Problema 4: Seja  $N$  a quantidade de cartas em uma pilha, marcou-se uma quantidade arbitrária de cartas. Quantas cartas precisamos retirar da pilha para garantir que todas as cartas marcadas saíam?

Problema no jogo: Seja 2 o número de cartas em uma pilha, marcou-se uma quantidade arbitrária de cartas com o número 1. Quantas cartas precisamos retirar da pilha para garantir que todas as cartas marcadas saíam?



Resolução: Duas cartas, pois como a marcação foi arbitrária, a carta que ocupa a base da pilha poderia também estar marcada.

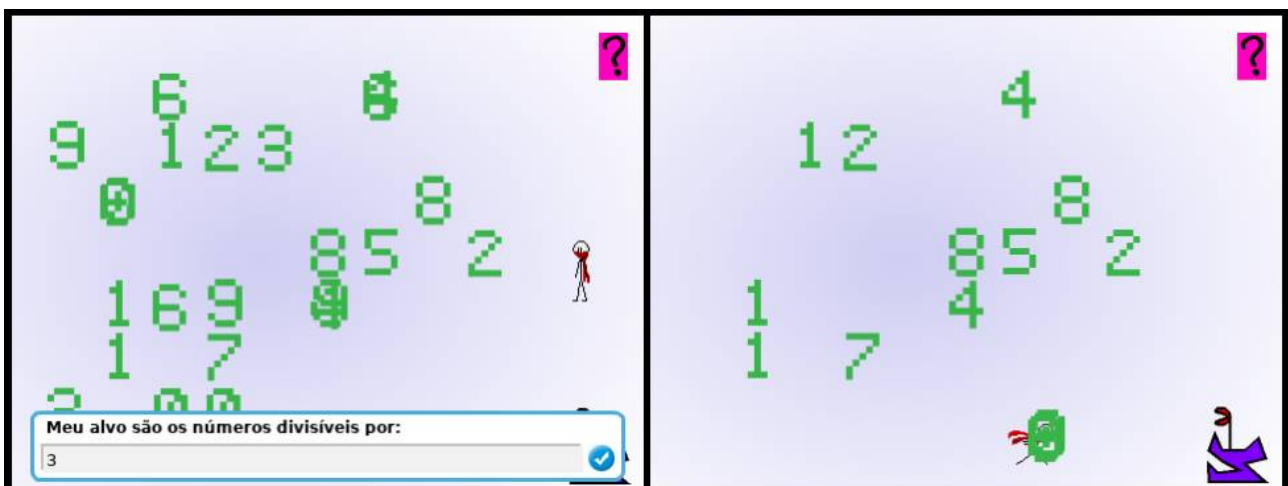


**Figura 30:** Situação do problema 4 (fonte própria).

Problema 5:  $Z$  é um conjunto composto por números naturais. Podemos criar ou mesclar conjuntos transferindo seus elementos do conjunto com um divisor comum escolhido. Seja  $Y$  um subconjunto de  $Z$ , quais transferências devemos fazer para que  $Y$  deixe de ser um subconjunto?

Problema no jogo:  $Z$  é o conjunto formado por números naturais menores do que 10. Podemos criar ou mesclar conjuntos transferindo seus elementos do conjunto com um divisor comum escolhido. Seja  $Y$  o subconjunto de  $Z$  formado por  $\{0, 3, 6, 9\}$ , quais transferências devemos fazer para que  $Y$  deixe de ser um subconjunto?

Resolução: Devemos transferir uma vez todos os elementos de  $Z$  com divisor comum 3 para um novo conjunto.



**Figura 31:** Situação do problema 5 (fonte própria).

Problema 6: Suponha que um segmento retangular do plano cartesiano variando de  $(a, b)$  até  $(c, d)$ , com  $c > a$  e  $d > b$  teve suas bordas conectadas com seus lados opostos formando a superfície de um toro (um sólido geométrico similar a uma rosquinha ou uma câmara de pneu). E que a movimentação ocorra a partir de distâncias inteiras e somente em direções horizontais e verticais. Foram espalhados no plano  $N$  bloqueios puntiformes, todos eles ocupando posições inteiras do plano cartesiano. Partimos da posição  $(x, y)$ , existe um objeto na posição  $(w, z)$  e nossa base fica na posição  $(k, j)$  sendo  $x, y, w, z, k, j$  números inteiros ocupando posições desocupadas deste toro, como determinar se existe um caminho desbloqueado ligando  $(x, y)$ ,  $(w, z)$ ,  $(k, j)$ ?

Problema no jogo: Seja  $a=-7, b=0, c=0, d=4$ . O personagem começa na posição  $(-1, 0)$  e foram colocados 5 bloqueios puntiformes nas posições  $(-2, 0), (-2, 1), (-2, 2), (-2, 3), (-2, 4)$ . Como determinar ao menos um caminho desbloqueado ligando um objeto na posição  $(-3, 3)$  até a base na posição  $(0, 0)$ ?

Resolução: Avance até a posição  $(-1, 3)$ , então avance na direção  $(0, 3)$  e continue em frente até a posição  $(-3, 3)$ . Vá para a posição  $(-3, 2)$ , então avance na direção  $(-7, 2)$  e continue em frente até a posição  $(0, 2)$ . Vá para a posição  $(0, 0)$ .

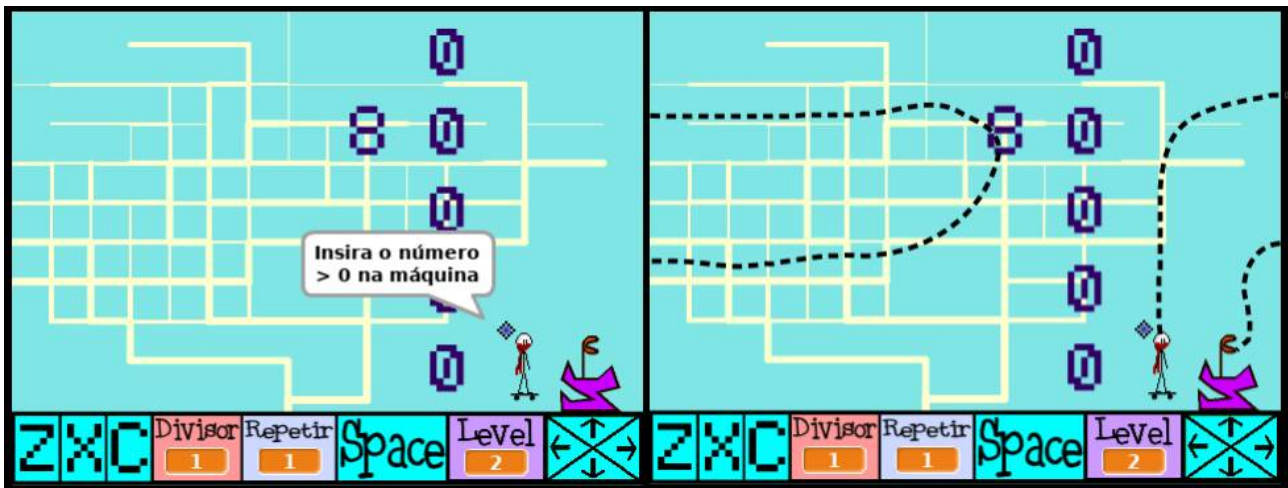


Figura 32: Situação do problema 6 (fonte própria).

Problema 7:  $Z$  é um conjunto composto por números naturais. Podemos criar ou mesclar conjuntos transferindo elementos de subconjuntos com um divisor comum escolhido. Seja  $Y$  um subconjunto de  $Z$ , quais transferências devemos fazer para que  $Y$  deixe de ser um subconjunto?

Problema no jogo:  $Z$  é o conjunto formado por números naturais menores do que 30. Podemos criar ou mesclar conjuntos transferindo seus elementos do conjunto com um divisor comum escolhido. Seja  $Y$  o subconjunto de  $Z$  formado por pelos números zeros, quais transferências devemos fazer para que  $Y$  deixe de ser um subconjunto?

Resolução: Podemos escolher o divisor comum 6 e percorrer somente o subconjunto que não possuem divisíveis por 6 maiores do que 0.

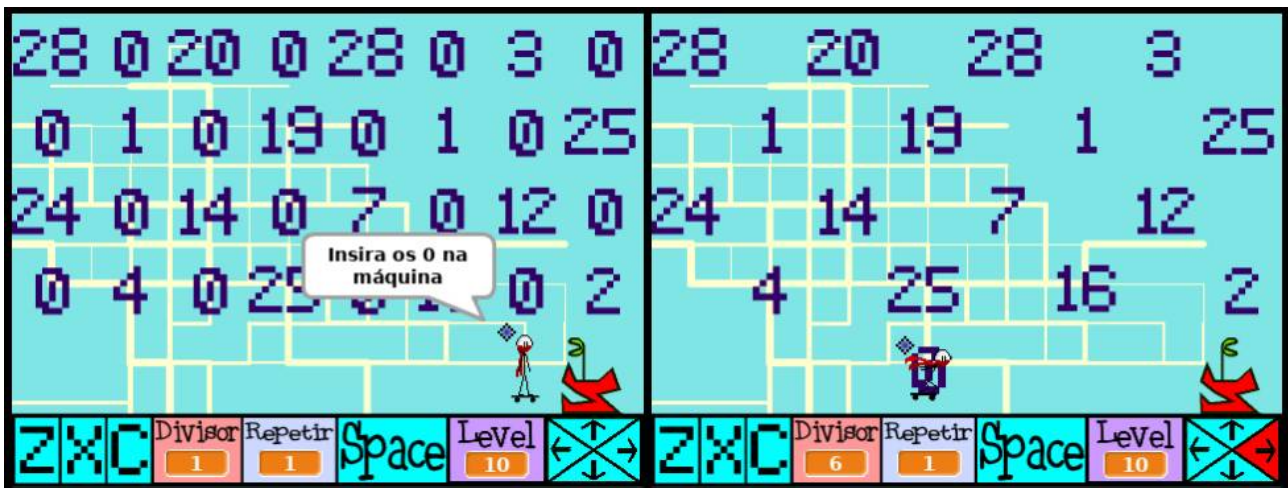


Figura 33: Situação do problema 7 (fonte própria).

Problema 8:  $Z$  é um conjunto composto por números naturais. Podemos criar ou mesclar conjuntos transferindo seus elementos do conjunto com um divisor comum escolhido. Sejam  $Y$  um subconjunto de  $Z$ , e sejam  $y_1, \dots, y_n$  subconjuntos de  $Y$ , sendo  $n$  um número natural, quais transferências devemos fazer para que  $y_1, \dots, y_n$  deixem de ser subconjuntos?

Problema no jogo:  $Z$  é o conjunto formado por números naturais menores do que 30. Podemos criar ou mesclar conjuntos transferindo seus elementos do conjunto com um divisor comum escolhido. Seja  $Y$  o subconjunto de  $Z$  formado por pelos números divisíveis por 9, quais transferências devemos fazer para que os subconjuntos de  $Y$ ,  $\{0\}$ ,  $\{9\}$ ,  $\{18\}$ ,  $\{27\}$  deixem de ser subconjuntos?

Resolução: Podemos escolher um divisor qualquer superior a 30 e transferir de  $Z$  para um novo conjunto os números  $\{0\}$ , pois 0 é o único número natural divisível por qualquer número maior do que ele mesmo. Então podemos transferir de  $Z$  para um novo conjunto, os números com divisor comum 27. Em seguida transferir de  $Z$  para um novo conjunto os números com divisor comum 18. Por último, transferir de  $Z$  para um novo conjunto os números com divisor comum 9.



Figura 34: Situação do problema 8 (fonte própria).

Problema 9: Seja  $Z$  um conjunto com  $N$  objetos com os valores ocultos 0 e 1. Podemos conferir se quaisquer subconjuntos de  $Z$  possui ao menos um objeto com valor 1. Sabemos que existem  $N$  objetos com valor 1 e  $M$  objetos com valor 0, quais verificações precisamos fazer para localizar todos os objetos com valor 1?

Problema no jogo: Seja  $Z$  um conjunto com 2 objetos com valores ocultos 0 e 1. Podemos conferir se quaisquer subconjuntos de  $Z$  possui ao menos um objeto com valor 1. Sabemos que existe 1 objeto com valor 1 e 1 objeto com valor 0, quais verificações precisamos fazer para localizar todos os objetos com valor 1?

Resolução: Precisamos somente verificar um objeto, pois se ele tiver valor 1, é o próprio. Caso contrário, o objeto que não verificamos deverá ter valor 1.

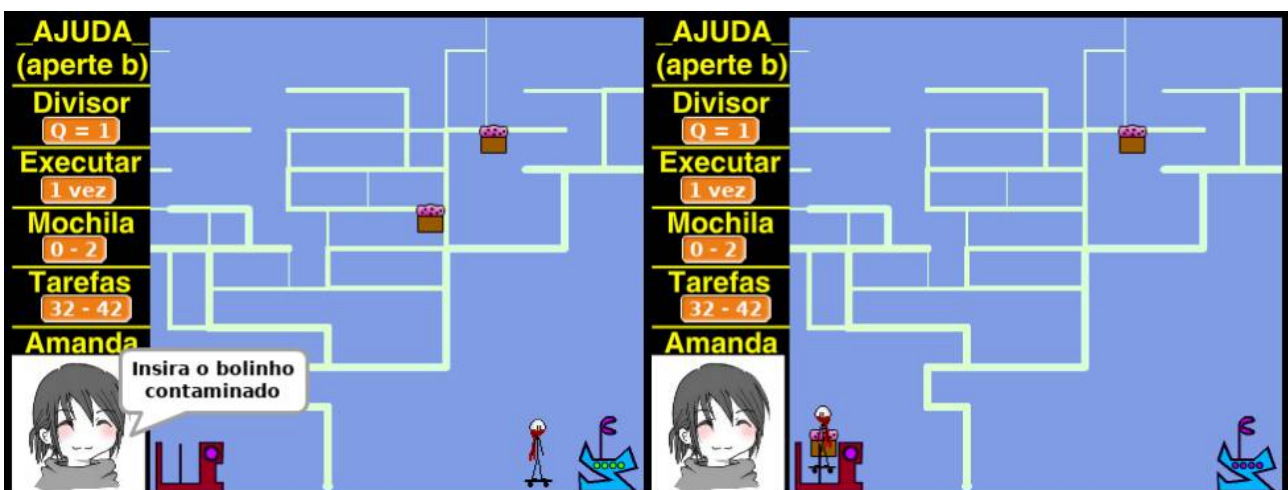


Figura 35: Situação do problema 9 (fonte própria).

Problema 10: Seja  $Z$  um conjunto com  $N$  objetos com os valores ocultos variando entre  $a$  e  $a+b$ , sendo  $a$  e  $b$  números reais maiores que zero. Sejam  $X$  e  $Y$  subconjuntos quaisquer de  $Z$ , podemos comparar a soma dos valores destes dois subconjuntos e determinar se a soma de  $X$  é maior, igual ou menor do que a soma de  $Y$ . Sabemos que existem  $N$  objetos com valor  $a+b$  e  $M$  objetos com valor  $a$ , quais verificações precisamos fazer para localizar todos os objetos com valor  $a+b$ ?

Problema no jogo: Seja  $Z$  um conjunto com 3 objetos com valores ocultos variando entre  $a$  e  $a+b$ , sendo  $a$  e  $b$  números reais maiores que zero. Seja  $X$  e  $Y$  subconjuntos quaisquer de  $Z$ , podemos comparar a soma dos valores destes dois subconjuntos e determinar se a soma de  $X$  é maior, igual ou menor do que a soma de  $Y$ . Sabemos que existe 1 objeto com valor  $a+b$  e 2 objetos com valor  $a$ , quais verificações precisamos fazer para localizar todos os objetos com valor  $a+b$ ?

Resolução: Precisamos comparar o subconjunto  $X$  com 1 objeto e o subconjunto  $Y$  com 1 objeto. Se a soma de  $X$  for maior, então seu objeto é o que queremos. Se a soma de  $X$  for menor, então o objeto de  $Y$  é o que queremos. Se a soma de  $X$  for igual a soma de  $Y$ , então o objeto que não pertence a  $X$  união  $Y$ , é o que queremos.



Figura 36: Situação do problema 10 (fonte própria).

## 6.2. Aplicação e resultados

Com acesso aos testes realizados na narrativa, fizemos a releitura de dois casos em particular: as 27 crianças do Ensino Fundamental I e os 142 alunos do Ensino Médio Técnico. Priorizamos estes testes por duas razões, os dados coletados eram quantitativos, simplificando o processo de análise e envolviam a aprendizagem de conteúdos matemáticos durante o jogo.

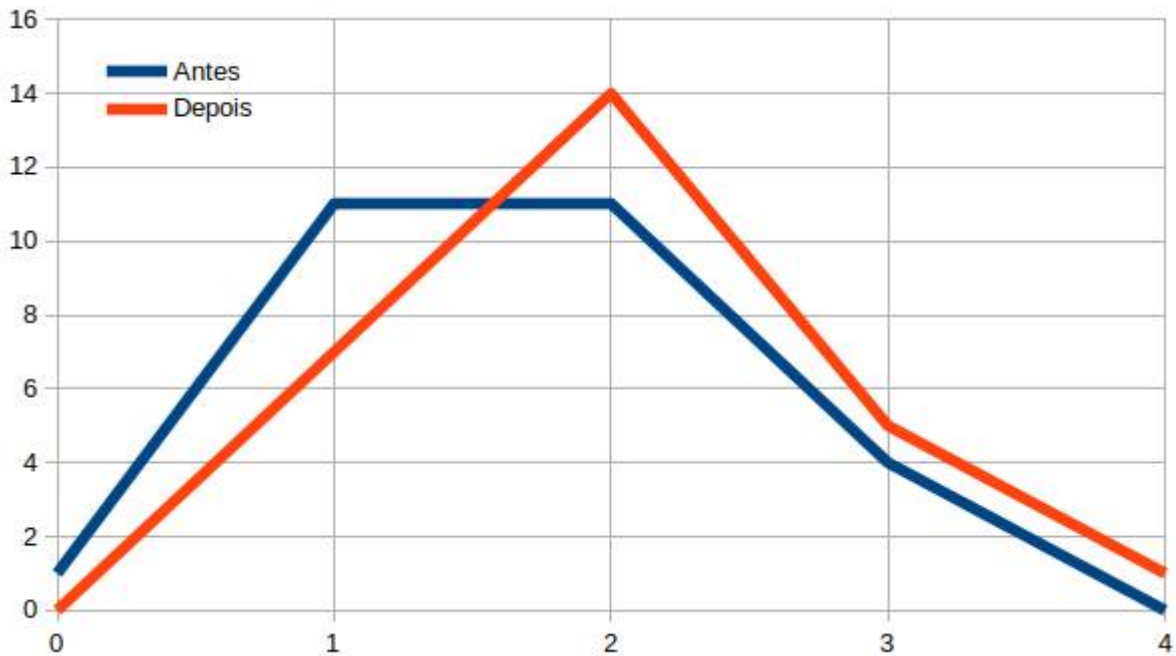
As respostas foram corrigidas atribuindo nota mínima 0 e máxima 1. Cada alternativa correta equivalia a uma fração da unidade, enquanto cada alternativa errada reduzia a mesma fração.

Para a análise dos resultados, aproximamos as notas no valor inteiro mais próximo. Nesta aproximação consideramos duas casas decimais. As notas com 50 centésimos da unidade eram aproximadas para cima. Por exemplo 6,50 era aproximado para 7,00 enquanto 6,49 era aproximado para 6,00.

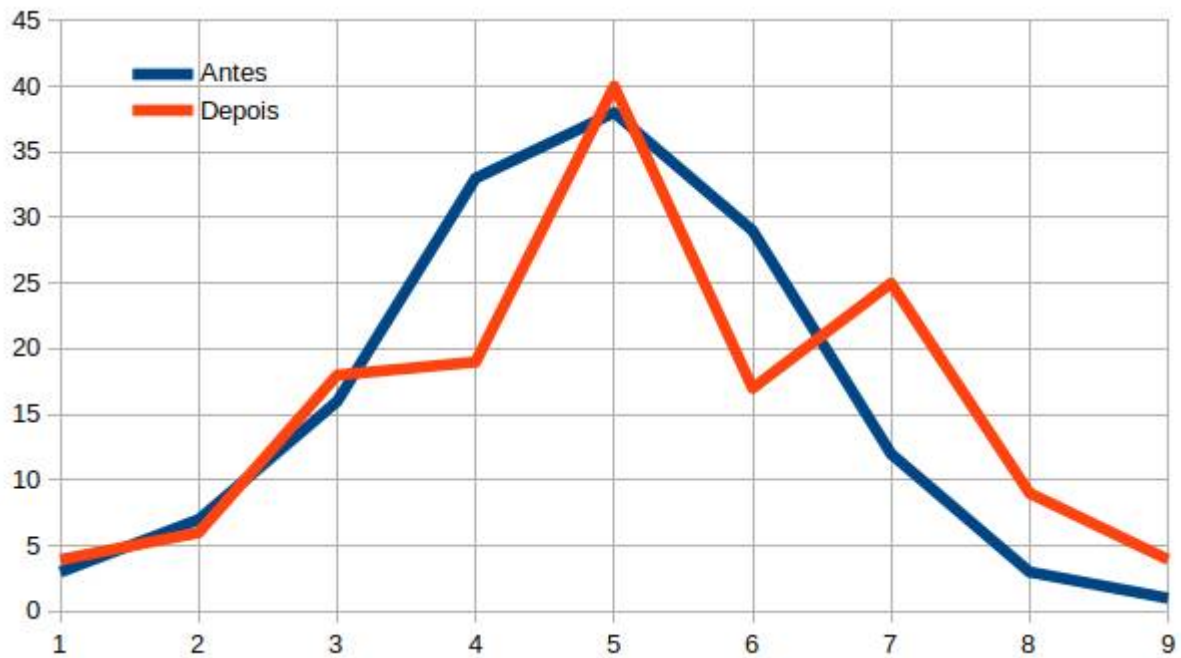
Distribuimos em categorias os resultados de cada par de questões para ambos os grupos. O procedimento utilizado foi a média aritmética da porcentagem de acertos de cada par de questões (questão N e questão 20+N, sendo N um número Natural entre 1 e 10). No caso das crianças por uma diferença média de 22% pudemos separar suas respostas em duas categorias de notas. Já no caso dos alunos do Ensino Médio Técnico, conseguimos formar três grupos de acertos, cuja diferença média das questões com maiores acertos para aquelas com acertos medianos era de 20,5% e a diferença média das questões com acertos medianos para aquelas com menores acertos era de 18%.

Assim, as crianças do Ensino Fundamental I obtiveram melhores resultados nas 4 primeiras questões de cada questionário. A variação de acertos no primeiro questionário (questões 1 à 4) ficou entre 26 e 42%, e no segundo questionário (questões 21 à 24) entre 37 e 46%. As 6 últimas questões de cada questionário tiveram índices de acertos bem menores. A variação de acertos no primeiro questionário (questões 5 à 10) ficou entre 0 e 7% e no segundo questionário (questões 25 à 30) entre 4 e 19%.

Os alunos do Ensino Médio Técnico obtiveram melhores resultados nos pares de questões 1 e 21, 4 e 24, 7 e 27. A variação de acertos no primeiro questionário (questões 1, 4 e 7) ficou entre 65 e 84%, e no segundo questionário (questões 21, 24 e 28) entre 69 e 82%. Os estudantes obtiveram resultados medianos nos pares de questões 2 e 22, 3 e 23, 5 e 25, 8 e 28, 9 e 29. A variação de acertos no primeiro questionário (questões 2, 3, 5, 8 e 9) ficou entre 34 e 55% e no segundo questionário (questões 22, 23, 25, 28 e 29) entre 37 e 64%. Seus piores resultados ocorreram nos pares de questões 6 e 26, 10 e 30. A variação dos acertos no primeiro questionário (questões 6 e 10) ficou entre 12 e 19% e no segundo questionário (questões 26 e 30) entre 13 e 23%.



**Figura 37:** Frequência de notas no Ensino Fundamental I (fonte própria).



**Figura 38:** Frequência de notas no Ensino Médio Técnico (fonte própria).

Para cada grupo, calculamos a frequência das notas em ambos os questionários e comparamos sua distribuição a partir de dois gráficos de linhas. O eixo horizontal representando o intervalo de notas e o eixo vertical a frequência absoluta de cada nota. Uma forma de analisarmos estes gráficos é observando as regiões onde a linha vermelha (notas após o jogo) está acima da linha

azul (notas antes do jogo). Se o jogo proporciona a aprendizagem dos conteúdos tratados nos questionários, então à esquerda do gráfico a linha azul deve sobressair-se sobre a linha vermelha, enquanto à direita do gráfico, a linha vermelha deve sobressair-se sobre a linha azul. Em outras palavras, isto indica que a quantidade de jogadores que obtiveram as notas mais baixas no primeiro questionário, diminuiu após o jogo. E por consequência, o número de jogadores a obterem as notas mais altas, aumentou.

Podemos considerar que em ambos os grupos, houve o aumento no desempenho dos jogadores nos conteúdos matemáticos tratados pelo jogo comuns aos questionários. E como não tiveram intervenções entre o jogo e os questionários, podemos concluir que o jogo foi o principal mediador desta aprendizagem.

### 6.3. Teoria e prática

Comparando o que foi discutido na revisão da literatura com a narrativa do diário de *game design*, desenvolvemos para cada obra os principais pontos refletidos nesta experiência. O intuito desta análise é enunciar ao leitor os aspectos considerados por alguém no perfil de um *designer de problemas* na hora de desenvolver seu próprio *puzzle dinâmico*, assim como, observar as características do *game design* discutidas da literatura que tiveram relações simétricas ou assimétricas com a prática.

Do artigo de Fellows (1996), **The Heart of Puzzling: Mathematics and Computer Games**. A introdução do Pega e Larga discute como acoplar problemas matemáticos com processos de manipulação da informação, um aspecto presente no artigo que propõe o uso de manipuladores da informação comuns a jogos populares em listas de problemas da categoria NP. Contudo, nesta experiência, percebemos que embora qualquer problema da categoria NP possa ser usado como um desafio, pois sua resolução não é nítida independente do manipulador da informação disponibilizado ao jogador (dado que se assim fossem, estes problemas seriam facilmente resolvíveis na vida real), o uso de problemas de ordens polinomiais baixas (como  $n$ ,  $n^2$ ,  $n^3$ ), também podem apresentar soluções nebulosas o bastante para serem desafiadores aos jogadores quando munidos de manipuladores da informação específicos. Assim, usar problemas da categoria NP para este método, parece uma condição suficiente, porém não necessária.

Outra relação do artigo com a experiência descrita, foi a manutenção parcial da forma como a informação era manipulada (desde a versão Handles). Enquanto os problemas variavam constantemente dentro de suas contextualizações (seção 6.1), os recursos para resolvê-los pouco mudavam, se assemelhando à etapa descrita no artigo como “girar a manivela” que manipula os

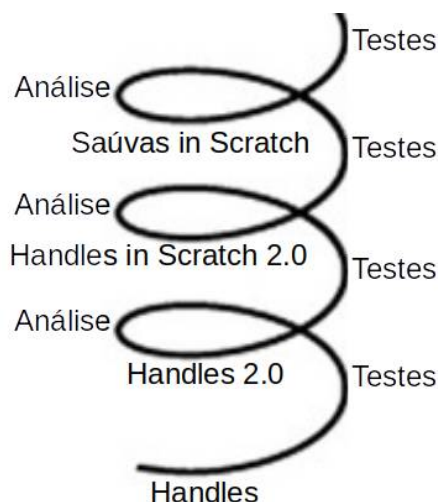


dados do problema. Como exemplo deste “giro” temos os problemas envolvendo soltar os números a partir da escolha do divisor comum zero. Somente após produzir várias versões do jogo, percebeu-se como adaptar tanto os problemas quanto a manipulação da informação para nivelar esta forma de interação do jogo com os jogadores.

Do livro de Costikyan (2013), **Uncertainty in Games**. Tanto o livro quanto a narrativa percebem o jogo como uma subclasse do ato de jogar, visto que a interação e experiência do jogador fundamentaram o desenvolvimento dos projetos. Outro aspecto relacionado a esta obra, são os tipos de incerteza presentes.

Tomando a última versão (Saúvas in Scratch) por exemplo. Os níveis iniciais evocam no jogador a incerteza de performance, pois o jogador precisa mover-se entre espaços estreitos. A incerteza de solução aparece por conta do jogo não conter lições ou fatos relacionados a Matemática, o que pode levar o jogador a noções equivocadas ou dúvidas a cerca do conjunto resposta, o que é averiguado na hora de executar uma possível solução. Complexidade analítica é a incerteza predominante na parte final do jogo, que surge da necessidade dos jogadores lidarem com uma árvore de decisões complexa, selecionando as sementes sem vê-las, permanecendo a dúvida em torno das decisões certas terem sido tomadas.

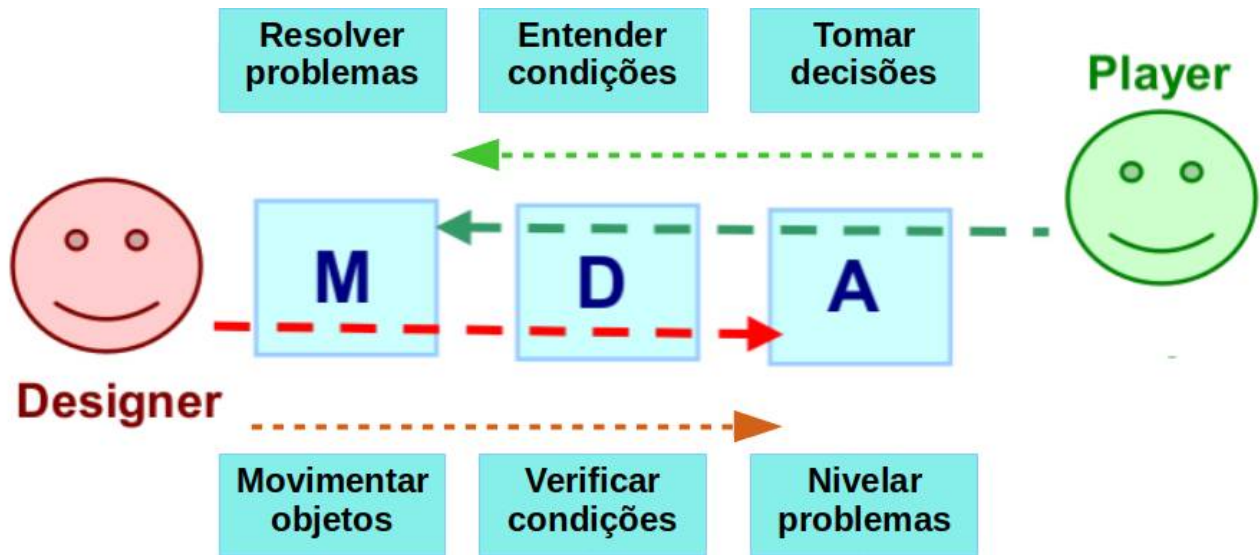
Do artigo de Zimmerman (2003), **Play as research: the iterative design process**. Nas cinco versões do jogo, discutidas durante a narração, percebemos o processo de testagem contínuo logo após a construção do protótipo. As análises dos testes preservaram a interação com o jogador presente desde o Pega e Larga e fizeram os problemas matemáticos se adaptarem a ela formando o Handles. Que quando testado, indicou a falta de uma interface clara, levando ao Handles 2.0. Seus testes mostraram que o jogo não sensibilizava os jogadores, gerando então Handles in Scratch 2.0, que se mostrou longo e complexo demais para jogadores iniciantes. Resgatando características de 3 versões anteriores com os recursos técnicos da atual, permitindo produzir Saúvas in Scratch.



**Figura 39:** Modelo iterativo deste *puzzle dinâmico* (adaptado de ZIMMERMAN, 2003).

O método de *game design* iterativo proposto pelo artigo aparece em todo ciclo narrativo das 5 versões do jogo discutidas no diário de *game design*. Podemos perceber de todas as versões que em nível de base já iniciaram seu processo de testagem. Isto levou a uma tomada de decisões antes que investimentos mais sérios fossem feitos, como a escolha por preservar a mecânica de Pega e Larga e eliminar toda sua problemática deste jogo.

Do artigo de Hunicke, LeBlanc e Zubek (2004), **Mda: A formal approach to game design and game research**. Dentro da narrativa os termos mecânicas, dinâmicas e estéticas aparecem várias vezes ao longo de todo o corpo do texto. A forma como são empregados no diário estão de acordo com o modelo MDA proposto no artigo, designando respectivamente para mecânica, componentes do jogo em nível de regras ou funcionamento isolado das partes: para dinâmica, as relações do sistema, ou como partes isoladas se relacionam criando interações de jogo; Para estética, na sensação do jogador ao enxergar aquele sistema em funcionamento. De forma geral, podemos perceber na experiência de evolução das versões do jogo, as seguintes trajetórias do ponto de vista de *game designer* e jogadores:



**Figura 40:** Modelo MDA deste *puzzle dinâmico* (adaptado de HUNICKE, LEBLANC, ZUBEK, 2004).

Na prática, a ideia inicial que antecedeu a primeira versão foi formulada na visão de jogador, como um jogo onde os problemas matemáticos fossem resolvidos com a tomada de decisões; A diversão estaria na sensação de resolvê-los sem fazer cálculos. O sistema seria compreender as condições que cercam um problema a partir das instruções ou leitura do cenário, e as regras do jogo estipulariam os meios para resolvê-lo. Isto se consolidou somente nas versões posteriores, pois a princípio o jogo saiu bastante diferente do que era estipulado.

Na primeira versão, Pega e Larga, a mecânica determinava a ordem dos números inseridos na máquina. A dinâmica estipulava os números e a operação da máquina. A estética era descobrir a operação que a máquina realizava. Esta experiência coincide com a proposta do artigo, de que pequenas mudanças em uma direção, resultam um efeito cascata no extremo oposto.

Do artigo de Gee (2005), **Good video games and good learning**. No decorrer da narrativa é possível perceber a atenção direcionada na construção de um jogo do qual os usuários aprendam sozinhos o funcionamento. Não havendo lições ou fatos matemáticos a serem lidos dentro destes cinco jogos. Dessa forma, esperou-se até que os jogadores conseguissem aprender a jogar sozinhos (início de Handles in Scratch 2.0) para investigar se os conteúdos matemáticos do jogo também eram aprendidos. Os resultados tanto na narrativa quanto na seção 6.2 mostraram a eficiência desta versão em ensinar seus próprios conteúdos de Matemática.

Embora no processo de construção destes jogos, diversos princípios da aprendizagem tenham aparecido, “ferramentas inteligentes e conhecimento distribuído” parece ser o mais influente em todas as versões. Também foi considerado como um princípio ausente nos jogos desenvolvidos

pelo PIBID e FoG durante a introdução deste trabalho. Nas versões do jogo (tanto pela narrativa quanto por experimentá-las) podemos perceber que o computador empresta ao jogador por meio do jogo, sua velocidade e precisão na hora de realizar operações Matemáticas. Permitindo que o jogador resolva diversos problemas matemáticos tomando decisões e pegando emprestado a capacidade de cálculo do computador para ajudá-lo.



**Figura 41:** Exemplo de interação com ferramentas inteligentes do *puzzle dinâmico* (fonte própria).

Do artigo de Nakamura e Csikszentmihalyi (2009), **The concept of flow**. O conceito de fluxo proposto pelo artigo, pode ser percebido em duas ocasiões desta narrativa. A primeira envolvendo os testes de Handles 2.0, onde os participantes tinham o rosto e o jogo filmados. Esta abordagem procurava verificar se o jogo mantinha o jogador em uma constante e leve variação entre ansiedade e apatia mediante as ações de jogo. Um resultado positivo mostraria de acordo com o artigo, que o jogo coloca o jogador no máximo de suas capacidades. Podemos observar nestes testes que se usou o modelo gráfico simplificado para o intervalo de fluxo (figura 14), que aparece em muitos textos de *game design*.

Outra ocasião onde o conceito de fluxo foi investigado envolvia os testes de Handles in Scratch 2.0. Quando durante as entrevistas, verificava-se a perda da noção temporal pelos jogadores, uma característica comum a pessoas que estiveram em estados de fluxo.

Do livro de Bogost, Ferrari e Schweizer (2010), **Newsgames: Journalism at Play**. Podemos relacionar esta obra com o jogo ocupando o papel de uma mídia da comunicação. Inicialmente temos o elemento do enigma, que instiga o público a procurar soluções dentro das condições estipuladas. Também podemos identificar no jogo o aspecto de letramento, visto que trabalha seus próprios conteúdos necessários para ser consumido. A interação do público com o conteúdo do material pode ser percebido mediante o diálogo de alguns jogadores com o mecanismo de *feedback* Amanda, durante alguns testes do Handles in Scratch 2.0.

Temos também na facilidade computacional para editá-lo em suas versões mais avançadas, o jogo como recurso para construção de críticas. Isto permite sua reutilização em diferentes contextos, como por exemplo, quando os jogadores que testavam o *Handles in Scratch 2.0*, editaram o código do jogo, adaptando na estética do jogo uma crítica própria ao contexto político do Brasil.

Do livro de Juul (2013) **The art of failure: An essay on the pain of playing videogames**. A relação deste livro com a narrativa é implícita a forma como as falhas aparecem no jogo. Este jogo eminentemente obriga os jogadores que não sabem jogar, a falharem para aprenderem. Prova disto está na ausência de lições aos jogadores, logo é estimulado que o jogador teste hipóteses sucessivas vezes. Em todas as versões o fracasso é sutilmente incentivado, dando ao jogador, ilimitadas chances de errar; porém a frustração do nível ser reiniciado sem dicas ou permitir pequenos ajustes, pode levar ao desprazer da falha. Isto foi discutido na narrativa dos testes de *Handles 2.0*, quando os jogadores com o rosto filmado permaneciam em ansiedade ou apatia.

Sem alterar os benefícios ao jogador, a inclusão do mecanismo de *feedback* Amanda pareceu quebrar a seriedade da falha. Isto aparece nos testes de *Handles in Scratch 2.0*, onde os educadores conversavam com a personagem. Também aparece nos questionários das crianças do Ensino Fundamental I e alunos do Ensino Médio Técnico, que se mostraram positivos em relação ao jogo, e pelo próprio incentivo a falhar que os jogadores observados tinham. Ao que parece, a principal consequência deixava de ser recomeçar o nível e se tornava irritar ou cansar a personagem.

Do livro de Koster (2014), **A Theory of Fun for Game Design**. Podemos relacionar a narrativa a este livro na forma como o jogo é tratado, sendo um presente ao cérebro que busca resolver os 10 problemas enunciados na seção 6.1. Os gráficos, a evolução dos desafios e o contexto do jogo visam amenizar o estresse no processo de resolução. De forma similar a esta obra, a narrativa explora como uma interação com os jogadores pode suportar contextos muito adversos, desde buscar números para uma máquina, coletar sementes para um formigueiro, recolher dinheiro para um político.

Também associado ao livro, temos a construção de jogos desafiadores, que lidam com problemas sem uma solução correta definida. Nas versões expostas do jogo e na análise dos problemas (seção 6.1), o desafio do jogo não é inserir na máquina/formigueiro a resposta correta. E sim, reunir a resposta correta do conjunto de todos os objetos. A inserção na máquina/formigueiro serve apenas para verificar a validade da solução encontrada, a qual pode ser obtida das mais variadas maneiras, como exemplificado nos testes de *Handles in Scratch 2.0*, onde uma jogadora utilizou o teletransporte para recolher cada número diferente de 0.

Do livro de Shuytema (2008), **Design de Games: Uma abordagem prática**. Este guia apareceu pouco relacionado com a narrativa. Se revermos a definição de um *designer de problemas*,

perceberemos que o sujeito com este perfil possui experiência no desenvolvimento de jogos, enquanto a forma como o livro explora o *game design* ocorre de modo mais introdutório, trazendo ao leitor, exercícios práticos sobre os processos criativos necessários para construir jogos.

Outro aspecto que parece contribuir para a baixa aplicação deste guia no processo prático narrado, foram os conhecimentos técnicos a quem este livro se destina, por colocar boa parte dos exemplos da construção de um jogo a partir da linguagem de programação *script Lua*. O que não pôde ser aplicado diretamente por alguém com domínio na linguagem de programação *Scratch*.

Do livro de Rogers (2012), **Level Up!: Um guia para o design de grandes jogos**. Um aspecto muito peculiar discutido neste livro e relacionável com a narrativa são os 3 C's. No quesito *personagem/character* (capítulo 5), temos a preocupação da forma corresponder ao papel no jogo. Assim, a busca de objetos é associada com um boneco de palitinho que move as pernas, um *skatista* de palitinho e a uma formiga colheiteira, enquanto a designação das tarefas é atribuída a uma usuária irritada ou a uma formiga rainha. Também associado ao personagem, temos seu uso como forma de medida para o jogo, visto que o tamanho dos números, sementes e bolinhos foram projetados a partir de sua altura.

Em relação à câmera (capítulo 6), temos associada na narrativa a presença dos HUDs (capítulo 8), onde ocorre o aproveitamento do espaço ocupado pelas informações auxiliares para variar o cenário desde uma região retangular com largura mais acentuada até algo mais próximo de um quadrado.

Aspectos como controles (capítulo 7) foram analisados durante os testes do *Handles in Scratch 2.0*, para verificar se o jogo era difícil por conta de seus controles confusos no teclado ou se a dificuldade realmente estava na resolução dos problemas.

Outros aspectos mais sutis também podem ser associados a este livro, como a preocupação com a sonoridade no título do jogo. Porém a forma como o *designer de problemas* trata o plano diretor do jogo foge ao modelo proposto na documentação (capítulo 4). Diverge também desta obra a direção na qual os níveis são construídos (capítulo 9), tomando como base a experiência do jogador ou a jogabilidade pretendida para então desenvolver os níveis e o mundo de jogo.

Do livro de Chandler (2012), **Manual de produção de jogos digitais**. Muito do que se discute nesta obra é inaplicável a uma equipe individual que busca construir um *puzzle dinâmico* a partir de recursos simplórios. Como por exemplo, o processo de contratação e manutenção de talentos, jogos *on-line multijogador em massa*, *voiceovers* entre outros, sendo um texto voltado para o gerenciamento de equipes profissionais para a produção e comercialização de jogos. Porém o modo organizacional de projetos proposto no livro foi em parte aderido durante a narrativa, que dividiu o processo:

- a) Formação do conceito, o trecho que precede a pré-produção, onde são discutidas as reflexões que movimentaram o novo ciclo de produção;
- b) Pré-produção, onde são enunciadas e explicitadas os requisitos visuais e lógicos para a construção do jogo;
- c) Produção e testes, períodos concomitantes que acompanham o desenvolvimento dos níveis e revisão dos requisitos do jogo;
- d) Finalização, etapa de arquivamento dos recursos e análise crítica do projeto.

Outro aspecto comum ao diário de *game design* com este livro, é a importância de organização do próprio código do jogo e identificação automática de erros. O que apareceu inicialmente em *Handles* quando o cachecol vermelho foi utilizado para avisar ao desenvolvedor de que a habilidade estaria ativa. Também em *Handles 2.0* quando após reorganização do código, inseriu-se as teclas do jogo mudando de cor quando ativas. Facilitando identificar problemas de ativação das teclas. Por último em *Handles in Scratch 2.0*, aderiu-se a modularização do jogo e codificação dos níveis, reduzindo as componentes que precisavam ser checadas antes de liberar o jogo.

Dos livros de Salen e Zimmerman (2012), **Regras do jogo: fundamentos do design de jogos: volumes 1, 2, 3 e 4**. A forma de construção da narrativa acompanha de certo modo o desenvolvimento dos volumes desta obra. Inicialmente a discussão se encontra voltada ao sistema cibernético, que possui as seguintes três componentes autorreguladoras: o sensor que mede um aspecto do ambiente (disposição do problema na tela de jogo), o comparador que avalia e decide sua ação (reações do personagem com o objeto) e o ativador que reage (máquina/formigueiro).

Em meio ao desenvolvimento das versões, percebemos a preocupação na experiência de simulação proporcionada ao jogador. Que embora todos os jogos sejam simuladores, nem todos os simuladores são de fato jogos.

Por último, notamos a retórica cultural do jogo na escolha da personagem Amanda com aparência juvenil em mangá, um estilo de desenho japonês bastante difundido no Brasil. Ou ao cenário de formigas carregando sementes a um formigueiro, um contexto comum em todo o mundo.

## 7. Considerações finais

Por este estudo exploratório não contar com hipóteses de pesquisa a serem respondidas, dado que objetivamos explicar suas descobertas com base em teorias relacionadas. Somente podemos considerar a pergunta de pesquisa (como ocorre a prática de um *designer de problemas* no desenvolvimento de um *puzzle dinâmico*?) a partir do objetivo geral de estudar o processo de um *designer de problemas* durante a construção de um *puzzle matemático* eficiente. Assim, podemos entender a eficiência de um *puzzle dinâmico* ligado a aprendizagem, durante o jogo, de seus conteúdos necessários para jogá-lo, neste caso a resolução de alguns problemas Matemáticos;

De acordo com os testes mostrados na seção 6.2 deste trabalho, o *puzzle dinâmico* Handles in Scratch 2.0 construído a partir deste método narrado apresentou com dois públicos bastante distintos (Ensino Fundamental I, Ensino Médio Técnico) resultados positivos da forma como aprenderam a jogá-lo enquanto jogavam. Isto apareceu nas melhoras dos resultados nos testes de Matemática antes e depois, que avaliavam a resolução de problemas Matemáticos cuja compreensão era requisito para conseguir jogar. Dessa forma, dado sua capacidade de ensinar aos seus próprios jogadores os conteúdos Matemáticos necessários para jogá-lo, analisar como as principais teorias do *game design* discutidas neste trabalho se relacionam com sua construção, permitem e auxiliam o tratamento dos sintomas do *Wicked Problem* de criar materiais semelhantes para o ensino de Matemática (MATEAS, STERN, 2005).

A seguir construiremos uma análise acerca de como sintetizamos esta experiência (a revisão da literatura relacionada com a experiência e as contribuições para a educação Matemática). A forma como esta construção ocorre, discute em um esquema simplificado por qual autor e obra um *designer de problemas* em uma situação inicial de desenvolvimento de um *puzzle dinâmico*, poderia se basear para concluir seu trabalho em situação semelhante a esta descrita no trabalho.

Embora no diário de *game design*, pouco tenha se usado Shuytema (2008), isso ocorreu, principalmente, por conta do autor dos documentos já possuir experiências no desenvolvimento de jogos. Assim, este livro poderia ter uma contribuição maior para alguém novo nesta atividade, que busque através de exercícios e tarefas, estimular as habilidades criativas e lógicas envolvidas na elaboração de um jogo. Dessa forma, esta base pode ser considerada o passo inicial deste ciclo, quando a experiência criativa ou os materiais de suporte não são suficientes para um palpite aproximado daquilo que queremos. Também é um material adequado para *designers de problemas* que compreendam a linguagem *script Lua*, ou outras linguagens cujos exemplos em *script Lua* deste livro possam ser facilmente adaptados.



Como primeira tentativa de ajustar o palpite inicial no formato de um jogo digital para ensino de Matemática, temos a proposta de Fellows (1996) de discretizar o jogo em problemas matemáticos e manipuladores de dados. Este é um processo que exige alguma profundidade em Matemática para generalizar o problema e as ferramentas envolvidas na sua resolução. Nesta experiência, este método foi adequado para ajustar as soluções dos desafios a partir da escolha de manipuladores apropriados. De forma simples, esta fase envolve escolher axiomas e propriedades ao nível de um problema, para quem for resolvê-lo não achar trivial ou muito trabalhoso. Reconhecendo a importância deste método, como passo inicial ao desenvolvimento, apresentamos no apêndice 4, um exemplo passo a passo de como é possível aplicá-lo na combinação de um problema NP com a mecânica de um jogo digital popular.

Com um esquema de jogo pronto, Rogers (2012) trata de aspectos muito particulares ligados a experiência de jogar e sugestões para facilitar o trabalho do *game designer*. Diferente da abordagem de Shuytema (2008) que tem o enfoque mais preparatório, este livro trabalha a prática que não é visivelmente percebida no desenvolvimento do jogo. Na experiência narrada, é possível perceber diversos ajustes sutis que passariam despercebidos dentro de um filtro objetivo de detalhes. Como por exemplo: a facilidade de jogar estar relacionada ao aparelho físico; a posição dos botões do controle estar de acordo com o alcance dos dedos do jogador; a forma do personagem indicar suas funções; o tamanho do personagem ser utilizado como unidade de medida para o jogo. O papel deste livro no processo descrito e experiência narrada, se assemelha a um catálogo generalizado de precauções e sugestões aos desenvolvedores de jogos. Tê-lo como base a partir desta fase da produção, pode prevenir o *designer de problemas* de alguns entraves.

A perspectiva de Hunicke, LeBlanc e Zubek (2014) proporciona ao *designer de problemas* uma forma de expressar e tratar as componentes de seu jogo durante toda a produção. A própria percepção de que o jogo digital é um artefato e não uma mídia, auxiliaria nesta etapa, a reconhecer e contornar o efeito de cascata presente nos interesses de quem constrói o jogo e de quem o joga. Em termos práticos, os jogadores ao testar o jogo sugerem mudanças que eles mesmos não compreendem o custo para construir e a influência que provocaria na jogabilidade. O processo de mudanças do jogo em um nível de construção (mecânica, dinâmica, estética) não é linear, sendo difícil de se prever as consequências a menos quando nos colocamos na visão dos jogadores.

Nesta etapa do desenvolvimento, de acordo com a experiência narrada, a ideia já se encontraria no formato de um jogo e as sugestões dos jogadores a testarem o produto cresceriam constantemente, um aspecto no qual a organização de projetos tratada no livro de Chandler (2012) se torna bastante conveniente. Justamente por este livro se concentrar na organização de equipes com diversos profissionais específicos, é que parece simples usá-lo para organizar uma equipe

individual. De fato, as sugestões do autor ultrapassam as preocupações necessárias para um *designer de problemas*, mas também cobre aspectos organizacionais ao nível de um projeto complexo, pois o *puzzle dinâmico* é um jogo pequeno se comparado a produções da indústria (conforme foi visto no capítulo 5, Documentos), e por isto é possível coordenar sua produção mesmo sem um sistema (como ocorreu na produção das duas primeiras versões, Pega e Larga, e Handles, conforme narrado no diário de *game design*). Utilizar a forma sistemática de organizar um grande projeto neste caso, pode ser aplicada a um projeto pequeno (o contrário nem sempre se aplica). Assim, nesta etapa, o processo construtivo pode se beneficiar ao ser reestruturado, ter suas componentes organizadas e contar com o desenvolvimento de recursos simplificados para detecção de falhas e ajustes no jogo. Requisitos necessários para qualquer grande projeto e que tornam o desenvolvimento de um pequeno projeto mais eficiente, dado que ainda serão realizados muitos ajustes e correções de falhas ao longo da produção. E a falta de uma organização, poderá tornar o avanço do projeto confuso e mais trabalhoso do que o necessário.

Perceber que um produto deve satisfazer os interesses de seu usuário é uma característica fundamental para o modelo de design iterativo. No caso, conforme Zimmerman (2003) propõe, o jogo também é um produto e precisa satisfazer os interesses do jogador. Logo, seu desenvolvimento sugerido deve seguir um processo similar ao modelo de design iterativo. Partindo de um protótipo, testando-o, analisando seus resultados e reestruturando o produto para um novo ciclo deste processo. No caso, o *designer de problemas*, após a organização de seu método para trabalho e de uma reflexão sobre como as modificações no jogo repercutem na percepção do jogador; poderá enfrentar uma série de testes e iterações no seu jogo, sem se perder naquilo que está sendo desenvolvido e analisando de forma reflexiva os efeitos de cada mudança. Se essas características forem ignoradas, poderiam levar facilmente o *designer de problemas* a uma série de mudanças ao acaso na esperança de que alguma delas coincida exatamente com o interesse dos jogadores.

Ainda que o jogo esteja com aparência de jogo e seus testadores mais experientes gostem, há uma preocupação latente sobre os novos jogadores. Nesta experiência, percebemos a importância dos princípios da aprendizagem enunciados por Gee (2005), no sentido de que um jogo digital deve ensinar ao jogador como jogá-lo. Logo se a interação com o usuário pretendida não está sendo alcançada é importante refletir a partir de produtos semelhantes, que princípios da aprendizagem eles utilizam para que seus jogadores aprendam rapidamente a utilizá-los. Uma alternativa seria de forma crítica observar quais princípios da aprendizagem o jogo que desenvolvemos realmente trabalha. Neste aspecto, procurar entender se componentes do jogo não relacionados a interação pretendida estão desviando a atenção dos nossos jogadores. Outro princípio a ser observado fortemente, são as ferramentas inteligentes e conhecimento distribuído, pois os jogos digitais

possuem um potencial de interação com o jogador dentro de problemas matemáticos que não deve ser ignorado. Sua velocidade para representar graficamente funções, lançar eventos não-determinísticos, verificar condicionais de forma incansável, são só algumas das ações possíveis para auxiliar o trabalho com problemas Matemáticos, os quais permitem abordagens das quais o lápis e papel não suportariam em tempo hábil de resolver vários problemas de natureza real.

Um princípio da aprendizagem proposto por Gee (2005) é o risco. Um incentivo ao jogador correr riscos e falhar nos jogos sem medo das consequências. Da mesma forma, Juul (2013) tece um detalhado estudo sobre as consequências paradoxais da falha nos jogos, um aspecto que deve ser considerado no processo de incentivar o jogador a falhar. Considerando que a falha possui muitos aspectos desejáveis e indesejáveis, encontrar uma medida em que o jogador se sinta confortável para falhar é permitir que novas hipóteses sejam testadas sem amedrontar o jogador. Porém a ausência de consequências remove a falha do jogo, que embora seja frustrante, é procurado pelas pessoas que jogam. Assim, experimentar nesta etapa do desenvolvimento, maneiras de desviar a atenção do jogador sobre sua incapacidade de resolver os problemas pode ser um reforço para que arrisquem novas hipóteses e aprendam com seus erros.

Com o produto finalizado após diversas iterações e ajustes nos mecanismos de aprendizagem e incentivos a falha, é importante averiguar o fluxo no jogo. Oposto aos primeiros testes para averiguar o estado de fluxo descritos no diário de *game design*, é mais simples analisarmos se ocorre nos jogadores sensações comuns a quem vivencia o estado de fluxo. Uma prova sólida de que os jogadores vivenciaram realmente o estado de fluxo é bastante difícil, inviável e pouco útil para o interesse de desenvolver um *puzzle dinâmico*.

Em relação ao que foi realizado no diário de *game design*, utilizar do estado de fluxo para comprovar que os jogadores aprendem os conceitos envolvidos no jogo por estarem durante o ato de jogar, em poder de suas capacidades máximas, é um método muito mais complexo e subjetivo, podendo ser substituído pela aplicação de um questionário anterior e posterior sem outros mediadores senão o próprio jogo. Dessa forma, com base nestas experiências e na compreensão do estado de fluxo proposto por Nakamura e Csikszentmihalyi (2009), reunir evidências isoladas de comportamentos comuns a quem vivencia o estado de fluxo é um indicativo forte para que o *puzzle dinâmico* tenha boa fluidez.

Com um jogo que ensina os jogadores a jogarem, incentiva-os a falhar e compartilha características de um estado de fluxo, resta agora investigar a incerteza envolvida. Diferente do planejamento inicial de um jogo ou de sua fase de prototipagem, a análise da incerteza provocada no jogador é diretamente ligada ao produto final. Segundo a análise das incertezas nos jogos de Costikyan (2013), várias formas de incerteza não estão ligadas as ações dos jogadores em tempo de

jogo. Podem envolver a própria evolução do jogo, ou o significado social da história que o jogo conta. Porém, um aspecto que não pode ser ignorado é que a incerteza movimenta o jogo por meio do ato de jogar. Se não há incertezas, para o jogador não há jogo. Dessa forma, antes de pensarmos em encurtar nosso jogo, é importante observarmos se trabalhamos com algum tipo de incerteza inesgotável. Algo que o jogador não consiga reduzir a trivialidade dos resultados. A própria mudança das habilidades, dos cenários, dos contextos, renovam a incerteza vivenciada pelos jogadores. Erros no seu ajuste, podem fazer do jogo uma sequência interminável de ações triviais e determinísticas, ou um evento aleatório de complexidade caótica.

Como um dos aspectos finais no desenvolvimento de um *puzzle dinâmico* de acordo com o diário de *game design*, temos a necessidade de presentear o cérebro, enquanto este procura reconhecer padrões. Segundo Koster (2014), esta vem a ser a essência de qualquer jogo, um desafio cerebral com seu estresse aliviado por gráficos, histórias e músicas. Porém, encontrar desafios reais é algo que precisa ser avaliado como característica da diversão inerente ao jogo. No caso da experiência relatada tanto no diário de *game design* quanto na seção 6.1, os desafios das cinco versões do jogo não dizem respeito a conhecer números que obedeçam determinadas propriedades e sim, como selecioná-los dentre um contexto turbulento com ferramentas limitadas. De forma similar aos contextos reais onde o cérebro está acostumado a lidar com problemas que permitem inúmeras soluções não objetivas, os jogos desafiadores colocam o jogador frente a situações semelhantes. Neste ponto, cabe ao *designer de problemas*:

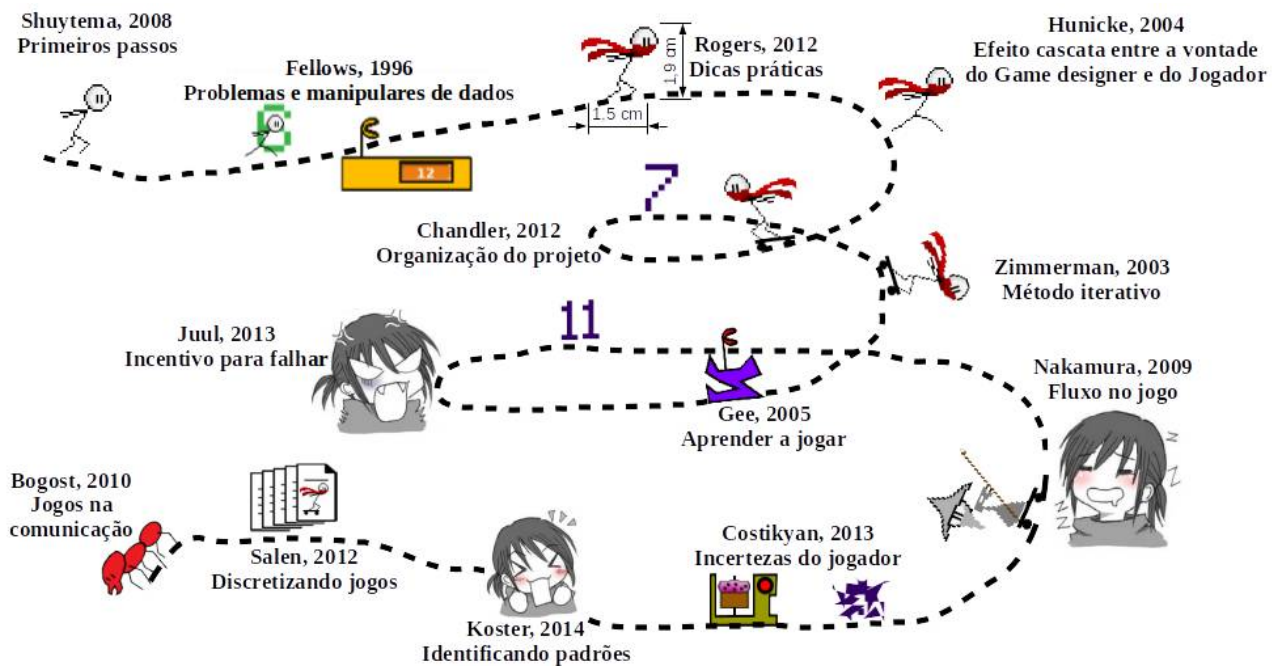
- a) Identificar que padrões precisam ser reconhecidos para jogar seu jogo.
- b) Reconhecer estes padrões de forma simplória é o ato de jogar, e o custo para estes padrões serem reconhecidos é o próprio jogo.
- c) Tomar decisões sobre aumentar ou reduzir a dificuldade dos padrões a serem reconhecidos;
- d) Reforçar ou relaxar os custos para reconhecê-los.

O resultado destas escolhas será a diversão proporcionada pelo jogo, sem a qual os jogadores não virão a jogá-lo.

Se afastando da perspectiva prática e aproximando da acadêmica, o *designer de problemas* como licenciado em Matemática, por vezes, precisará defender seu produto em ciclos acadêmicos, onde a prática se afasta da teoria. Nestas condições, os quatro volumes do trabalho de Salen e Zimmerman (2012a, 2012b, 2012c, 2012d) trazem uma fundamentação detalhada dos jogos com um sentido bastante amplo, permitindo que os mesmos sejam reduzidos a suas componentes teóricas mais técnicas. Tomando a experiência narrada como exemplo, este livro não se mostrou aplicável nos processos práticos de desenvolvimento, por seu caráter descritivo teórico.

Por último, esta análise remete ao papel comunicador dos jogos, os quais se assemelham muito a outras mídias jornalísticas. Venho a considerar a contribuição de Bogost, Ferrari, Schweizer (2010) ligada ao final do processo de construção de um *puzzle dinâmico* por um *designer de problemas*, visto que para comunicar antes é preciso construir uma mídia bem fundamentada. Algo que diferente dos *game designers* profissionais, para um *designer de problemas* não é um processo tão previsível e controlável. Tratar o jogo como artefato, e usá-lo como mídia através de modificações na sua estética sem compreender o efeito de cascata proposto por Hunicke, LeBlanc e Zubek (2004), pode comprometer sua interatividade. Da mesma forma que a falta de uma organização de projeto como a de Chandler (2012), torna inviável construir novos conteúdos sem prejudicar o jogo. Mesmo a experiência narrada pelo diário de *game design*, onde os alunos editaram uma versão do jogo, embora a mecânica de movimentação permanecesse funcional, outras componentes essenciais foram comprometidas. Assim, de acordo com a experiência narrada, o *puzzle dinâmico* nas condições posteriores a esta trajetória de construção, está viável para ser utilizado como uma matriz de novos conteúdos sem grandes prejuízos ou dificuldades de construção.

Abaixo, enunciamos o esquema de trajetória das referências analisadas de acordo com a experiência narrada e discutida acima. Quando uma obra possuía mais que um autor, utilizamos somente o primeiro dos autores como referência dentro do esquema.



**Figura 42:** Esquema de trajetória das referências analisadas (fonte própria).

Conforme aparece na própria narrativa do diário e no título do projeto aprovado pelo Comitê de Ética e Pesquisa (Pensamento Computacional no Ensino da Matemática), o jogo a priori tinha como objetivo dentro de um contexto com problemas Matemáticos, desenvolver o Pensamento Computacional de seus jogadores. Um termo definido por Wing (2006) como o conjunto de habilidades e atitudes que cientistas da computação utilizam no tratamento de problemas, para que estes possam ser resolvidos por computadores. Assim, aproveitando os trabalhos de Silva (2017a) e Silva (2017b), que relaciona o jogo na versão *Handles in Scratch 2.0* como um recurso em potencial para o ensino de Métodos Numéricos, tanto pelo uso de algoritmos comuns de busca (linear, bissecção, trisseção) como pela elaboração de algoritmos de solução automatizados. Esperamos investigar em uma continuidade desta própria pesquisa, a maneira com que habilidades comuns ao Pensamento Computacional aparecem nos jogadores após sua experiência jogando, e se estas são extrapoladas em contextos diferentes do próprio jogo.

Esta pesquisa trouxe como devolutiva à comunidade os seguintes produtos:

- a) Um *puzzle dinâmico* (*Handles in Scratch 2.0*) cuja eficiência em ensinar seus próprios conteúdos foi verificada com crianças e jovens;
- b) Um *puzzle dinâmico* (*Sauvas in Scratch*) cuja eficiência em ensinar seus próprios conteúdos ainda precisa ser verificada;
- c) Um diário de *game design* narrando a experiência e processo criativo de um *designer de problemas* na construção de um *puzzle dinâmico*;
- d) A literatura sintetizada de treze obras relacionadas com diversas teorias do *game design*;
- e) A proposta de um método embasado em teoria e prática para que *designers de problemas* construam outros *puzzles dinâmicos*;
- f) Uma releitura exemplificada do método de Fellows (1996) sobre a construção de *enigmas* e *estruturas de ação* para jogos digitais.

Para futuras pesquisas, esperamos investigar a partir de diários de *game design* e seus produtos, como profissionais de áreas ligadas à Educação (licenciaturas em Artes, Línguas, Geografia, Física entre outras) constroem de modo individual ou em grupo, jogos digitais para o ensino de seus respectivos temas. Pois embora os produtos de pesquisas envolvendo o desenvolvimento de jogos para ensino de disciplinas específicas venham a público na forma de monografias, artigos ou mesmo disponibilizando-o como produto (MAZIERO, 2014, GUERREIRO, 2015), as frustrações, conflitos, desafios, que fogem de uma descrição acadêmica (por se tratar de uma narrativa pessoal e prática) mas que evidenciam de fato a realidade vivenciada no enfrentamento do *Wicked Problem* que é construir um jogo digital, permanecem ocultas e não

podem ser usadas com sintomas no tratamento de materiais semelhantes (produção de jogos digitais semelhantes).

Outro tema relacionado a produção de jogos digitais em grupos de profissionais ligados à Educação interessante de ser pesquisado, diz respeito à *Co-Engineering*, um tema que discute as atividades concorrentes (trabalhos separados porém diretamente relacionados com a construção do produto), as atividades colaborativas (trabalho de diferentes profissionais em uma relação de confiabilidade e interesse comum a ser alcançado), as atividades cooperativas (relação conjunta no trabalho para consolidar e combinar uma série de elementos individuais). Um grande benefício da *Co-Engineering* é em relação ao custo de gerar um produto estar muito mais relacionado a correções e ajustes, do que no seu conceito, design, desenvolvimento, produção e testes (SIGOGNE, 2016).

De forma semelhante ao problema da *Co-Engineering*, temos na narrativa do diário de *game design* deste trabalho que uma parte significativa do trabalho do *designer de problemas* foi em vão, gerando um material que não pôde ser reaproveitado (desde conceitos, design, desenvolvimento, produtos e testes). Se esta fosse uma produção com capital aplicado, seu custo de produção apareceria como o suficiente para produzir pelo menos mais um jogo digital nesta mesma qualidade. Por esta razão, quando pensamos em trabalhar com a produção de jogos digitais por grupos de profissionais ligados à Educação, nos deparamos com os conflitos semelhantes aos narrados neste diário e ligados a uma falta de investimentos na *Co-Engineering*, trazendo um custo maior (seja em tempo, esforço ou capital) na hora de ajustar e consolidar os trabalhos isolados.

## Referências

- ALMEIDA, A. V.; MAGALHÃES, F. O. **Robert Hooke e o problema da geração espontânea no século XVII**. Revista Latinoamericana de Filosofia e História da Ciência, São Paulo, v. 8, n. 3, p.367-388, 2010. Disponível em: <<http://www.revistas.usp.br/ss/article/view/11209>>. Acesso em: 03 fevereiro 2018.
- AVEDON, E. M.; SUTTON-SMITH, B. **The study of games**. Nova York: John Wiley & Sons, 1971.
- BAILEY, K. **Methods of social research**. Simon and Schuster, 2008.
- BAURU, **Currículo Comum para o Ensino Fundamental Municipal de Bauru**, Bauru: Secretaria Municipal de Educação, 2013.
- BOGOST, I.; FERRARI, S.; SCHWEIZER, B. **Newsgames: Journalism at Play**. Massachusetts: MIT Press, 2010.
- BROUGÈRE, G. **Jogo e educação**. Porto Alegre: Artes Médicas, 1998.
- CAILLOIS, R. **Man, play, and games**. Urbana: University of Illinois Press, 2001.
- CHANDLER, H. M. **Manual de produção de jogos digitais**. [S.l.]: Bookman, 2012.
- COSTIKYAN, G. **Uncertainty in Games**. Massachusetts: MIT Press, 2013.
- FELLOWS, M. R. **The Heart of Puzzling: Mathematics and Computer Games**. In: Proceedings of the 1996 Computer Games Developers Conference, pp. 109–120, 1996.
- GARDNER, H. **The Unschooled Mind: How Children Think and How Schools Should Teach**. New York: Basic Books, 1991.
- GAREY, M.; JOHNSON, D. Computers and Intractability: **A Guide to the Theory of NP-completeness**. San Francisco, W.H. Freeman, 1979.
- GEE, J. P. **Good video games and good learning**. Phi Kappa Phi Forum, p. 33–37, 2005.
- GIL, A. C.. Como elaborar projetos de pesquisa. 4. ed. São Paulo: Atlas, 2002.
- GUERREIRO, M. A. S. **Os efeitos do Game Design no processo de criação de Jogos Digitais utilizados no Ensino de Química e Ciências – O que devemos considerar?** Dissertação (Mestrado) – Universidade Estadual Paulista. Faculdade de Ciências, Bauru, 2015.



- HAMMERSLEY, M.; ATKINSOM, P. **Ethnography: principles in practice**. London: Tavistock, 1983.
- HUIZINGA, J. **Homo ludens: a study of the play element in culture**. Boston: Beacon Press, 1955.
- HUNICKE, R.; LEBLANC, M.; ZUBEK, R. **Mda: A formal approach to game design and game research**. In: Proceedings of the AAAI Workshop on Challenges in Game AI. [S.l.: s.n.], 2004. p. 04–04.
- JUUL, J. **The art of failure: An essay on the pain of playing video games**. [S.l.]: Mit Press, 2013.
- KAMII, C.; DEVRIES, R. **Jogos em grupo na educação infantil**. São Paulo: Trajetória Cultural, 1991.
- KOSTER, R. **A Theory of Fun for Game Design**. [S.l.]: O'Reilly, 2014.
- KRULIK, S.; RUDNICK, J. A. **Reasoning and Problem Solving: A Handbook for Elementary School Teachers**. Allyn and Bacon, 1993.
- MATEAS, M., STERN, A. **Build it to understand it: Ludology meets narratology in game design space**. Proceedings of the Digital Interactive Game Research Association (DiGRA), 2005.
- MAZIERO, H. **Jogos Digitais no Ensino de Matemática - Um instrumento de diagnóstico das concepções dos alunos sobre diferentes representações dos números racionais**. Dissertação (Mestrado) – Universidade Estadual Paulista. Faculdade de Ciências, Bauru, 2014.
- NAKAMURA, J., CSIKSZENTMIHALYI, M. **The concept of flow**. In Synder, C. R., & Lopez, S. J. (Ed.). Oxford University Press, 2009.
- NCTM, National Council of Teachers of Mathematics. **Principles and Standards for School Mathematics**. Reston, VA, 2000.
- PETKOVIC, M. S. **Famous puzzles of great mathematicians**. Providence: American Mathematical Society, 2009.
- RITTEL, H. W. J., WEBBER, M. M. **Dilemmas in a General Theory of Planning**, in Policy Sciences 4, Elsevier Scientific Publishing, Amsterdam, pp. 155-159, 1973.
- ROGERS, S. **Level Up!: Um guia para o design de grandes jogos**. [S.l.]: Blucher, 2012.
- SALEN, K.; ZIMMERMAN, E. **Regras do jogo: fundamentos do design de jogos: principais conceitos: volume 1**. São Paulo: Blucher, 2012a.

- SALEN, K.; ZIMMERMAN, E. **Regras do jogo: fundamentos do design de jogos: regras: volume 2**. São Paulo: Blucher, 2012b.
- SALEN, K.; ZIMMERMAN, E. **Regras do jogo: fundamentos do design de jogos: interação lúdica: volume 3**. São Paulo: Blucher, 2012c.
- SALEN, K.; ZIMMERMAN, E. **Regras do jogo: fundamentos do design de jogos: cultura: volume 4**. São Paulo: Blucher, 2012d.
- SCHUYTEMA, P. **Design de Games: Uma abordagem prática**. [S.l.]: Cengage Learning, 2008.
- SELLTIZ, C.; et al. **Métodos de pesquisa nas relações sociais**. São Paulo: Herder, 1967.
- SIGOGNE, A. et. al. **Co-Engineering: A Key-Lever of Efficiency for Complex and Adaptive Systems, Throughout Their Life Cycle**. G. Auvray et al. (eds.), Complex Systems Design & Management. Springer International Publishing Switzerland. 2016.
- SILVA, M. H. P. D.; CÂMARA, T. R. Exploração Halloween, um exemplo de que não é preciso saber programar para desenvolver um jogo virtual. In: **Anais do XI Encontro Nacional de Educação Matemática**. Curitiba: SBEM, 2013.
- SILVA, M. H. P. D. Jogo digital para o ensino de métodos numéricos. In: ENCONTRO REGIONAL DE MATEMÁTICA APLICADA E COMPUTACIONAL, 4., 2017, Bauru. **Caderno de trabalhos completos e resumos**. Bauru: Unesp, Faculdade de Ciências, 2017a. p.170-177. Disponível em: <<http://www.fc.unesp.br/#!/departamentos/matematica/eventos2341/ermac/cadesnos-de-trabalhos-completos-e-resumos/>>. Acesso em: 16. nov. 2017.
- SILVA, M. H. P. D. A aprendizagem de métodos numéricos com o jogo digital Handles in Scratch 2.0. In: **Revista Eletrônica Paulista de Matemática**, Edição ERMAC, volume 10, dez. 2017. Bauru: Universidade Estadual Paulista “Júlio de Mesquita Filho”, 2017b.
- SMOLE, K. S., DINIZ, M. I., CÂNDIDO, P. **Jogos de Matemática de 1º ao 5º ano** (Série Cadernos do Mathema – Ensino Fundamental). Porto Alegre: Artmed, 2007.
- TAYLOR, D., PROCTER, M. **The literature review: a few tips on conducting it**. Disponível em <<http://advice.writing.utoronto.ca/types-of-writing/literature-review/>> acesso em 18/04/2018.
- WING, J. M. **Computational thinking**. Commun. ACM, v. 49, n. 3, p. 33–35, 2006.
- ZABALZA, M. A. **Diários de Aula: Contributo para o estudo dos dilemas práticos dos professores**. 1. ed. Porto: Porto Editora, 1994.

ZASLAVSKY, C. **Mais jogos e atividades Matemáticas do mundo inteiro: diversão multicultural a partir de 9 anos**; tradução Adriano Moraes Migliavaca. Porto Alegre: Artmed, 2009.

ZIMMERMAN, E. **Play as research: the iterative design process**. In: Design Research: methods and perspectives, 2003.

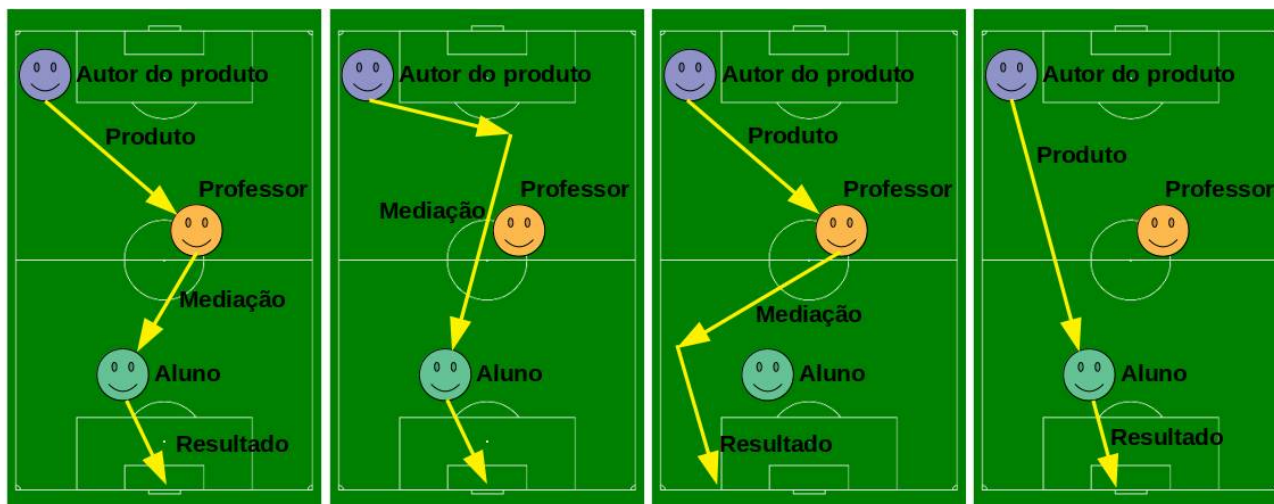
ZYDA, M. **Creating a science of games**. COMMUNICATIONS-ACM 50.7 (2007): 26.

## Apêndices

### 1. Diário de *game design*

Em março de 2016 ingressei no mestrado em Ensino de Ciências. Escolhi estudar com meu orientador o tema Pensamento Computacional. A princípio pensei naquilo que seria mais conveniente, elaborar um curso sobre o assunto, avaliar professores da Educação Básica antes e depois, para então averiguar a eficiência das explicações e do método. Após algumas conversas percebi que seria mais interessante trabalhar dentro de um produto, algo externo que pudesse ser avaliado. Novamente pensando na linha mais simples, imaginei uma sequência didática voltada para professores da Educação Básica ensinarem o Pensamento Computacional aos seus alunos. Contudo, trabalhar com um produto a ser mediado, envolvia uma preocupação dupla, visto que tanto o produto quanto a forma como os mediadores aprendem precisariam ser muito bem afinados.

Nesse sentido, o objetivo desta pesquisa não era tentar em dois anos mudar o mundo. Assim, embora reconheça o valor de um professor como mediador para a aprendizagem de um conteúdo, sua interferência dificultaria a compreensão dos requisitos do produto. Dado que nas condições ideais o produto estaria adequado ao professor que realiza uma mediação eficiente, fazendo com que o aluno apresente os resultados adequados, validando a funcionalidade do sistema. Porém em um quadro menos otimista, as deficiências do produto poderiam ser corrigidas durante a mediação do professor, fazendo com que o aluno apresente os resultados adequados, nos levando a validar um sistema falho. Por outro lado, um produto com os requisitos necessários poderia ser mediado de forma errada, provocando um resultado inadequado, levando a uma reformulação equivocada do sistema. Mas, em uma abordagem direta do produto com o aluno, tomamos um resultado que nos indica a necessidade ou não de uma reformulação do mesmo.



**Figura 1:** Esquema de investigações dos efeitos do produto dado os resultados (fonte própria).

Dessa forma, o objetivo desta pesquisa passou a ser o desenvolvimento, a afinação e validação de um produto capaz de desenvolver diretamente no público infantojuvenil algumas das habilidades e competências comuns ao Pensamento Computacional. Lidando com esta relação do ser humano trabalhando em parceria com os computadores, comecei a imaginar como deveria ser um jogo com estéticas semelhantes a usar uma calculadora. No sentido de que realizar grandes cálculos no papel é tedioso e trivial (a menos de erros processuais). Também a calculadora sozinha não resolve nenhum problema de Matemática. Mas quando usamos uma calculadora, podemos resolver problemas de Matemática apenas tomando decisões. Atribuindo à máquina todo o processo mecânico de cálculo, e nos preocupando somente com a organização da resolução do problema.

Um exemplo desta distribuição de tarefas (entre ser humano e computador) pode ser a partir de uma empresa, da qual o jogador é o patrão que dá as ordens ao seu funcionário, neste caso a calculadora. Esta executa cegamente as ações e devolve o resultado solicitado. Embora pareça que todo o desafio estivesse na realização dos cálculos, na verdade o motivo para solicitá-los é o real problema do jogo. Por exemplo, em uma situação onde preciso montar um cubo maciço com 55 peças unitárias em cada dimensão, sendo cada 11 peças no valor de 1 real. Preciso determinar o custo total desta construção antes de solicitar a verba da instituição. Se for pedido menos dinheiro do que o necessário, o patrão precisará pagar do bolso a diferença. Se for pedido mais dinheiro do que o necessário, o patrão poderá ser acusado de desvio. O funcionário que calcula tem a garantia de realizar os cálculos corretamente (a menos de um erro em seu *software*). Assim, embora o encargo de calcular seja do funcionário, o motivo destes cálculos é o objetivo fim, que continua no patrão, junto com as consequências de uma escolha errada ser tomada durante o processo.



**Figura 2:** Esquema de resolver problemas tomando decisões (fonte própria).

Então, nessa perspectiva planejei um jogo cujo contexto seria um **Escritório de Calculistas** em analogia a um período anterior as calculadoras. Neste cenário haveria um patrão, que seria o jogador, e seus funcionários que seriam a própria interface do jogo. No exemplo abaixo o patrão é o personagem Wilson, e seus funcionários controlados pelo computador são Marc, Kate e r2d2. Todo funcionário remete um custo ao patrão, assim como o estabelecimento e multas caso não consiga cumprir um serviço dentro do prazo de entrega. Logo, o jogador precisa manter seu negócio funcionando com lucro, senão terá que demitir funcionários ou reduzir o espaço do seu estabelecimento, que também compromete o número máximo de funcionários que ele pode suportar.



**Figura 3:** Interface exemplo do jogo **Escritório de Calculistas** (fonte própria).

O jogador pode a qualquer momento analisar suas condições e a de seus funcionários pelos quais achou interessante contratar. **Marc** domina bem as quatro operações na velocidade de 180 cálculos por hora e cobra 120\$ por dia de serviço. **Kate** além de dominar as quatro operações, também sabe trabalhar com potências, na velocidade de 90 cálculos por hora e cobra 150\$ por dia de serviço. **r2d2** é capaz de realizar apenas somas, na velocidade de 1000 cálculos por hora e cobra 210\$ por dia de serviço. O aluguel do seu escritório custa ao patrão 100\$ por dia de atividade.

A interação neste jogo ocorre mediante propostas de serviços que chegam ao patrão. Por exemplo, o reitor de uma universidade traz a medida de cada calçado dos ingressos nos cursos de exatas desde 2009 a 2018, no total de 4500 medidas. Ele solicita que em até 24 horas seja entregue a média dos sapatos por ano e oferece pelo serviço 800\$, porém quer uma indenização de 100\$ por hora de atraso pelo serviço até um máximo de 8 horas, após isto o pedido é cancelado, mas a multa permanece.

Assim, o patrão analisa como poderia resolver este problema nas suas atuais condições: *Não tenho nenhum outro trabalho em andamento, então os funcionários estão livres para trabalhar neste. A **r2d2** é rápida e erra pouquíssimo, ela conseguiria em 4 horas e meia realizar 4500 somas, que é o necessário para obter o total de medidas de todos os anos. Então, passando a tarefa de divisão para o **Marc**, que é apenas um cálculo, o serviço estaria completo em aproximadamente 4 horas e 30 minutos.*

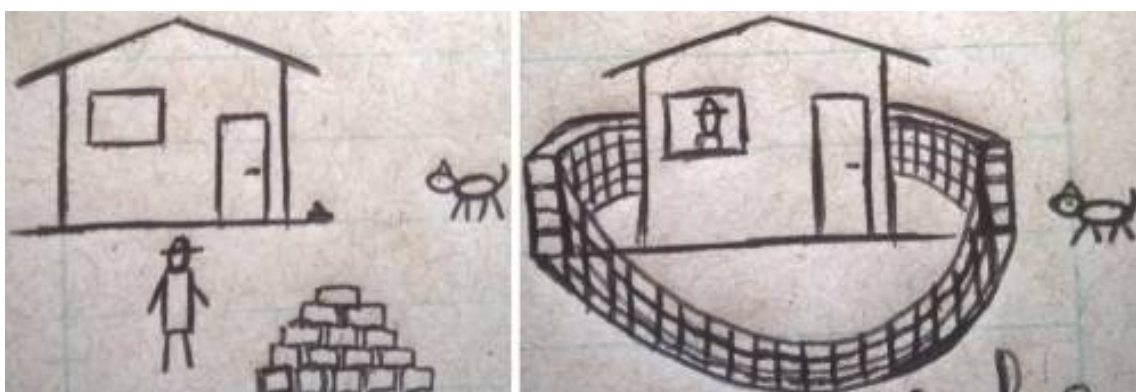
Outra forma de pensar seria: *Em 4 horas o **Marc** é capaz de resolver 360 somas e a **Kate** 180 somas. Assim, se passar para a **r2d2**, 3960 somas para serem feitas, deixar 360 com o **Marc** e 180 com a **Kate**. Em apenas 4 horas teremos concluído as 4500 somas necessárias, restando apenas fazer uma divisão, o que permite que o trabalho todo seja entregue em aproximadamente 4 horas.*

Porém o **Escritório de Calculistas** se aproximava mais de uma gestão empresarial do que de desenvolver as habilidades do Pensamento Computacional. Coloquei esta ideia de lado também pela dificuldade envolvida na interface com o usuário e no *software* que gerenciasse os serviços e processos.

Formulando outro jogo, procurei investir nas dinâmicas que solucionassem problemas simples e não triviais a partir da tomada de decisões. Para isso pensei em um quadro bastante amplo onde o personagem do jogador seria um boneco quase de palitinho, com sua região peitoral no formato retangular, que corresponderia a espaços para colocar comandos. Assim, a medida que o personagem fosse evoluindo dentro do jogo, este retângulo que inicialmente seria bem pequeno, aumentaria. Isto tornaria o personagem mais alto, dando também a impressão de que ele inicia

como uma criança e vai amadurecendo de acordo com suas conquistas. Apenas como forma de registro, na época batizei este jogo de **As aventuras do garoto-mate**.

O personagem teria uma casa onde haveria uma mesa de planejamentos. Nessa mesa ele poderia montar funções específicas a partir da combinação de comandos. Tanto o tamanho do retângulo do personagem, quanto o tamanho da mesa seriam condições aprimoráveis no decorrer do jogo, dado que uma mesa grande facilita a construção de funções bem mais complexas. E um personagem com um retângulo grande pode receber um número maior de ações a realizar. Um exemplo destes desafios enfrentados seria lidar com um cachorro que constantemente vem sujar os arredores da sua casa. Uma solução para isto seria a partir da construção de um muro ao seu redor. Esta ação seria dada pelo jogador ao personagem a partir de estruturas iterativas similares a “for (i=0; i<10; i++)”. Embora o jogador também possa realizar a construção manualmente, o intuito destes desafios é serem repetitivos e cansativos de serem realizados, incentivando assim o uso de rotinas automáticas.



**Figura 4:** Situação do problema do cachorro em **As aventuras do garoto-mate** (fonte própria).

Contudo, esta ideia, foi colocada de lado por conta das dificuldades previstas na elaboração da interface com o usuário e na construção do *software* que gerenciasse este personagem, dado que até a ocasião planejava construir toda sua estrutura na linguagem C++ da qual apresento pouco domínio (este jogo será retomado em **Handles in Scratch 2.0**).

Buscando algo viável de ser construído nas minhas atuais condições, reduzi ao máximo o contexto do jogo até a ideia de trabalhar dentro de uma planilha eletrônica. A vantagem desta abordagem estava no *software* da planilha conter funções Matemáticas, permitir as entradas de dados ou fórmulas pelo usuário e mostrar facilmente os valores de saída, além de que, minhas habilidades permitiam construí-lo. Assim, o jogo envolveria a resolução de problemas Matemáticos com pequenos grupos de dados, onde o jogador teria as próprias funções de cálculo da planilha para



auxiliá-lo. Para identificação do projeto, este jogo ficou sendo **ExQuest**, em relação ao famoso *software* de planilhas eletrônicas, junto com a palavra *Quest* que do inglês significa missão. No protótipo construído, o jogo pedia o apelido do jogador, que após inserido apresentava as células da planilha dando boas vindas ao “apelido”. Apresentando o personagem na forma de boneco de palitinho, então iniciando o **Episódio 1: o chapeleiro**. Onde há um problema de encontrar médias de 40 valores localizados na linha R da própria planilha. Os dados do problema são sorteados na ocasião em que o problema é descrito. Como gratificação o chapeleiro oferece um chapéu ao personagem do jogador. Abaixo do problema há espaço para uma resposta. O jogo somente continuará quando a resposta inserida for a correta para os 40 valores recém-sorteados nesta iteração do jogo.

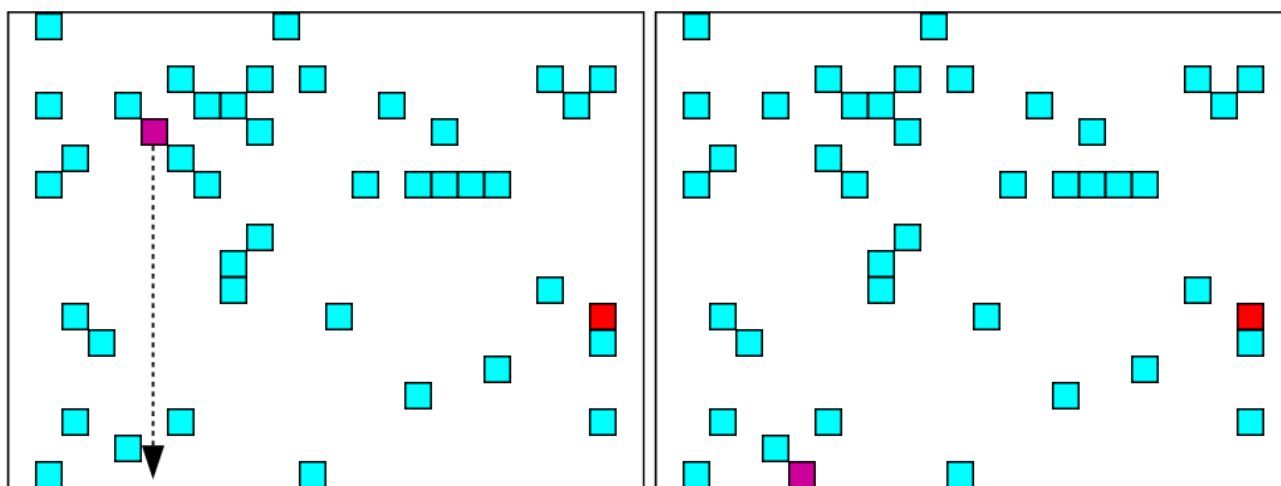
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	Nickname:	Nantathu																38
2	Welcome to the game Nantathu																	39
3	Este é seu personagem		+	0														40
4																		43
5																		48
6	Episódio 1: o chapeleiro		+	0	x	s		+	0									38
7	Anotei 40 medidas (em R1 até R40) de cabeças dos meus clientes,																	35
8	se achar a média para mim, eu lhe dou o chapéu que você quiser																	50
9	Resposta:																	37

**Figura 5:** Interface gráfica construída no jogo **ExQuest** (fonte própria).

O projeto foi abandonado tanto pela dificuldade em construir uma interface gráfica com personagens mais elaborados, como também pela planilha eletrônica permitir que o jogador veja ou edite os conteúdos de células invisíveis. Embora este projeto tivesse bastante potencial de avaliação dos jogadores, analisando como estes trataram e resolveram os problemas, preferi trabalhar em uma ideia semelhante a manipulação e disposição dos dados, mas em um contexto que utilizasse uma dinâmica mais semelhante a mecânica de jogos de ação. Como mover, arrastar, deslocar objetos a partir de direcionais do teclado, por exemplo em vez de digitar fórmulas, selecionar regiões de dados a partir de operações específicas.

Na direção de construir algo mais manipulativo e dinâmico, comecei a trabalhar com a linguagem de programação *Scratch 2.0*. Já havia alguma experiência com esta linguagem ensinando crianças a construírem jogos digitais, mas a estrutura era bastante simples e voltada para a compreensão das estruturas necessárias para cada tipo de jogo. Assim, resgatando esta experiência iniciei um trabalho em uma dinâmica similar a pequenos desafios presentes em vários jogos digitais, os chamados **Box-sliders**. Onde o jogador precisa com seu personagem empurrar algumas caixas ou objetos do cenário até que atinjam determinadas posições para abrir uma porta. No básico desta estrutura, construí uma malha quadriculada onde gerava ao acaso bloquinhos azul-claro para

servirem de bloqueios. O bloquinho vermelho era o objetivo a ser alcançado enquanto que o bloquinho roxo se move de acordo com a setinha pressionada pelo jogador até que colida com um bloquinho azul-claro ou com a fronteira da tela. Experimentando este jogo na versão onde os bloquinhos azul-claro são gerados em posições aleatórias, comecei a notar que por vezes o nível não tinha solução. Ou seja, o bloquinho vermelho estava em uma posição que era impossível de ser alcançada pelo bloquinho roxo. Também percebi que haviam espaços onde se o bloquinho roxo entrasse, seria impossível de sair. Assim, em vez de determinar algumas distribuições para os bloquinhos azul-claro que permitissem a solução do problema, considerei bem mais interessante interromper a construção do **Box-slider** tradicional para gerar um **Quasi-box-slider**. Nesse sentido o jogador precisaria lidar não só com uma tomada de decisões que o levariam até a solução (bloquinho vermelho), mas também as consequências de suas decisões (posições impossíveis de escapar ou posições que é impossível retornar), assim como o problema do nível não apresentar solução na atual instância do jogo, ou mesmo não apresentar solução independente das decisões que fossem tomadas. O desafio de resolvê-lo ou determinar sua impossibilidade de solução parecia muito mais interessante do que apresentar um problema com uma solução prevista.



**Figura 6:** Protótipo funcional do jogo **Quasi-box-slider** (fonte própria).

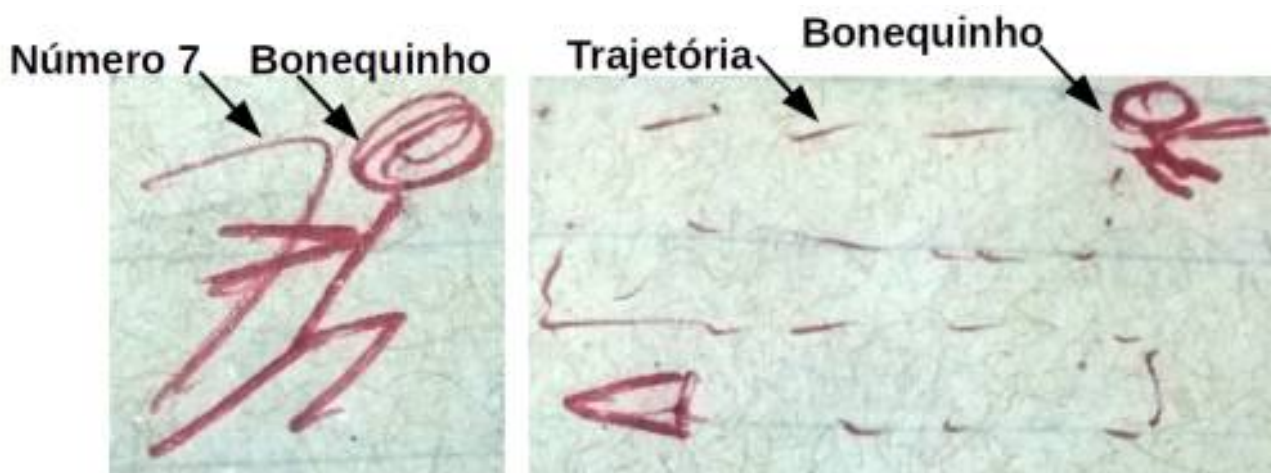
Embora utilizasse bem o princípio de tomar decisões para resolver um problema, ainda havia a falta de uma “interferência” no sistema. Como se o jogador para realmente desenvolver as habilidades do Pensamento Computacional dentro deste sistema, precisasse do poder de alterar as condições e regras de movimentação dentro de certos limites. Como se o jogo fosse uma unidade editável, onde a sua solução possa ser construída a partir de ajustes das próprias configurações do jogo, tornando soluções impossíveis em possíveis. Ficou como sugestão a criação de novos blocos ajustáveis que pudessem ter efeitos de acordo com o número de impactos sofridos pelo bloquinho

roxo colidindo nele. Como se fosse um sistema cíclico onde o jogador a partir das colisões conseguisse configurá-lo para resolver seu problema impossível ou mesmo simplificá-lo.

Apesar deste jogo parecer ideal para trabalhar as habilidades do Pensar Computacionalmente, faltava-lhe uma ação mais direta com algum conteúdo curricular. Dado que avaliar o desenvolvimento do Pensamento Computacional é extremamente complicado, o mais simples no caso seria identificar habilidades do currículo comum escolar. Entre elas, na nossa preferência e por conta de minha própria formação inicial, a Matemática. Assim uma questão que precisava acrescentar ao jogo era a presença de um subgrupo do conhecimento curricular Matemático que fosse facilmente identificado e com o qual pudesse trabalhar dentro de uma avaliação escrita.

## Pega e Larga

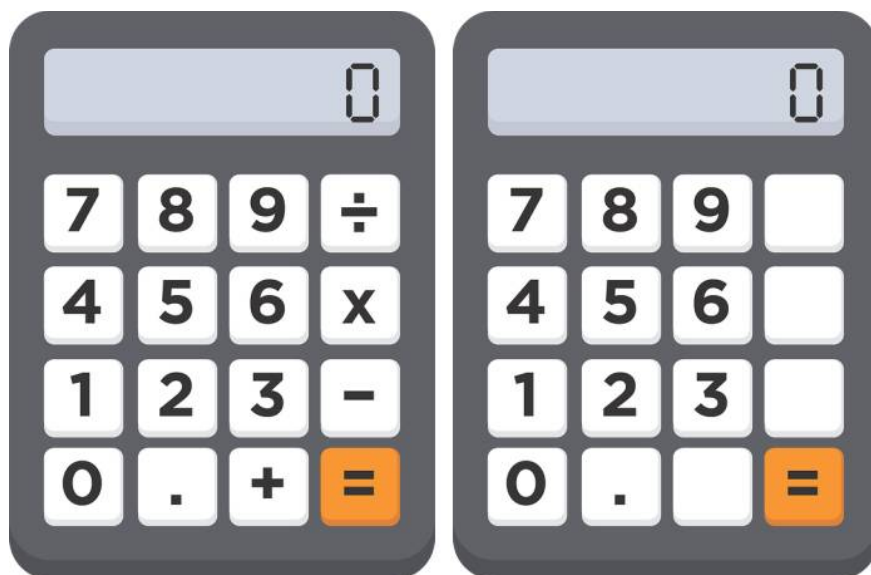
Sem descartar a ideia raiz do **Quasi-Block-Sliders**, de movimentar objetos pela tela, alterar configurações do jogo e encontrar uma solução para um problema que pode ou não ter solução nas suas configurações iniciais. Também aproveitando o protótipo do **ExQuest**, onde os dados estão dispostos na tela e podem ser selecionados quase que manualmente (no caso, usando o mouse), imaginei um jogo com um grupo de dados a serem movidos pelo jogador. De modo quase similar ao arrastar blocos, mas com um personagem protagonista sendo o responsável pelo seu deslocamento. Como se ele os arrastasse/puxasse por uma determinada trajetória estipulada pelo jogador. Neste contexto, como o objetivo do jogo seria a movimentação da informação, nem ela e nem o personagem poderiam ser o destaque da tela. Todo o contexto deveria ser simplista, ressaltando a própria configuração do problema composto pelos elementos personagem e números. Dessa forma, tendo eu uma preferência natural por desenhos simplistas, comecei a esboçar esta dinâmica a partir de bonecos de palitinho. Abaixo, imagens dos primeiros esquemas rascunhados com esta ideia.



**Figura 7:** Esquemas iniciais do jogo **Pega e Larga** (fonte própria).

A movimentação dos dados e a forma do personagem era somente a base para começar o ataque a este problema. O segundo aspecto a se observar foi então que tipo de desafios posso acrescentar em um jogo onde o personagem mova números de um lado para o outro da tela? Nesse sentido, o mover números pode ter algumas funções práticas, ou ele está abrindo campo visual de outro objeto, mas para isto não faria sentido usar números, e sim outros tipos de figuras mais relacionadas, como pedras ou plaquinhas por exemplo. Outra função para esta movimentação de números é a seleção deles de acordo com um problema. Mas isto reduziria a uma busca pela resposta correta a um problema dentre várias alternativas. Embora esta fosse uma abordagem possível, ela já é muito comum por ai afora. Então comecei a procurar outros tipos de desafios que pudessem ser tratados a partir da escolha de determinados números.

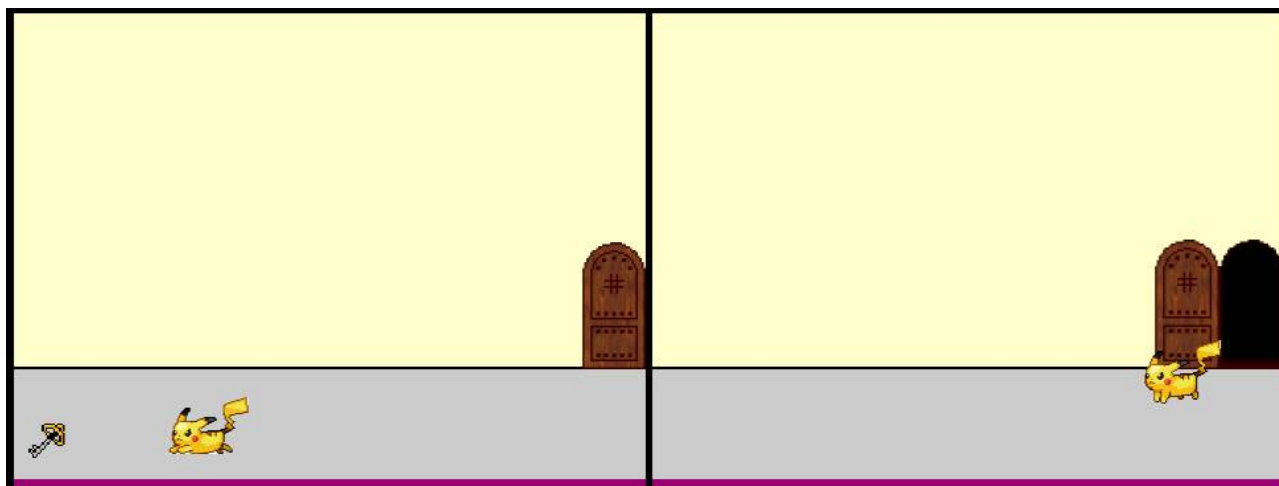
Um problema que já vi em diversos lugares com vários nomes diferentes é a chamada **Calculadora maluca**. Onde por vezes este problema vem na forma textual, ou no formato digital iterativo como se fosse mesmo um jogo. Mas sua ideia é a seguinte, temos uma calculadora convencional, porém algumas de suas teclas estão apagadas. O desafio envolve consertá-la de acordo com algumas experimentações. Ou seja, se as teclas de números estão apagadas, basta apertar uma a uma e saberemos qual número cada tecla representa. Porém no caso das operações estarem apagadas a situação se torna mais complicada. Pois para descobrirmos que operação uma tecla representa precisamos selecionar dois números e operá-los, para então de acordo com o resultado obtido determinar qual operação realizaria aquilo. No caso isto pode levar a algumas ambiguidades, como  $0-0 = 0+0 = 0*0$ , ou  $1*1 = 1/1$ , ou  $2+2 = 2*2$ , ou  $4-2 = 4/2$ . Assim, escolhendo com algum cuidado os valores a serem operados, podemos determinar quais das quatro operações básicas estão sendo realizadas e assim “consertar” a calculadora.



**Figura 8:** Calculadora e suas teclas omissas do problema **Calculadora Maluca** (fonte própria).

O interessante deste problema para a dinâmica de arrastar os números pela tela, é que dado um local para a entrega dos números (equivalente a digitar um número na calculadora), podemos aplicar sobre o número uma operação desconhecida e apresentar ao jogador seu resultado. E ao final do processo de inserção destes números, o jogador precisaria determinar a partir dos resultados qual operação estava sendo realizada. Este contexto embora pareça simples pode ser facilmente ampliado ao utilizarmos, por exemplo, uma calculadora científica com suas operações omissas. A análise das ambiguidades aumentaria muito o esforço para determinar o comportamento da função. O que no caso das quatro operações básicas poderia ser determinado com somente dois “bons números”, no caso de uma calculadora científica a situação além de exigir um pleno conhecimento das funções Matemáticas da calculadora, também pediria uma escolha apropriada dos valores com o intuito de reduzir o espaço de possibilidades desta operação desconhecida.

No caso do jogo, estes dados deveriam ir até um *check-point* com uma função específica (mas desconhecida), que calcula estes dados e apresenta ao jogador o resultado (similar ao processo de arrastar ou selecionar um conjunto de dados numa planilha eletrônica). Então no jogo, após inserir os números nesta parte responsável por calcular, e observar como o valor se altera, o jogador precisaria responder qual operação estava sendo realizada. Para construir estas dinâmicas pensei em aproveitar o jogo **Abra a porta**, desenvolvido em *Scratch 2.0* que já havia produzido para ensinar as crianças na elaboração de estruturas condicionais como uma porta que exige para ser aberta, que uma chave seja coletada anteriormente.

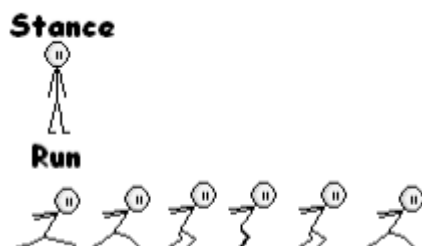


**Figura 9:** Jogo **Abra a porta** (fonte própria).

No caso do novo jogo, em vez de abrir a porta, a entrega de todos os números para a “função que os calcula” abriria uma janela de pergunta para o jogador responder qual operação ele imagina ser realizada. Por último, considerando que a interação com o jogador envolveria principalmente pegar números e soltá-los neste local específico, decidi intitular o protótipo deste novo jogo como **Pega e Larga**.

**Pré-produção:** Ainda em março do mesmo ano, comecei a formular o plano diretor deste jogo. Tinha em mente o jogo **Abra a porta** como base, assim como o personagem carregaria números, considerei usar uma forma mais similar a função, algo com pernas compridas, bracinhos e aparência humana simplificada. Resumindo, um boneco de palitinho como já aparecia nos rascunhos da dinâmica de movimentar números (figura 7). A escolha por um jogo bidimensional com visão de perfil teve por intuito tanto a facilidade de sua construção na linguagem *Scratch 2.0*, como também uma disposição dos objetos do problema (personagem, números e máquina) de modo mais claro ao jogador, evitando assim a ocultação de elementos importantes.

Então, iniciei uma busca por um personagem desenhado em palitinho que tivesse um bom perfil para este jogo. Como somente possuía fins de pesquisa com este produto, não me ative a procurar em galerias disponíveis na internet, como a do artista Gigobyte98 no seu portfólio hospedado no site <https://www.deviantart.com/>. Embora desenhar um boneco de palitinho não seja uma tarefa difícil, em virtude de minhas próprias habilidades deficientes neste aspecto, achei preferível o uso de um personagem já estruturado e bem produzido. Dessa forma, analisando as 80 posições diferentes deste mesmo bonequinho disponíveis no documento *sprite-sheet* do artista. Tomei 7 delas como necessárias para a movimentação na tela e o transporte dos números.



**Figura 10:** Posições do personagem escolhido do artista Gigobyte98.

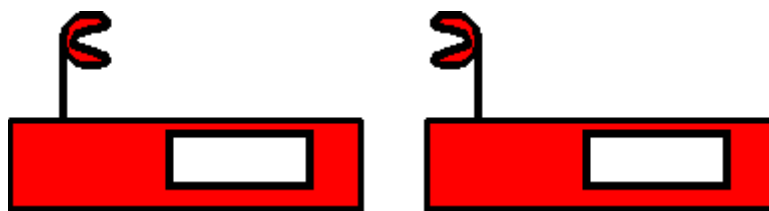
A escolha pelos números foi muito mais simples, dado que no próprio acervo da linguagem *Scratch 2.0*, já temos uma série de letras e números em diferentes formatos. Assim, tomei números escritos no formato pixelado variando de 0 a 9. A razão do formato destes números é a facilidade na sua identificação se comparada a escrita cursiva ou outras fontes de texto. Além de que seu formato pixelado mantinha o mesmo padrão de forma com o personagem. Tornando este visualmente mais próximo de um jogo retrô.



**Figura 11:** números de 0 a 9 disponíveis no acervo do *Scratch 2.0*.

Contudo para determinar o objeto que realizaria a operação Matemática desconhecida, pensei sobre algo indefinido e que tipicamente seria reconhecido como a “máquina”. Como fã da série *Doctor Who*, recordei-me da temporada de 2011, onde o protagonista (chamado somente de Doutor), assume um dia comum de trabalho na empresa em que seu amigo humano trabalha. Então, em pouco tempo na mesa do Doutor já havia uma máquina estranha construída com materiais extremamente simples, com duas espátulas de cozinha em cima como se fossem antenas que ficavam girando como se buscassem uma melhor recepção de sinal. O funcionamento daquilo não entrava em questão no episódio, contudo um simples relance naquele objeto estranho com uma antena em movimento já significa se tratar de uma máquina. Talvez muito mais significativa visualmente do que os atuais gabinetes dos computadores, ou mesmo os notebooks. Um sensor em constante movimento acima de um objeto de formato chamativo para o ambiente, seriam indícios claros daquilo que realiza as operações desconhecidas neste jogo. Assim, a fiz no próprio editor de imagens do *Scratch 2.0*, na cor vermelha para chamar maior atenção dos jogadores. Sua antena da mesma cor e seu formato retangular para ilustrar algo objetivo (na minha imaginação algo mais arredondado poderia significar uma maior flexibilidade), o que vai de acordo com a intenção da

máquina realizar as operações a partir dos números recebidos. O espaço em branco em seu interior é destinado a expor o valor inicial e pós operação. Similar a tela de uma calculadora convencional que a cada nova entrada, atualiza ao usuário o valor resultante.



**Figura 12:** A primeira máquina (fonte própria).

Após as entradas dos números na máquina, o jogador precisaria determinar qual é a operação realizada. Isto ocorreria a partir de uma janela de pergunta com espaço para inserção de uma resposta escrita pelo jogador. Este recurso (janela de pergunta e resposta) já é disponível na própria linguagem *Scratch 2.0*, tornando esta implementação bastante simples. Sem necessidade de criar uma imagem de “janela” para a interação com o jogador. Em linhas gerais, determinamos os seguintes aspectos visuais necessários para o jogo:

- a) O jogo será bidimensional com visão de perfil;
- b) O protagonista será um boneco de palitinho;
- c) A parte do jogo que recebe e faz operações (chamada de máquina) será de cor quente (vermelha, laranja, amarelo) com formato retangular, terá uma antena em constante movimento e um visor marcando seu valor atual;
- d) Os números serão pixelados e terão a altura do personagem;
- e) Quando todos os números forem inseridos na máquina, abrirá uma janela perguntando a operação que estava sendo realizada, com um espaço para o jogador digitar a resposta.

No jogo **Abra a porta** a chave era única, quando tocada ela sumia e enviava um aviso para a porta abrir em contato com o personagem. Já no **Pega e Larga**, o número precisaria assumir diferentes valores entre 0 e 9. Assim uma alternativa neste caso foi utilizar um único objeto, com 10 imagens diferentes associadas. Então, quando o jogo iniciar o objeto original é escondido dos olhos do jogador, e são criados clones deste número, associando a cada um uma imagem diferente e referente ao seu valor. O mesmo valor permanece em uma variável própria ligada aquele clone. Esta abordagem tem como vantagem a possibilidade de organizar o código por trás do comportamento do número em um único objeto. Os clones agem como se fossem objetos próprios, porém são gerados dentro da própria ativação do jogo. Sendo mais simples associar N imagens a um objeto do que coordenar o comportamento de N objetos com códigos próprios.



Outra abordagem simplista é o problema de transportar números. Para isso, cada clone de um número possui uma função gatilho que se ativa quando tocada pelo personagem e quando uma variável que indica se o personagem já carrega algum número for falsa. Esta função quando ativa, troca repetidas vezes a posição do número para a mesma posição do personagem, até que o número toque na máquina. Quando isto ocorre, seu valor é inserido na máquina e o clone é deletado, evitando que este objeto volte a interferir na resolução do problema. Em termos práticos esta estrutura da impressão ao jogador de que o número foi agarrado pelo personagem, e quando toca na máquina ele foi inserido nela. O que é a intenção proposta nesta dinâmica.

A posição inicial da máquina na tela tinha por intuito manter a tela do problema limpa o bastante para que o jogador selecionasse os números e os levasse até a máquina sem a preocupação de tocá-lo na máquina ao acaso. Assim, decidi posicioná-la no canto inferior direito da tela, e todos os números seriam dispostos em posições espalhadas pela tela que não coincidam com a localização da máquina.

A movimentação do personagem é orientada pelos direcionais do teclado. No caso do deslocamento horizontal, as imagens do personagem movimentando as pernas são trocadas na mesma sequência, dando a impressão de que ele está andando. Contudo a movimentação vertical será estática, embora não pareça tão natural, soa melhor do que se rotacionasse o personagem para que ele andasse nas paredes. A opção de fazê-lo andar para cima de perfil também não fica natural a menos que o gráfico do jogo seja o chamado 2.5D, que é um gráfico 2D inclinado, produzindo uma impressão de tridimensionalidade.

Uma outra preocupação foi com a quantidade dos números na tela. Dado que os números seriam sorteados entre 0 e 9, há o risco dentro do problema da **Calculadora maluca** de números cujas operações apresentam resultados ambíguos. Assim, por exemplo, se disponibilizo ao jogador dois números, situações como já discutidas anteriormente podem apresentar resultados dos quais é impossível determinar qual operação é realizada. Contudo ao acrescentarmos um dado valor inicial ao visor da máquina, isto conta como um valor extra a ser operável. Assim, escolhi três números a serem levados para a máquina. Embora ainda existam casos onde a solução não possa ser determinada, a chance disto ocorrer seria pequena o bastante para resolvermos os desafios desta dinâmica em quase todas as ocasiões.

Por uma questão de simplicidade, para não precisarmos lidar com números racionais, decidimos não utilizar a operação de divisão entre as opções, reduzindo-as a adição, subtração e multiplicação. De forma geral, para construir estas interações precisei considerar os seguintes aspectos lógicos no jogo:

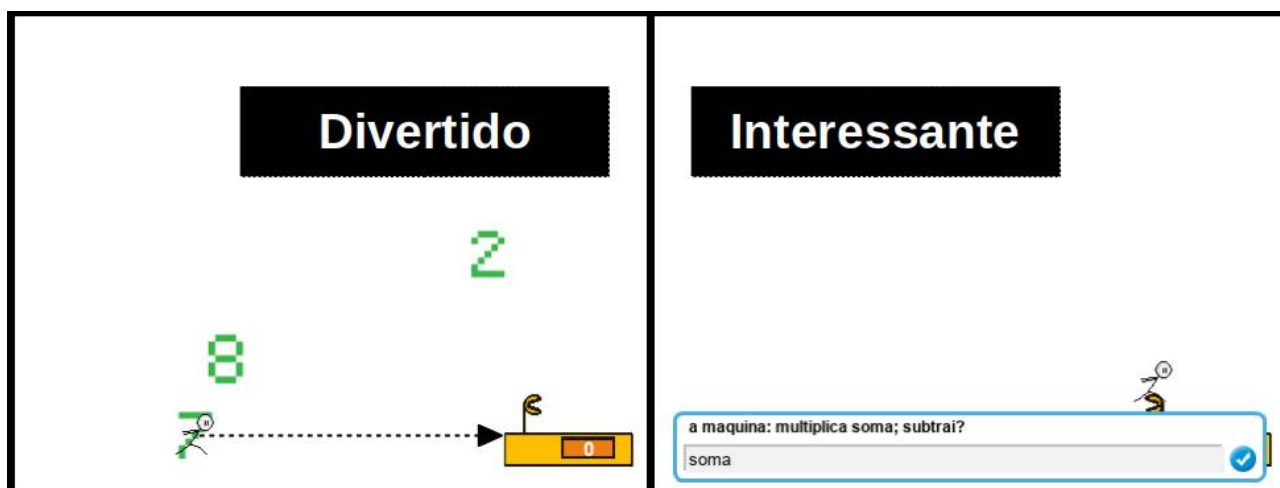
- a) Todos os níveis têm 3 números entre 0 e 9 escolhidos aleatoriamente e espalhados pela tela, uma máquina no canto inferior direito e o personagem do jogador localizado à esquerda da máquina;
- b) O jogador controla o personagem através dos respectivos direcionais do teclado;
- c) O primeiro número que encostar no personagem ficará com ele, até o número tocar na máquina;
- d) Quando o personagem carrega um número, outros números que encostam nele, não ficam presos;
- e) O visor da máquina possui um valor inicial predeterminado ao início de cada nível;
- f) Cada vez que um número toca a máquina, ela realiza com seu valor atual uma operação Matemática (multiplicação; soma; subtração) utilizando o número recebido e atualiza o visor.
- g) Embora a janela de respostas aceite qualquer sentença escrita pelo jogador, somente “multiplicação”, “soma” ou “subtração” estão entre as respostas válidas;
- h) Se a resposta for correta, o jogador avança ao próximo nível, senão o nível atual se reinicia;
- i) O jogo não tem fim, a máquina sorteia uma operação e um valor inicial a cada novo nível.

**Produção:** O fato de conseguir reaproveitar um projeto antigo, agilizou bastante o processo de produção deste jogo. Toda a estrutura de movimento, variáveis, condicionais e reação entre os elementos do cenário foi trivial de ser construída. O que trouxe um pouco mais de dificuldade era o ajuste dos clones do objeto número. Dado que trabalhar com clones requer um cuidado adicional nas condições que os eliminam. Pois diferente do objeto original, o clone só existe durante a ativação do jogo, sua configuração e seu comportamento são um produto da configuração e comportamento que determinamos para o objeto original. Essa dificuldade de entender o funcionamento do clone, senão durante seu tempo de vida, torna esta uma ferramenta ao mesmo tempo que poderosa, perigosa para o próprio *software* do jogo.

Com a cautela necessária, aproveitando a estrutura do jogo **Abra a porta**, todo o processo de produção deste jogo iniciou na metade do mês de Abril e durou cerca de uma semana até atingir uma versão de protótipo jogável. Com apenas um nível, este era replicado infinitamente, variando de forma aleatória: os valores dos números espalhados na tela; a função que a máquina realiza; o valor inicial da máquina.

**Testes:** A fase de testes começou ao final do protótipo jogável. Em abril de 2016 eu e um colega da faculdade realizamos os primeiros testes do jogo com o intuito de testar sua funcionalidade. Mas, logo percebemos um aspecto crucial sobre sua jogabilidade. Resolver o problema da **Calculadora maluca** no contexto do jogo era seu objetivo inicial e isto parecia interessante. Mas um aspecto mecânico necessário para este objetivo, que é a movimentação dos números pela tela a partir de um personagem, isto era divertido. Nessa percepção, entendemos que

poderíamos encontrar outros problemas interessantes de serem resolvidos, mas deveríamos reforçar neles o uso desta mecânica que se mostrava divertida. Dado estes resultados, o projeto foi encerrado na fase de protótipo.



**Figura 13:** Análise do protótipo em quesito mecânica e objetivo (fonte própria).

**Finalização:** A versão foi encerrada no fim de Abril. Como pretendia manter a estrutura, era importante preservar o máximo desta versão no seu estado original. Salvando as *sprites* do personagem, e mantendo um registro de datas de cada versão desde o início ao término do projeto. Visto que no caso do **Abra a porta**, a disposição do projeto agilizou muito o desenvolvimento do **Pega e Larga**. Do mesmo modo que esperávamos que este bem organizado, agilizaria a produção do seu sucessor. Seus materiais foram arquivados para reuso e em uma análise dos aspectos positivos e negativos deste projeto, concluí que:

- a) Deveria investir mais na movimentação do personagem e dos números;
- b) O visual simplista do personagem deixava o jogo carismático;
- c) Faltava instruções para os jogadores saberem o que fazer;
- d) Os desafios se tornavam triviais muito rapidamente;
- e) O jogo tinha poucos atrativos visuais;
- f) A solução digitada não suportava respostas similares.

## Handles

Mantendo a estrutura da versão anterior (**Pega e Larga**), passei a investigar problemas resolvíveis com esta mecânica de arrastar números. Em outras palavras, como resolver um problema movendo os dados de um lado para o outro? Retomando a situação do **ExQuest**, onde os dados estavam a disposição do jogador para processá-los com o recurso de planilha eletrônica que



intuitiva fosse percorrer do mais próximo até o mais distante pegando-os ao mesmo tempo, o que faria com que ele estivesse com o máximo do peso na posição mais distante do seu destino. Outra estratégia para este problema seria ir primeiro ao objeto mais distante, e fazer o percurso de coleta contrário. Assim, ele só carregaria o peso máximo na posição mais próxima do seu destino.



**Figura 15:** Duas soluções para o problema de buscar os sacos pesados (fonte própria).

Um jeito de relacionar outros problemas nas dinâmicas de arrastar os números pela tela é aumentando as ferramentas do jogador. Um recurso natural aos sistemas de carregar objetos, seria a possibilidade de agarrar e depois soltá-los. Uma abordagem nova que esta ferramenta permite, é desobstruir trajetórias. Porém esta mecânica junto com a captação de vários objetos ao mesmo tempo gerava um novo problema sem solução até o momento, a formação de pilhas de objetos amontoados. Assim, a menos que o jogador consiga selecionar um objeto em específico, ele sempre que tentasse recolher algum objeto, agarraria todo o conjunto ao mesmo tempo. Então, para evitar isto poderíamos usar uma mecânica de selecionar o elemento do topo desta pilha de objetos. Juntando com a possibilidade de soltar objetos e coletar vários ao mesmo tempo, temos no jogo uma nova classe de problemas. Envolvendo a manipulação de objetos com pouco espaço, exigindo sua troca de posições para liberar espaço de trabalho.

Um jogo similar em dinâmica, é o **Torres de Hanói**, onde o jogador precisa passar as peças em uma determinada ordem de um pilar ao outro, usando um terceiro pilar como espaço auxiliar para esta movimentação. O grande problema deste jogo é o espaço para movimento das peças com um espaço excedente, isso exige do jogador um grande número de movimentos para reordenar as N

peças ( $2^N-1$  movimentos). De modo análogo, esta ferramenta no jogo nos possibilitaria colocar um problema com muitos objetos para serem organizados dentro de um espaço pequeno. Tomando cuidado com o espaço reduzido e o número grande de objetos, visto que estes são elementos que afetam diretamente no tempo para solução do problema.

Decorrente da própria condição do algoritmo que controlava se o personagem agarraria ou não algum número do jogo, percebi que seria possível acrescentar uma função Matemática associada a isto. Dado que todos os números possuem seus próprios valores associados, pois no **Pega e Larga** isto era acrescentado à máquina. Então não seria difícil verificar quando o personagem entrasse em contato com um número, se uma propriedade associada a este número era válida. Desse jeito, acrescentando sua validade como condição para agarrar o número, isto permitiria ao personagem passar dentro de um amontado de números e retirar dali somente aqueles que validam a propriedade determinada. Nesta perspectiva, se o jogo especificasse em dada situação uma propriedade e uma missão de levar números certos até a máquina, haveriam caminhos até antes obstruídos por números errados à missão, que poderiam ser atravessados sem que o personagem agarre nenhum. Isto mudaria a perspectiva do jogador na hora de resolver problemas de busca de soluções obstruídas e de organização dos números em espaços pequenos, dado que nem todos os números visíveis estariam de fato no conjunto daqueles que o personagem agarrará ao tocá-los.

Reorganizando as ideias, percebi que a última proposta tinha um potencial que se assemelhava ao jogo **The Incredible Machine**, original da plataforma DOS. Nele o jogador possuía algumas ferramentas e precisava cumprir missões como, por exemplo, fazer uma bola de basquete cair dentro da cesta. A forma como isto ocorria era da mais variada possível, desde ratos rodando engrenagens, rampas, bolas caindo, sopradores antigos para lareira, tesouras, bexigas e tantos outros recursos difíceis de associá-los de acordo com a função para a missão. O mais interessante deste jogo é o módulo de edição, onde o jogador possui também a liberdade para alterar elementos do próprio jogo, como a força gravitacional e a velocidade do vento. Assim, um sistema de desafio que possa não funcionar na gravidade da Terra, pode funcionar na gravidade Lunar, ou na gravidade de Saturno. Isto torna imenso o leque de caminhos para resolver um dado problema. Diferente de quando todas as instâncias são controladas somente pelo *software*, deliberar algumas responsabilidades aos jogadores afeta seu modo de agir.



**Figura 16:** Jogo **The Incredible Machine**.

Da mesma maneira ao **The Incredible Machine**, se aumentarmos a responsabilidade do nosso jogador, meios alternativos de solucionar os desafios Matemáticos poderão surgir. Uma propriedade facilmente alterável, é a própria seleção dos objetos na tela. Que já é regida pela verificação de uma propriedade Matemática. Assim, se o jogador puder controlá-la, teremos poder para resolver problemas até então insolúveis com uma propriedade de seleção estática. O que vai de encontro com a proposta desse jogo ter um potencial para desenvolver nos jogadores, habilidades do Pensamento Computacional.

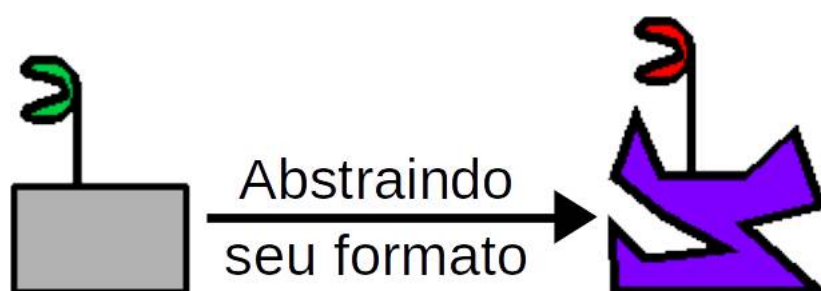
Dado o poder de seleção de números por uma propriedade controlável pelo jogador, tornou-se necessário analisar uma boa propriedade. Uma que permitisse a seleção dos mais variados conjuntos numéricos de forma única. Então percebi que a propriedade de “ $X \text{ mod } Y = 0$ ” teria uma função poderosa neste jogo. Esta propriedade verifica se o resto inteiro da divisão de  $X$  por  $Y$  é igual a 0. Controlando-a poderíamos quando quiséssemos, que um número fosse ou não agarrado pelo personagem, para isto determinando-a como verdadeira ou falsa a partir da escolha de  $X$  (número na tela) e  $Y$  (número escolhido pelo jogador).

Seu uso era ideal aos nossos interesses, dado que poderíamos soltar qualquer número verificando se um número  $X \text{ mod } 0 = 0$ . E pela impossibilidade de divisão por 0, este seria solto. Ou agarrar qualquer número, verificando se um número  $X \text{ mod } 1 = 0$ . E isto é verdade para todo  $X$  pertencente ao conjunto dos Inteiros. Outros grupos também poderiam ser formados a partir desta propriedade, como por exemplo, o grupo dos números pares, verificando quais números mod 2 são iguais 0. Ou o conjunto dos múltiplos de 3, verificando quais números mod 3 são iguais 0. De forma geral, todo número Natural maior que 0, pode ser escrito de maneira única a partir do produto de

números primos. Assim, em sentido oposto, todo número Natural maior que 0, pode ser determinado de maneira única a partir de seus divisores. No caso de selecionar somente os números 0, esta propriedade do mod ainda poderia ser utilizada, dado que o maior número na tela de jogo seja Z, então se o jogador escolher um  $Y > Z$ , para qualquer X na tela,  $X \bmod Y = 0$ , somente se X for igual a 0.

Apenas por uma questão de sonoridade com o título do jogo, procurei no idioma inglês um nome curto que representasse bem a dinâmica de manipular arrastando e soltando os problemas Matemáticos. Encontrei **Handles**, uma palavra cujo significado é bastante amplo: maçaneta; alça; manivela; manejar; manusear. Todos bastante próximos da ideia de resolver os problemas puxando os números como se eles tivessem alças, ou, mesmo que o problema possa ser resolvido com as mãos a partir da manipulação do personagem e de suas interações com os objetos.

**Pré-produção:** Em abril de 2016 comecei a construir o plano diretor deste jogo. Pude aproveitar muitos recursos do **Pega e Larga**, como os controles para movimentar o personagem, sua aparência, a interação dele com os números. Porém como a máquina agora desempenharia um papel de verificar um resultado e não mais de apresentar ao jogador uma informação, decidi remover o espaço da sua tela, destinado ao valor atual da operação realizada. Uma substituição imediata foi transformá-la em um quadrado sem o espaço para a informação. Porém, tinha uma preocupação latente sobre sua identificação pelo jogador. Pois na versão anterior do jogo, embora ela fosse simples, havia a tela com um número nela. Isto indica ao jogador que há algo a ser feito com este objeto. Mas o objeto por si só, parecia desinteressante. Analisando-o, como não precisava mais de um espaço para a informação, poderia torná-lo tão disforme quanto quisesse. Então, no próprio *software* do *Scratch 2.0*, comecei a puxar linhas daquele quadrado, sem nenhuma forma em mente, até que chegasse em um objeto sem semelhança a nenhum outro formato, algo tão estranho por assim dizer, que não lembrasse nada conhecido. Dessa forma, ao ser identificado, de imediato causaria uma certa curiosidade no jogador.



**Figura 17:** Máquina antes e depois da “abstração” de significados (fonte própria).



Com intenção de tornar o **Handles** mais próximo dos jogos comuns disponíveis na internet, imaginei necessário introduzir uma tela inicial com o nome do jogo e uma ideia de como este funciona. Assim, utilizei letras na fonte *Scratch* própria da linguagem *Scratch* e disponível no acervo da mesma, para escrever o nome do jogo em destaque. Abaixo, o personagem na sua forma mais atual (o motivo do cachecol vermelho será tratado mais adiante) carregando um número e movendo-se em direção de outros dois números. Com isto esperava passar a ideia principal deste jogo, que é movimentar números usando um boneco de palitinho.

Diferente do **Pega e Larga**, esta versão possui um final bem estabelecido. Para reforçar os incentivos ao jogador, também desenvolvi uma tela pós-jogo, como forma de parabenizar o jogador pela superação dos desafios. Procurei nela dar destaque ao personagem, como se o mesmo estivesse agradecendo ao jogador. E similar a disposição inicial onde haviam números e um personagem, na final coloco três imagens do personagem em movimento sequencial, numa tentativa de representar sua “liberdade” em relação a realização daquelas missões de transportar números.



**Figura 18:** Tela inicial e final do jogo (fonte própria).

Para auxiliar o jogador sobre o objetivo de cada missão e no uso das habilidades do jogo, imaginei uma tela de instruções ao início de cada missão. Pensei na tela preta com o número da missão atual e o total de missões, no caso de uma nova habilidade, instruções sobre como usá-la apareceria abaixo da missão, tudo em letras brancas e verdes. Após o início do nível, haveria uma mensagem em rosa choque, pedindo que passe naquela região para rever a missão. Após alguns segundos esta mensagem some e permanece apenas o sinal de interrogação que funcionaria do mesmo modo. Quando tocado pelo personagem, reinicia o nível apresentando ao jogador a tela de missão novamente. Com objetivo de suavizar o ambiente do problema, apliquei também uma coloração azulada em degradê partindo do centro da tela para os extremos. Assim, o cachecol

vermelho também surge como adereço ao personagem a partir do nível 5, como um sinal de que agora o jogador é capaz de selecionar os números a partir de seus divisores comuns.

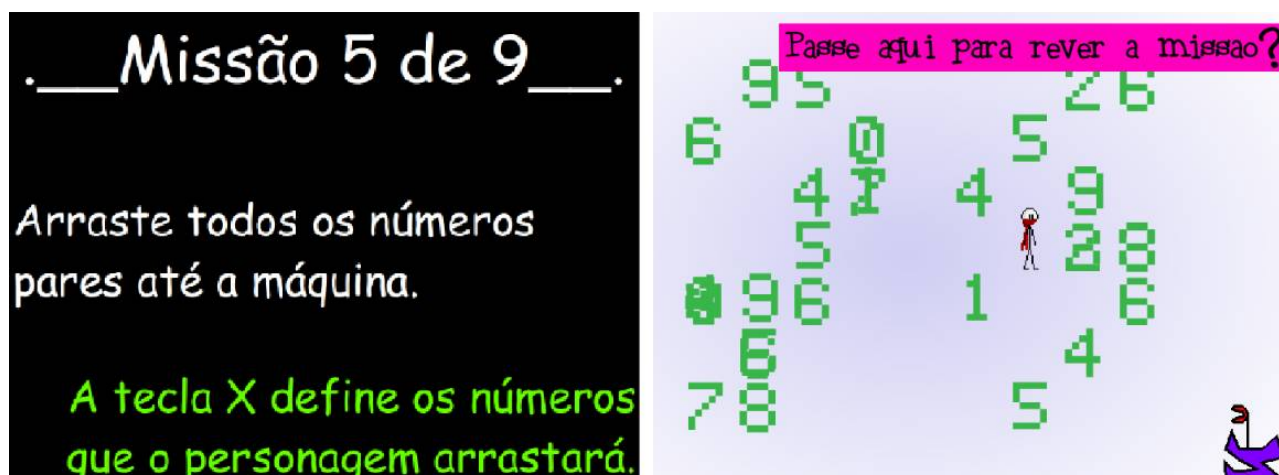


Figura 19: Tela de instruções e de jogo (fonte própria).

Em linhas gerais, descrevo estes e outros aspectos visuais considerados para o jogo:

- a) Idem ao anterior (*O jogo será bidimensional com visão de perfil*);
- b) Idem ao anterior (*O protagonista será um boneco de palitinho*);
- c) Idem ao anterior (*Os números serão pixelados e terão a altura do personagem*);
- d) A parte do jogo que recebe os objetos carregados pelo personagem (chamada de máquina) será da cor azul, com formato abstrato e terá uma antena em constante movimento;
- e) Informações sobre o nível atual, o total de níveis, a missão atual e instruções sobre habilidades, estarão dispostas telas pretas, cada qual referente a um nível do jogo;
- f) A tela inicial do jogo tem a imagem do protagonista e a mensagem “SPACE” piscando;
- g) A tela final do jogo tem a imagem do protagonista e uma mensagem de parabéns ao jogador;
- h) O protagonista possui dois visuais, um básico e o outro com um cachecol vermelho;
- i) O fundo de todos os níveis é levemente azulado do centro para as bordas;
- j) Cada vez que o jogador pressionar a tecla para alterar o *Divisor*, abre uma janela perguntando qual valor para *Divisor* o jogador quer, junto com um espaço para o jogador digitar uma resposta.

Lidar com as diferentes interações do personagem nos números foi um processo complexo. O que até então envolvia apenas agarrá-los, agora envolvia verificar qual condição estava ou não ativa na ocasião. Também era preciso controlar as condições de vitória e derrota imediata de cada nível (o que não existia no **Pega e Larga**), visto que um jogador pode falhar ao inserir um número errado entre vários certos na máquina. Foi preciso construir uma lista ativa a cada nível que verifica dentre os números que correspondem a resposta certa, quantos ainda faltam entrar na máquina.

Assim, como para cada número errado ativo naquele nível, um gatilho que interrompe o nível inteiro e declara fracasso na missão, caso seja inserido na máquina. Estes e outros aspectos lógicos que considere na elaboração deste jogo estão listados a seguir:

- a)** Idem ao anterior (*O jogador controla o personagem através dos respectivos direcionais do teclado*);
- b)** Todos os níveis terão números entre 0 e 9 escolhidos aleatoriamente ou predeterminados, com posicionamentos aleatórios ou predefinidos na tela, uma máquina no canto inferior direito, o personagem do jogador localizado no centro e uma mensagem de ajuda no canto superior direito;
- c)** O primeiro número que encostar no personagem ficará com ele, até o número tocar na máquina;
- d)** Quando o personagem carrega um número, outros números que encostam nele não ficam presos;
- e)** Cada vez que um número toca a máquina ele some da tela e é considerado inserido na máquina;
- f)** Se todos os números corretos forem inserido na máquina, o jogador avança ao próximo nível, se ao menos um número errado for inserido na máquina, o nível atual se reinicia imediatamente;
- g)** A tela de informações aparecerá por 8 segundos antes do início de cada nível;
- h)** A mensagem de ajuda surge ao início ou reinício de cada nível na forma de uma frase, mas após 5 segundos ela se encurta para um único caractere “?”;
- i)** Se o personagem tocar na mensagem de ajuda, o nível será reiniciado automaticamente e a tela de ajuda do nível atual será mostrada ao jogador até que este pressione qualquer tecla;
- j)** O jogador somente avança para o primeiro nível do jogo, quando aperta “espaço” na tela inicial;
- k)** A partir do nível 5, o visual do personagem se transforma do básico para o cachecol vermelho;
- l)** A partir do nível 2, o jogador adquire um comando para o personagem soltar números;
- m)** A partir do nível 3, o personagem consegue agarrar quantos números quiser ao mesmo tempo;
- n)** A partir do nível 4, o jogador adquire um comando para o personagem pegar apenas o primeiro número que encontrar, dando preferência aqueles que estão em camadas superiores;
- o)** O personagem recebe uma penalidade de movimento diretamente proporcional com a quantidade de números que carrega.
- p)** A partir do nível 5, o jogador adquire um comando para alterar a condição que faz o personagem agarrar números, isto ocorre a partir da abertura de uma janela onde o jogador digita o que quiser, que ocupará o espaço de uma variável numérica chamada de *Divisor*.
- q)** Sempre que o personagem toca em um número, este número verifica se seu valor é divisível pelo valor do *Divisor*, caso positivo ele ficará preso no personagem até que seja inserido na máquina ou o jogador utilize o comando de soltar números, senão o número permanece no seu lugar;
- r)** O jogo possui 9 níveis.

**Produção:** Em maio de 2016 a construção do jogo começou com o reaproveitamento da mesma estrutura de nível do **Pega e Larga**, acrescentando as novas configurações nesta estrutura. Porém este processo não ocorreu de forma bem planejada, então a medida que percebia um erro no jogo, logo abria o código e mexia em algumas componentes até consertá-lo. Mas esta falta de zelo pelo que era modificado, gerava uma estrutura bastante sensível a modificações. Neste período, ainda não compreendia a necessidade de investir na qualidade, otimização e modularização do código. Achava erroneamente que se o jogo funcionasse minimamente, eu poderia em pouco tempo corrigir os defeitos e deixá-lo otimizado. Outro conflito foi a impossibilidade de criar planos de fundo com textos no próprio *Scratch*, que utilizassem acentos ou ç. Para isto precisaria desenhá-los ou trabalhar com imagens de fora do *Scratch*. Nesta situação, optei por escrever as telas de auxílio sem acentuação ou ç. A seguir explico de forma sucinta a configuração de cada nível.

- 1) 3 números aleatórios espalhados pela tela. A missão do jogador é levar todos (um a um) para a máquina.
- 2) 3 números aleatórios maiores do que 0 espalhados pela tela e uma linha de números 0 bloqueando o acesso até a máquina. Neste nível o jogador ganha a habilidade de soltar números. A missão do jogador é levar todos os números maiores do que 0 para a máquina.
- 3) 9 números aleatórios espalhados pela tela. Neste nível o jogador ganha a habilidade de agarrar vários números ao mesmo tempo. A missão é levar todos até a máquina.
- 4) O número 1 e o número 7 sobrepostos ao centro da tela. Neste nível o jogador ganha a habilidade de retirar apenas o número de cima de uma pilha. A missão é levar somente o 7 até a máquina.
- 5) 10 números múltiplos de 3 e 10 números não-múltiplos de 3 espalhados pela tela. Neste nível o jogador ganha a habilidade de alterar a variável *Divisor*. A missão é levar somente os múltiplos de 3 até a máquina.
- 6) 10 números pares e 10 números ímpares espalhados pela tela. A missão é levar somente os números pares até a máquina.
- 7) 10 números pares e 10 números ímpares espalhados pela tela. A missão é levar somente os números ímpares até a máquina.
- 8) 10 números zeros e 10 números maiores que 0 espalhados pela tela. A missão é levar somente os números zeros até a máquina.
- 9) 10 números zeros e 10 números maiores que 0 espalhados pela tela. A missão é levar somente os números maiores que 0 até a máquina.

**Testes:** De maio a junho de 2016 realizei junto a um colega os mesmos testes do **Pega e Larga** para encontrar falhas e avaliar a interação com os elementos do jogo. Mas no decorrer dos

próprios testes, confundia quando o personagem estava ou não com a habilidade de mudar o critério do *Divisor*. Isto me levava várias vezes a considerar uma falha, tentar corrigi-la e depois perceber que a falha era em outro lugar e tentar desfazer minha última correção. Para evitar criar mais problemas na hora de corrigir erros, decidi criar algo que diferenciasse visualmente o personagem quando esta habilidade estivesse ativa. Assim, inseri um cachecol vermelho no personagem. A escolha pelo cachecol foi um gosto pessoal por este elemento em diversos contextos, desde o Pequeno Príncipe a outros personagens em Animes (por exemplo a Mikasa Ackerman, de Shingeki no Kyojin) e jogos digitais (**Strider Hiryu** da plataforma Super Nintendo) que utilizam-no também. Este adereço embora não intuitivo com sua função, tinha como objetivo chamar minha atenção sobre a habilidade, mas percebi agradar também a várias pessoas, assim, foi mantido em definitivo.

Como forma de testar a interface com o usuário, reuni 5 amigos em ocasiões diferentes com perfis peculiares para jogarem e depois responderem a uma entrevista estruturada. Esta entrevista perguntava diversos aspectos de como o jogador percebia o funcionamento dos comandos no jogo, ou como era sua jogabilidade e principais dificuldades. As questões eram respectivamente:

- a) Qual sua idade?
- b) Você já teve videogame ou contato contínuo com um? Se sim, com que idade? Como era sua relação com videogames ou jogos virtuais? Atualmente você joga, com que frequência?
- c) Até que nível você chegou?
- d) Qual missão foi mais difícil? Por quê?
- e) O jogo apresentava instruções sobre os comandos?
- f) Qual era a função da tecla Z?
- g) Qual era a função da tecla X?
- h) Você utilizou o botão “?” na tela para rever o objetivo? Ele foi útil?
- i) Como era o personagem?
- j) Alguma observação adicional?

Os testadores tinham experiências muito diferentes, desde um rapaz que nunca teve um videogame, a uma jovem cujo pai é ligado a uma plataforma de videogames e ela recebia novos títulos para testar em primeira mão. Os resultados destes testes sugeriram que vários elementos do jogo não foram compreendidos pelos jogadores, independente das suas experiências. A dificuldade em interpretá-los dificultava o uso destas ferramentas ou atrasava o processo de *insight* necessário para solucionar os problemas Matemáticos envolvidos. Como por exemplo perceber que o conjunto dos números ímpares é o complementar do conjunto dos números pares, que são facilmente selecionáveis a partir do divisor comum 2. Ou que para selecionar os números 0 da tela, é

necessário a escolha de um divisor comum superior ao maior número na tela. Sequer a função de consulta da tela de instruções foi bem compreendida. Os jogadores tentavam apertar “?” no teclado, em vez de passar por esta parte da tela com o personagem. Nesta análise, todos os jogadores tinham imensa dificuldade em utilizar o critério dos divisores comuns para seleccionar os números certos e cumprir a missão.

Em outras ocasiões, apresentando o jogo para colegas de trabalho que estudam o mesmo tema. Estes concordam com as potencialidades do **Handles** no desenvolvimento das habilidades do Pensamento Computacional. Sugerindo entretanto que em vez do aprimoramento do jogo, fosse investido numa triangularização de como as habilidades possíveis de se desenvolver aparecem mediante os desafios e oportunidades que o jogo oferece.

**Finalização:** Parei de investir neste jogo em junho de 2016. As principais razões para isto foram as dificuldades em inserir novos elementos e desafios na estrutura atual do jogo. Era inviável construir mais desafios ou melhorar a jogabilidade na forma que o código havia sido escrito. O código por si só possuía muitos erros e muitas partes sobressalentes. Eram exceções em cima de exceções, construídos somente para funcionar minimamente. Mesmo quando pequenas as modificações, elas traziam muitas complicações no processo de corrigir o código.

Assim, percebemos que os problemas de Matemática neste jogo possuíam um grande potencial para nossa investigação, porém era mais fácil reconstruir o código, a corrigi-lo. E no ponto de vista prático, dado que independente das experiências dos testadores, estes apresentavam dificuldades em pontos considerados por mim e pelos meus colegas, como básicos para que o Pensamento Computacional fosse desenvolvido. Estes aspectos precisavam ser melhorados para que o jogo ensinasse ao jogador, de forma eficaz, como jogá-lo. Analisando este projeto considerei os seguintes aspectos positivos e negativos resultantes da experiência:

- a) A abrangência de problemas Matemáticos que podem ser tratados a partir da mecânica de controlar o critério do *Divisor*;
- b) O cachecol vermelho e o formato abstrato da máquina agradava de imediato todos que jogaram;
- c) Distribuir tarefas intelectuais entre o jogador (que toma as decisões) e o computador (que calcula os números divisíveis pelo *Divisor*) era muito positivo aos jogadores que aprendiam a jogar;
- d) O excesso de habilidades e sua redundância deixava o jogador confuso sobre como jogar;
- e) Por vezes o jogador escolhia uma estratégia funcional porém muito mais trabalhosa;
- f) As instruções eram muito genéricas sobre o uso das habilidades, confundindo o jogador;
- g) Cada habilidade é explorada em apenas um nível, dificultando que o jogador aprendesse a usá-la;

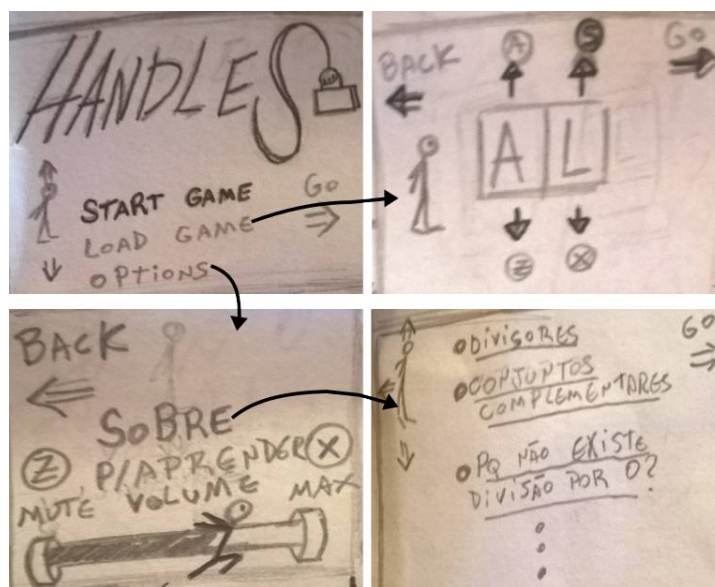
**h)** A entrada textual para o valor do *Divisor* substituía a variável pelo que fosse digitado, se uma palavra fosse digitada, teria o mesmo efeito que mudar o *Divisor* para 0, dado que  $X \bmod [\text{palavra}]$  é impossível, implicando numa afirmação falsa, e soltando qualquer número.

## Handles 2.0

Este foi um divisor de águas entre meu modo atual de construir jogos em *Scratch 2.0* e a maneira quase certa de se fazer isto. O código estava muito bagunçado, sua execução era sujeita a algumas falhas não identificáveis. Havia partes obsoletas que poderiam ou não afetar o jogo. Ele foi construído a partir de uma série de condicionais que o tornavam funcional, porém dificilmente editável. Então me coloquei na seguinte análise reflexiva: valeria a pena investir no seu formato atual, trabalhando para cruzar seus tópicos com as potenciais habilidades do Pensamento Computacional que ele desenvolveria? Ou talvez devesse repensar em toda sua estrutura, manter a ideia principal e fazer “certo” dessa vez? Neste período a universidade iniciou uma greve sem data de término, o que foi o incentivo necessário em tempo livre para abraçar a ideia de recriar o **Handles**, sem perder sua essência, mas utilizando uma abordagem bem mais profissional (se assim posso chamar meu esforço na sua construção). Apenas para distinguir durante o registro interno das versões, classifiquei o novo projeto como **Handles 2.0**.

A dificuldade em organizar os requisitos para o novo projeto eram imensas. Não cabia em uma única página de tamanho A4 os esquemas ou rascunhos que passavam por minha cabeça sobre como o jogo deveria ser e quais eram as consequências de cada decisão. Por vezes me atrapalhava, perdia uma folha ou não conseguia analisar o projeto inteiro de uma só vez. Ele estava como um imenso emaranhado de ideias que por mais esforço, seus nós não se desatavam. Então comecei a trabalhar em cartolinas, construí os esquemas ilustrados de cada nível do jogo atual, representações das habilidades pretendidas ao personagem, com as consequências e requisitos delas. O trabalho com uma cartolina começou a se soltar, tudo para compreender melhor a dificuldade envolvida para entender meu próprio projeto e conseguir identificar o que precisava ser feito sem usar os caminhos mais tortos até o objetivo, estes caminhos tortos eu diria serem os condicionais e exceções que usei no projeto anterior para deixá-lo somente funcional. Por fim, cada aspecto lógico das estruturas, listas, variáveis, blocos de comandos, tudo o mais que era necessário para construir este jogo se desenrolou em três cartolinas frente/verso. Assim, esta seção do diário discute um pouco sobre estas reflexões e como elas simplificaram toda a estrutura deste jogo.

**Pré-produção:** O projeto começou em junho de 2016 de forma ambiciosa, desejava colocar uma tela inicial com opções de seleção ao jogador. Que este pudesse escolher entre começar de onde parou, ou ajustar aspectos como volume da música e ver dicas e conceitos da Matemática envolvidos nos desafios. Queria o jogador selecionando estes elementos a partir dos direcionais e o personagem principal acompanhando-o na decisão (similar a tela inicial do jogo **Megaman X**). Até que uma escolha fosse feita e o personagem passasse correndo levando a opção para fora da tela e abrindo a tela escolhida (também baseado em **Megaman X**, que quando o jogador escolhe uma opção da tela inicial, o protagonista dispara um tiro de plasma na opção escolhida). Pensei nisso para que desde antes do jogo começar, o jogador entendesse o personagem como alguém que arrasta os objetos pela tela. Também queria inserir um sistema de *passwords* de duas letras para o jogador recomeçar do nível que parou, visto que um jogo longo dificilmente é jogado de uma só vez. Também planejava ter uma música própria como muitos jogos, e assim liberar ao jogador o poder de alterar seu volume próprio dentro do *software*. Também dispor uma opção para que o jogador acesse mais detalhes sobre os aspectos da Matemática tratados no jogo, como o funcionamento da divisão inteira, o que são conjuntos complementares, a inexistência da divisão por 0, e outros tantos que viabilizam a resolução objetiva dos desafios.

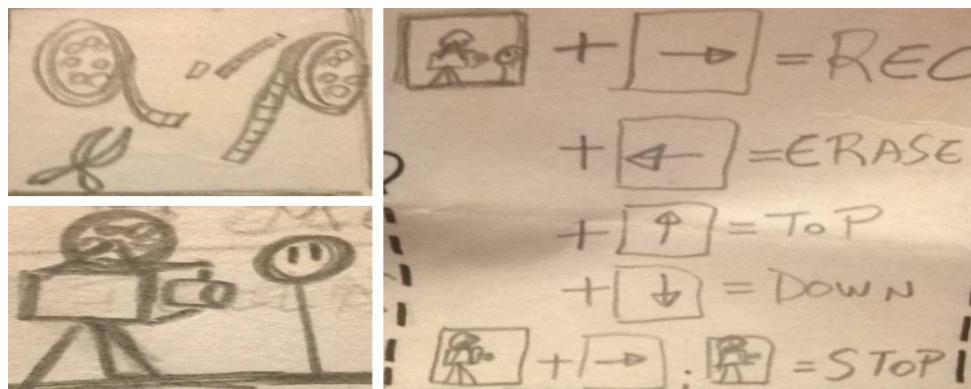


**Figura 20:** Rascunhos da dinâmica na tela inicial do jogo (fonte própria).

Na seção inicial deste diário, descrevi um quadro de jogo onde o personagem receberia ordens para buscar tijolos e construir muros (representado na figura 4). Naquela época isto parecia uma tarefa muito complexa, porém agora com um personagem que se movimenta, agarra e solta objetos, decidi reavaliar seus requisitos. Uma forma simples que pensei para o jogador dar estas ordens era a partir da “gravação de tomadas de jogo”. A lógica seria a seguinte, o jogador faz uma



série de movimentos em um espaço isolado da tela, e depois o reproduz na posição desejada para a ação ter o efeito desejado. Similar a acrescentar na edição de um vídeo, o trecho de outro vídeo repetidas vezes consecutivas, mesclando as ações de ambos para quem o assiste.



**Figura 21:** Esquemas de gravar o personagem e acrescentar nas ações de jogo (fonte própria).

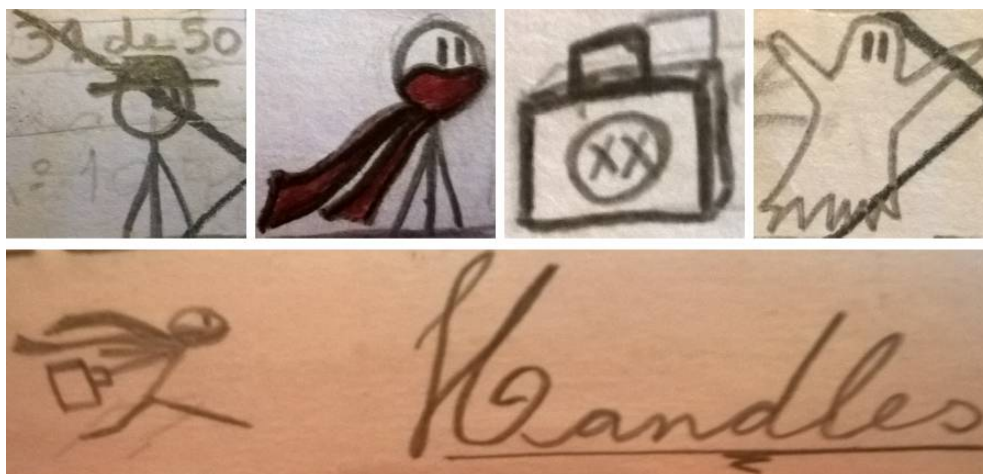
Na ideia de buscar os tijolos, este processo poderia ser implementado na ação de ir até a pilha trazer um tijolo novo para colocar no muro, vezes o número de tijolos necessários. Então, para construir esta mecânica, seria essencialmente necessário um comando de ativação da gravação, um outro para interrompê-la, e um terceiro para executá-la. Comandos adicionais poderiam ter a função de selecionar o número de vezes que ela será repetida, e até mesmo se não for a intenção gravar a nova por cima da antiga, poderia ter um comando voltado para escolher uma das gravações para executar, assim como para apagá-la. Em termos práticos, o jogo armazenaria as ações do personagem e as replicaria como se fosse o computador jogando com o personagem. A escolha desta abordagem se dava principalmente pela sua simplicidade, dado que na visão do jogador ele não precisaria criar um código ou desenvolver um algoritmo em uma linguagem abstrata, somente realizar as próprias ações de jogo e ter seus movimentos gravados como se tivesse esta “câmera especial” que permite reproduzir as ações gravadas a partir de outras posições iniciais.

Outra questão difícil de lidar foram as habilidades necessárias para o jogador. Pois dado uma habilidade, espera-se um uso. Seria estranho dar uma *jetpack* a um personagem que consegue acessar todos os locais do jogo apenas andando. Cheguei a pensar em algumas habilidades triviais como ferramentas para resolver um problema de Matemática. Como o “chapéu do contador” que permite o jogador verificar dado uma propriedade X, quantos destes elementos ele já pegou, assim por exemplo, o jogador precisa pegar todos os números da tela que sejam primos e soma dos algarismos seja maior que 7. Regulando o “chapéu do contador” para esta propriedade, de imediato o jogador saberá quantos elementos com este requisito ele está carregando e qual o total naquele nível. Outra habilidade que já existia e permanece desde a versão anterior até a última delas, é o

efeito “cachecol vermelho”. Uma habilidade para selecionar os números na tela a partir da escolha de um divisor comum a eles. Assim por exemplo, para agarrar somente os números 4 de um conjunto amontoado de números entre 1 e 10, o jogador pode selecionar os divisíveis por 8, removê-los do grupo, e então selecionar os divisíveis por 4. Outra habilidade trabalhada nesta ocasião inicial é a “maleta”. Uma liberdade para o jogador controlar qual a capacidade máxima de objetos que ele carregará. Assim, em um contexto onde ele queira selecionar exatamente metade do conjunto, podemos agarrar todos, descobrir o tamanho do conjunto, então reduzir o espaço livre da “maleta” para metade do total de elementos, logo o personagem só carregará a quantidade desejada, ou seja, metade do total de objetos da tela. Preocupado com a necessidade de ignorar os objetos, quando talvez estes não fossem números (que poderiam ser ignorados atribuindo *Divisor* igual a 0), pensei em uma habilidade de “Fantasma”. Ela permitiria o personagem transpassar qualquer obstáculo e ignorar qualquer objeto.

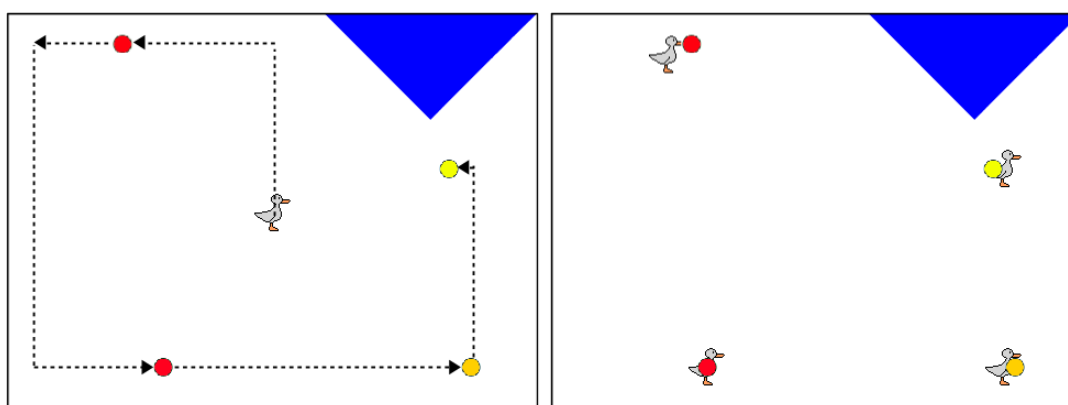
Contudo, a habilidade “chapéu de contador” seria muito difícil de ser implementada, dado que eu precisaria de um registro de todos os objetos da tela e de uma mecânica onde o jogador possa entrar com a propriedade a ser verificada. Não poderia simplesmente cobrar uma entrada textual em linguagem comum a programação Matemática como Octave ou Matlab. Mesmo permitindo-a, precisaria de um mecanismo de tradução da entrada para a lista do jogo. Outro motivo para abdicar desta habilidade é seu uso ter mais papel analítico do que ativo, o jogador saberia algo com esta habilidade, contudo a mesma não seleciona estes objetos. Uma habilidade assim poderia ser usada para selecioná-los, contudo isto ampliaria muito o leque de alternativas e requisitos necessários para jogar este jogo. Seria equivalente a um problema real onde dentro de um banco de dados o jogador precisasse determinar uma função seletora que trouxesse todos os objetos que respeitam uma dada propriedade. Mas tendo a liberdade para construir a função seletora e conhecendo a condição a ser selecionada, o único trabalho do jogador seria transcrever a condição para o espaço da função desta habilidade.

Também a habilidade “Fantasma” pareceu obsoleta, não só para números, mas também para qualquer outro objeto que se agarra no jogo. Visto que o jogador pode escolher a capacidade da habilidade “maleta” para 0. Ou seja, não tem espaço para carregar nenhum objeto e assim solta todos e também não os agarra. Dessa forma, determinei como principais habilidades para interação com os desafios principais do jogo seriam o “cachecol vermelho” e a “maleta”. Restando agora construir problemas que fizessem bom uso destas ações na sua resolução.



**Figura 22:** Esquema das habilidades do jogo (fonte própria).

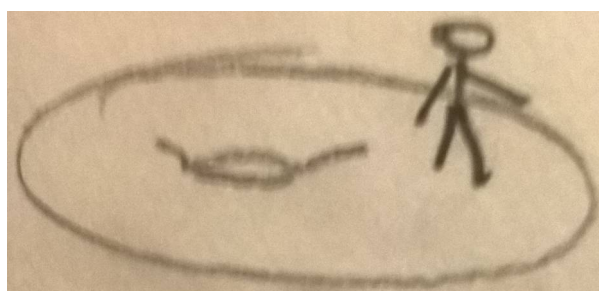
Analisando os requisitos necessários para gravar uma ação do personagem e reproduzi-la, lembrei-me de uma experiência anterior. Em um projeto de jogo digital chamado **Volta no Tempo**, o jogador precisava acionar vários botões ao mesmo tempo para abrir uma porta, mas ele não possui objetos arrastáveis para colocar sobre o botão. Contudo o personagem pode voltar no tempo. Então, o jogador pode ir até um botão, voltar no tempo, ir para o segundo botão enquanto sua “duplicata” temporal esta indo para o primeiro botão. E repetir isto até que todos os botões sejam pressionados.



**Figura 23:** Esquema básico do jogo **Volta no Tempo** (fonte própria).

O foco deste projeto era fazer os jogadores pensarem além das dimensões usuais, ou seja, determinar quanto tempo seria necessário esperar em um lugar até que quando voltasse no tempo, sua “duplicata temporal” conseguisse chegar até o botão. Uma dificuldade que tive nesta época era com a fronteira da tela, pois ao registrar os movimentos do personagem, se ele estivesse movendo-se contra uma parede, o jogo ficaria registrando um movimento inexistente. Ou mesmo se colocasse um personagem para realizar uma ação gravada que o fizesse andar contra uma fronteira, este ficaria se movendo contra ela acreditando estar se movendo, mas, na verdade, estaria parado. Uma

solução para este problema seria costurar a tela do jogo por seus extremos. Ou seja, se o personagem mover-se contra a lateral direita da tela, aparecerá na lateral esquerda da tela. O mesmo para o topo e a base da tela. Esta ideia tornaria a movimentação ilimitada, resolvendo o problema do personagem tentar mover-se e não conseguir, ou registrar um movimento inexistente na prática. O efeito prático desta “costura” é que o jogo agora ocorreria na superfície da figura geométrica tridimensional de um toro (uma rosquinha, ou uma câmara de pneu), embora seu gráfico visível ao jogador continua bidimensional, na prática esta estrutura permite raciocínios e estratégias possíveis no contexto tridimensional. Uma analogia que me agrada para explicar esta movimentação é como se fosse uma formiga caminhando sobre uma rosquinha voadora.

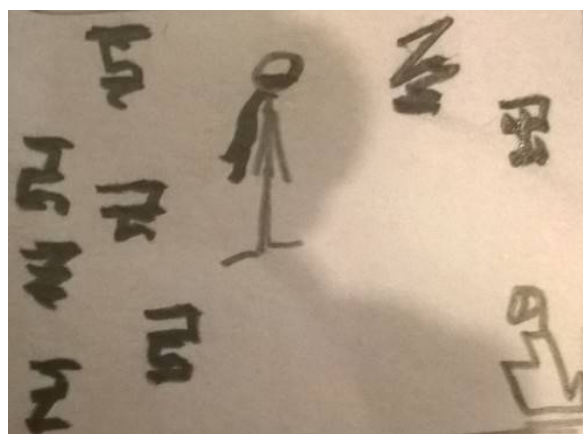


**Figura 24:** Situação esquemática do personagem sobre a superfície de um toro (fonte própria).

Dentro destas condições, planejei a tela de jogo com os comandos úteis do jogador visíveis, os valores de suas variáveis como *Divisor* e a capacidade da *Maleta*. Informações como o nome da fase, o número do nível atual e seu *password* para recomeçar neste nível. Após reflexões achei melhor remover o fator de *passwords*, em vez disto, liberar todos os níveis aos jogadores. Eles poderiam acessar qualquer nível do jogo a hora que quisessem, o que simplificaria muito a dinâmica do jogo, dado que os níveis não possuem memória.

Procurando acrescentar novos problemas resolvíveis dentro das atuais habilidades do personagem, percebi que o fator visual excluía as abordagens mais abstratas. O jogador não faria uma varredura em um conjunto que ele consegue perceber visualmente não ter determinado objeto do seu interesse. Posso exemplificar esta situação dentro de um problema simples, levar somente os números 1 até a máquina, tendo na tela números de 1 até 29. Se o jogador consegue visualizá-los, localiza os números 1, e caso tenha algum obstáculo, faz uma desobstrução que lhe permite agarrar somente os números 1. Porém, se o jogador não puder distingui-los visualmente, precisaria passar um filtro para selecionar os números 1 a partir do seu divisor comum. Contudo, os únicos divisores de 1 é ele mesmo ou seu oposto aditivo (-1), mas estes também são divisores de todos os outros números. A estratégia aqui seria separar tudo aquilo que é diferente de 1. Ou seja, os que possuem divisores comuns 2, 3, 5, 7, 11, 13, 17, 19, 23, 29. O que restou deverá ser o 1, dado que qualquer

número Natural maior que 1 e menor que 30, será divisível por, pelo menos, um número primo entre 2 e 29. Outra estratégia mais mecânica para este problema, seria realizar uma seleção iterativa partindo de todos os divisores maiores que 1. Ou seja, gravando o personagem se movendo do amontoado de números com o *Divisor* igual a 2, então mudando *Divisor* +1 e retornando ao monte. Ao final de 28 iterações deste algoritmo, o personagem terá retirado do amontoado de números, todos aqueles que são divisíveis por algum número entre 2 e 29. Permanecendo no monte somente os números 1. Uma mescla de ambas as abordagens neste caso (retirar somente os números primos e automatizar o processo) poderia ser retirando inicialmente os divisíveis por 2, então mudando o divisor comum para 3, automatizar o processo mudando a cada iteração o *Divisor* para *Divisor*+2, o efeito desta ação é remover todos os ímpares dado que todos os pares foram inicialmente retirados. Isto reduz as 28 iterações do modo automático para 14 iterações.

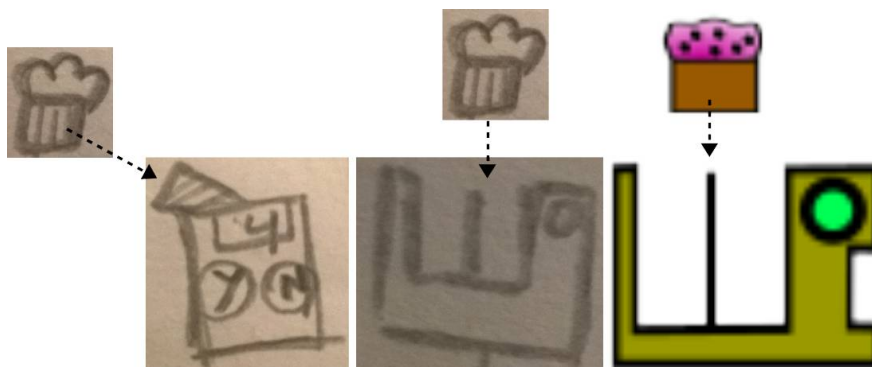


**Figura 25:** Rascunho do esquema de números indistinguíveis visualmente (fonte própria).

Uma outra classe de problemas que poderiam ser incorporadas ao fator de números indistinguíveis visualmente, é a entrega ordenada. Ou seja, estipular determinada sequência numérica, que deve ser inserida do maior para o menor, por exemplo, todos os pares. Assim, se na tela tem números entre 1 e 29, o jogador precisaria reuni-los e partindo do maior *Divisor* (28), levá-los até a máquina e retornar ao monte, então mudar para *Divisor* -2, repetindo este processo por 13 iterações, levaríamos até a máquina todos os números pares na ordem do maior para o menor.

Agora pensando em problemas que envolvam quantidades, isto parecia um tanto difícil de se tratar somente com números. Poderia atribuir-lhes problemas como “leve 4 números 8 até a máquina”. Porém é um problema tão simples que não vem a ser realmente um problema (o jogador selecionaria os números 8, depois mudaria a capacidade da maleta até ficar com só 4 deles). Buscando algum problema realmente interessante de se pensar numa solução, comecei a lembrar de algoritmos como a busca por bisseção. Dado um conjunto qualquer de elementos indistinguíveis, se existe apenas um elemento com uma propriedade diferente e verificável a partir de um teste

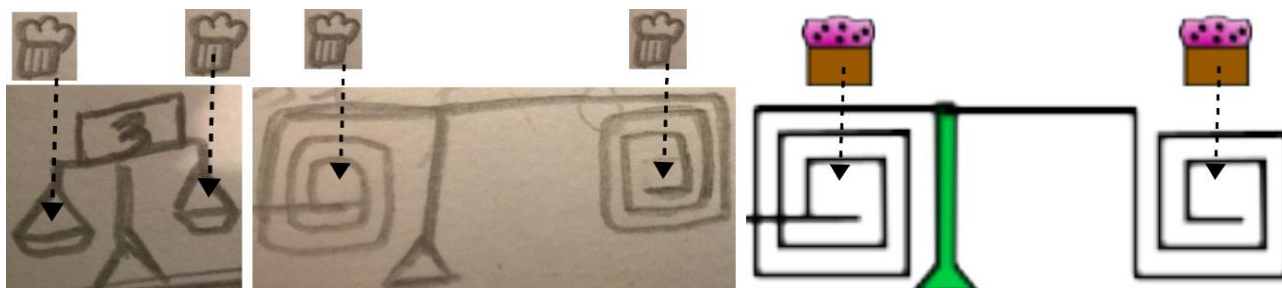
aplicado a um conjunto de qualquer tamanho, posso particionar meu conjunto de busca pela metade a cada novo teste, reduzindo-o até que ele tenha tamanho unitário e saiba, com certeza, qual elemento possui a propriedade verificável. De forma mais simples, pense nos objetos como bolinhos. Um dado bolinho está contaminado, o jogador consegue levá-los até um escâner que verifica se “dentro todos os bolinhos colocados no escâner”, existe pelo menos um contaminado. Embora este tipo de problema possa ser resolvido por este método, também permite uma abordagem mecânica. Levando um bolinho até o escâner, verificando se ele é contaminado, se for inserindo-o na máquina, e se não for, isolando-o na tela e retornando ao amontoado de bolinhos. Repetindo este processo pelo número de bolinhos na tela, teremos localizado aquele contaminado. Por uma questão de facilitar a inserção destes objetos no escâner, imaginei-o com um espaço como se fosse um micro-ondas aberto, e apenas para evitar que o jogador o coloque exatamente no vão da imagem e ela não o identifique em contato, ele poderia ter um pilar preto central.



**Figura 26:** Evolução dos esquemas conceituais do escâner no jogo (fonte própria).

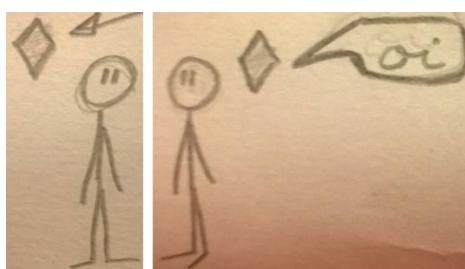
De modo análogo ao problema do escâner, poderia usar uma balança. Atribuindo a todos os bolinhos um peso igual, com exceção de um único mais pesado. O jogador poderia dividir o conjunto em três grupos, comparar dois deles, e determinar em qual lado está o bolinho mais pesado. Caso estejam em equilíbrio, significa que os dois comparados podem ser descartados e no terceiro grupo tem o bolinho mais pesado. Uma outra abordagem pode ser mecânica, identificando dois bolinhos com pesos iguais, podemos usar um deles como medida padrão, e então comparar um a um todos os outros bolinhos com aquele de medida padrão. Em questão da aparência da balança, precisava que ela emitisse um resultado identificável de forma espacial. No caso de uma balança usual de pratos, a diferença de altura de um prato para o outro seria muito pequena e dificultaria muito a tarefa de automatizar o processo, visto que a seleção automática dependeria muito da posição resultante da comparação. Então se o movimento do prato fosse lateral e move-se uma distância fixa entre uma posição e outra, o jogador conseguiria automatizar o processo de busca, percorrendo em cada ocasião uma delas na hora de recolher os objetos em uma dada posição. Porém

pelo mesmo problema do escâner, havia o risco do espaço para incluir o bolinho ser vazio em relação ao seu contato com a balança, fazendo com que a verificação não ocorra. Uma alternativa para este problema, era fazer da balança um espaço preenchido. Mas por questões estéticas, achei que ela ficaria mais elegante sendo uma região quadrada com metade de seu interior preenchido por linhas retas formando uma espécie de espiral.



**Figura 27:** Evolução dos esquemas conceituais da balança no jogo (fonte própria).

Durante a experimentação do efeito tela “costurada”. Comecei a identificar falhas geradas pela variação da espessura do personagem. Isto afetava sua posição e conseqüentemente fazia com que o efeito de ir parar no outro lado da tela, sofresse variações. Uma forma paliativa de tratar isto, foi acrescentando um minúsculo ponto branco acompanhando-o. Assim, sua largura em termos de código ficaria maior e seria sempre detectada quando cruzasse ou não o limite do identificador que o faria passar para o outro lado da tela. Porém, aquele pequeno ponto (que ninguém perceberia), ficava me incomodando. Eu sabia que ele estava lá, e conseguia vê-lo na hora de testar as versões. Pensei em tratar este ponto de forma mais chamativa, deixando-o com o aspecto de um elegante cristal flutuante que acompanha o personagem. Similar ao cristal azul que acompanha a personagem Rimururu do jogo **Samurai Shodown** (com a qual adorava jogar quando criança). E dado sua presença de palco (sendo um objeto bastante chamativo), comecei a imaginar alguma interação com o usuário a partir deste cristal azul como um personagem assistente de jogo.

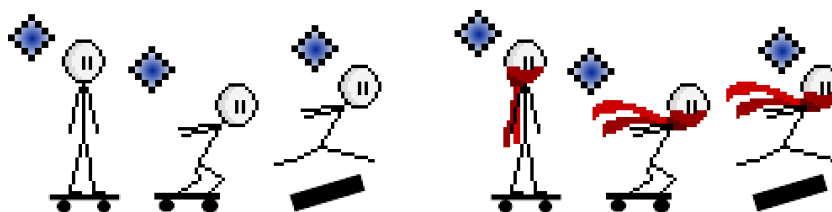


**Figura 28:** Conceitos iniciais do personagem cristal azul (fonte própria).

A forma como este projeto cresceu foi assustadora. Precisava jogar aos montes boas ideias para conseguir dar conta de construí-lo. Cheguei a abdicar das mecânicas dos objetos balança,

escâner, bolinho, e da ideia de “maleta” (embora na versão seguinte, **Handles in Scratch 2.0**, tenha resgatado estes conceitos aqui explicados). Um conceito que estava analisando durante a produção, e que afetou diretamente todo o produto é em questão a movimentação do personagem na vertical, pois enquanto o movimento horizontal era fácil, bastando mover as perninhas, o vertical não. Não dava para simplesmente dizer que ele pulou, ou ele se agachou, exceto que ele ficasse voando pela tela. Uma solução para o problema era substituir o personagem pelo ícone do cursor do mouse de “mãozinha”, na minha visão do jogo, era indiferente se o personagem agarrava um número, ou uma mãozinha o fizesse, até faria mais sentido que quem agarra e arrasta os números seja uma mão do que um boneco de palitinho. Contudo, meu assistente de projeto que testava desde a versão inicial, afirmou que não mais me auxiliaria se eu substituísse o personagem por uma mãozinha. Dizia ele que um boneco de palitinho arrastando números era uma das coisas mais legais deste jogo, e que isto devia ser preservado. Pensei muito, e a ideia que tive foi baseada na experiência que uma colega do grupo de desenvolvimento de jogos virtuais me contou.

Uma graduanda em Ciências da Computação especializada em animação para jogos 2D, um dia contou sobre uma estratégia de animar o movimento dos personagens de forma genial. Era fazendo aqueles “risquinhos” de quando alguém move as pernas rápido demais (similar ao desenho do Papa-léguas). Então, com apenas uma cena, daria ao jogador toda a sensação de um personagem bidimensional se movimentando. Pensei em como aplicar esta ideia, e cheguei na conclusão que meu personagem principal seria um boneco de palitinho *skatista*. Logo o movimento de andar horizontalmente poderia ser substituído por uma só imagem dele curvado no *skate*, enquanto que os movimentos de andar verticalmente poderiam ser substituídos por uma única imagem dele dando um “flip” no ar, onde solta o *skate* para um pequeno salto. Esta decisão poupou um imenso esforço em construção e desenhos do personagem, além de deixá-lo ao ver de alguns, ainda mais estiloso.



**Figura 29:** Personagens *skatistas* (fonte própria).

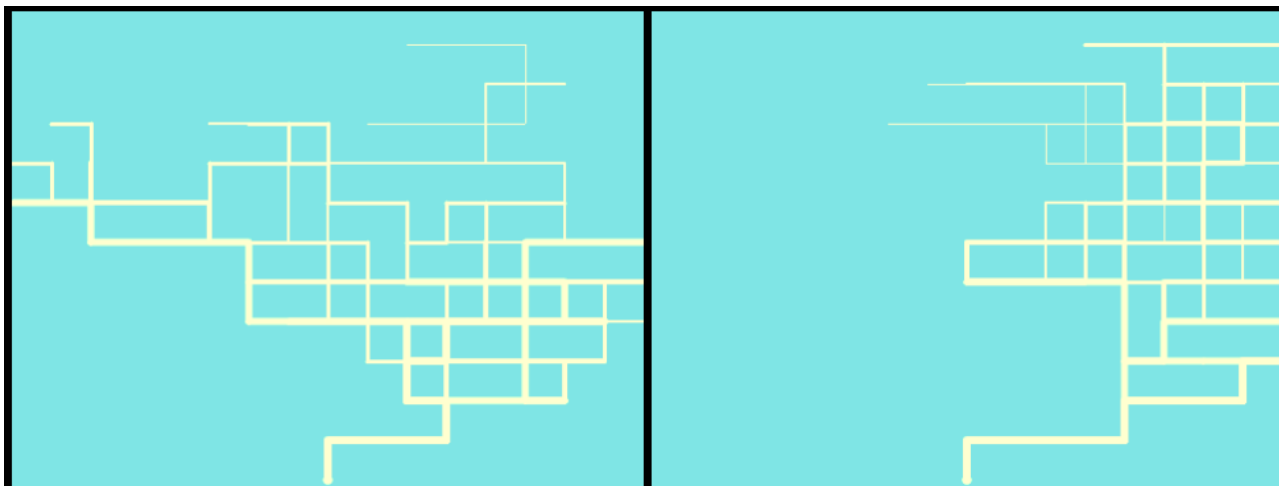
Colocar música no jogo pareceu uma tarefa bem difícil para alguém que não entende nem muito bem aquilo que ouve, assim quem dirá escolher sons agradáveis e coerentes de acordo com os desafios. Cheguei a pensar em uma música de fundo para ajudar no estudo ou na concentração, como uma ópera ou algo bem leve. Porém um problema lógico de usar arquivos de música é o seu tamanho. Enquanto o jogo inteiro tinha menos de 1 megabyte, os arquivos de música em .mp3



ocupariam um espaço bem maior. Embora a memória do computador não seja um problema prático nos dias de hoje, o acesso ao jogo pela internet seria. Pois se o projeto de 1 megabyte passasse a ter 50 megabytes, carregar o jogo online seria equivalente a um download deste tamanho, e isto para a internet que temos em algumas escolas é uma tarefa um pouco mais trabalhosa e lenta.

Uma solução que encontrei pesquisando sobre músicas em jogos digitais, foi compor minha própria melodia. Ou seja, dizer para o *software* que nota tocar a cada momento, de modo que a medida que vá tocando, o jogador perceba na forma de uma música. Tentei compor no Scratch a melodia da Pantera Cor-de-Rosa, composta por Henry Mancini. Porém ainda faltava um ajuste do som com o jogo, dado que embora me agradasse ouvi-la, imaginava que jogar por 30 minutos com este mesmo som seria um tanto cansativo para pessoas menos fascinadas por esta música. Então me veio uma luz. Se o objetivo do jogo era estimular habilidades do Pensamento Computacional, podia estimular o *Paralelismo* (habilidade ligada a realização de várias tarefas independentes ao mesmo tempo), propondo que o jogador acesse uma mídia em outra janela do computador e coloque a música favorita para ouvir enquanto joga. Até mesmo coloquei o cristal azul sugerindo isto ao jogador na tela inicial, mas depois removi esta mensagem e deixei-a apenas como sugestão pessoal a quem fosse jogar. O fato é, que se eu fosse aplicar em escolas ou grupos, precisaria ter fones de ouvido para todos os jogadores ou a música do jogo se tornaria um barulhão no meio de tantos computadores (isto se eles tivessem caixas de som).

Outra adaptação nesta versão, foram as telas de fundo próprias. Pensei em criar um fundo relacionado a circuitos de computador e que não poluíssem a tela com informações demais. Depois de muito pensar sobre o assunto, achei que um fundo aleatório no formato de linhas que começam grossas e vão se dividindo ao acaso em linhas mais finas, seria misterioso e evitaria cansar os olhos. Ainda serviria como um diferencial sutil entre a evolução dos níveis. Assim no próprio *Scratch 2.0* construí um gerador de fundos de tela, selecionei as telas mais bonitas e inseri como fundos do jogo.



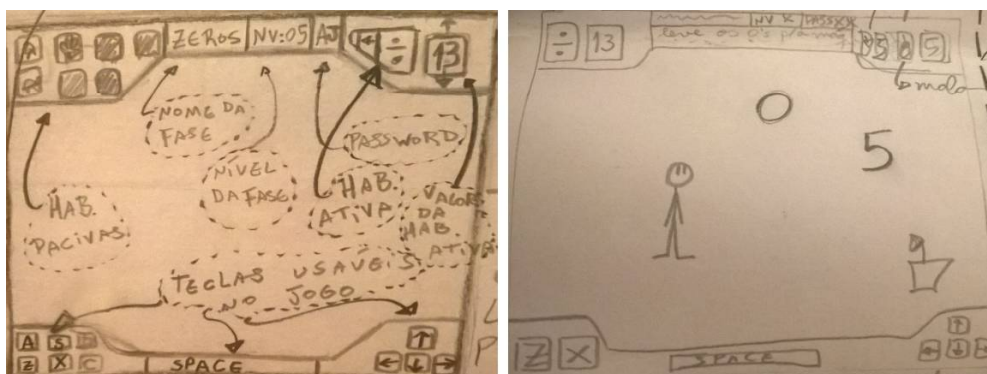
**Figura 30:** Exemplos de telas de fundos usadas no jogo (fonte própria).

De forma geral, determinei os seguintes aspectos visuais como sendo necessários a se considerar:

- a) Idem ao anterior (*O jogo será bidimensional com visão de perfil*);
- b) Idem ao anterior (*A parte do jogo que recebe os objetos carregados pelo personagem (chamada de máquina) será da cor azul, com formato abstrato e terá uma antena em constante movimento*);
- c) Idem ao anterior (*Os números serão pixelados e terão a altura do personagem*);
- d) O protagonista será um boneco de palitinho *skatista* usando um cachecol vermelho;
- e) O personagem será acompanhado de um pequeno cristal azul;
- f) Haverá uma barra inferior na tela com informações ativas sobre o jogo;
- g) A tela de ajuda conterá informações necessárias para todo o jogo;
- h) Quando uma tecla de comando está sendo pressionada, a mesma aparece em vermelho na barra inferior, caso contrário permanece na cor verde-piscina;
- i) Quando ativo o comando *gravar*, a silhueta do personagem permanece na sua posição inicial para orientar o jogador onde ele estava no início deste processo;
- j) É aplicado na máquina um filtro de cor diferente por nível;
- k) A tela final do jogo não possui mensagem de parabéns, porém indica ao jogador como acessar o código do próprio jogo;
- l) Após cumprir uma missão, uma tela cinza surge por 1 segundo antes do próximo nível começar.
- m) Todas as telas de fundo dos níveis serão diferentes porém mantendo o mesmo estilo gráfico;

Também imaginei que a tela de jogo teria informações que reagiriam ativamente ao jogador mudando de cor ou brilhando. Este recurso ajudaria a identificar quando um recurso está sendo usado durante a fase de testes funcionais do jogo, similar a quando inseri no jogo o “cachecol

vermelho” em **Handles** para indicar quando o *Divisor* estivesse ligado. Agora o jogador saberia até mesmo quando pressionasse os direcionais do teclado. Uma vantagem desta abordagem seria registrar e reproduzir a gravação da “tomada de vídeo”, pois como precisaria que o computador jogasse durante este momento, se eu fizesse o personagem reagir a estes “sinais”, então poderia fazer o jogador modificá-los a partir das teclas e o computador modificá-los também. Então, em vez de bloquear as ações do jogador de pressionar teclas, somente impediria que as ações do jogador modificassem as imagens dos botões da tela. Ou seja, tanto o jogador quanto o computador poderiam alterar estas imagens de ações em ocasiões isoladas.



**Figura 31:** Modelos de tela de jogo (fonte própria).

O uso desta lista de ações tomadas pela reação das imagens da tela evitariam também outro tipo de falha gerada pelo tempo de reação físico das teclas pressionadas. Então, registrando exatamente quantas vezes determinado botão trocou de imagem (variação de cor ou brilho), poderia guardar exatamente as mesmas ações que o jogador realizou, assim como replicá-las da mesma maneira, dado que elas em ambos os casos foram geradas a partir da reação das mudanças de suas imagens. Dessa forma, para realizar a construção destes e outros recursos, os seguintes aspectos lógicos foram necessários:

- a) O protagonista poderá ser controlado tanto pelos respectivos direcionais do teclado quanto por uma lista de comandos criada pelo jogador;
- b) Na tela de ajuda, o cristal dirá ao jogador a missão do respectivo nível;
- c) No início ou reinício dos níveis, o personagem ficará paralisado por 2 segundos e o cristal dirá sua missão, a missão some com qualquer tecla que o jogador pressionar após este tempo;
- d) Os extremos da tela estarão conectados com seus lados opostos, fazendo com que o personagem quando avançar por um extremo, apareça no outro;
- e) O jogador possui um comando para *inserir* (chamado de *insert*) na máquina os objetos que estejam tocando nela;
- f) Existem dois tipos de missão, aquelas de *conteúdo* e aquelas de *conteúdo-ordenado*;

- g)** Enquanto o comando *insert* estiver pressionado, o personagem não se moverá pelas teclas dos direcionais;
- h)** O jogador possui um comando para alterar a condição que faz o personagem agarrar números, isto ocorre segurando o comando *insert* e pressionando as teclas de direcionais para direita ou para esquerda, este valor será chamado de *Divisor*;
- i)** O jogador possui um comando para alterar a quantidade de vezes que a gravação dos movimentos será realizada, isto ocorre segurando o comando *insert* e pressionando as teclas de direcionais para cima ou para baixo, este valor será chamado de *Repetir*;
- j)** O jogador possui um comando para criar uma lista de ações para seu personagem realizar sozinho, que será chamado *gravar*;
- k)** O jogador possui um comando para reproduzir de forma autônoma uma lista de ações gravadas anteriormente por *Repetir* vezes, que será chamado *reproduzir*;
- l)** Uma vez ativo o comando *gravar*, todas as ações de movimento ou que alterem o valor do *Divisor* serão registrados nesta lista, isto durará até que uma destas condições seja satisfeita: o nível reinicie, o jogador avance para o próximo nível, o jogador acesse a tela de ajuda, o jogador pressione o comando *reproduzir* (neste caso o *reproduzir* somente encerra a gravação e o personagem é teletransportado para a posição inicial);
- m)** Uma vez ativo o comando *reproduzir*, todas as ações gravadas serão executadas por *Repetir* vezes e o personagem é teletransportado para a posição inicial ao final de cada lista de ações, o processo continuará até que uma destas condições seja satisfeita: o nível reinicie, o jogador avance para o próximo nível, o jogador acesse a tela de ajuda, a lista seja executada *Repetir* vezes, o jogador pressione o comando *gravar* (neste caso o *gravar* somente encerra a execução);
- n)** A variável *Divisor* pertence aos Inteiros e a variável *Repetir* pertence aos Naturais;
- o)** O jogo possui 21 níveis;
- p)** Após a lista de ações ser gravada, o jogo remove da lista as ações do personagem parado;
- q)** Todos os níveis terão números com valores entre 0 e 29 escolhidos aleatoriamente ou predeterminados, com posicionamentos aleatórios ou predefinidos na tela, uma máquina no canto inferior direito, o personagem do jogador localizado a sua esquerda;
- r)** Em alguns níveis os números terão uma aparência indistinguível visualmente;
- s)** Nas missões de *conteúdo*, se todos os números corretos forem inserido na máquina, o jogador avança ao próximo nível, mas se ao menos um número errado for inserido na máquina, o nível atual se reinicia imediatamente;

- t) Nas missões de *conteúdo-ordenado*, não basta que somente os números corretos sejam inseridos na máquina, isto também precisa ocorrer em uma ordem preestabelecida, se ao menos um número errado ou fora de ordem for inserido na máquina, o nível atual se reinicia imediatamente;
- u) Enquanto o personagem estiver executando automaticamente uma lista de ações, o jogador fica impedido de movimentar o personagem e o comando *insert* fica desabilitado;
- v) A partir do nível 13 todos os números ficam indistinguíveis visualmente.

**Produção:** De julho a outubro de 2016 trabalhei na construção do jogo. Seus níveis agora foram construídos de modo mais organizado, estando todos agrupados cada um em um bloco próprio. Alterar as configurações de um nível envolvia modificar somente um bloco. Também comecei a trabalhar com listas de posicionamentos específicos para os números, assim eles se dispersariam de modo aleatório ou determinístico de acordo com o comando desta lista. A parte da produção que foi mais árdua, envolveu construir um condicional que verificasse se os números estavam ou não na ordem determinada pela missão.

No decorrer da construção pude corrigir a ação de cruzar a tela independente da largura do personagem. Embora o cristal azul já não fosse mais necessário, me afeiçoei a ele. Sendo a primeira interface com o usuário do tipo personagem, pois quando o nível iniciava, na prática o personagem dizia ao jogador a missão, mas a altura do balão de fala dava a entender que era o cristal quem estava falando. Decidi colocá-lo também na tela de ajuda para dizer a missão pro jogador.

Outro aspecto a considerar neste jogo, foi a inclusão de uma tela de ajuda com informações completas sobre todos os desafios e mecânicas necessárias para jogar. Para corrigir a dificuldade de acentuação e uso do ç no *Scratch*. Construí esta tela em um *software* externo e importei ao *Scratch*. Porém quando importado um arquivo que não seja em formato *.svg*, o *Scratch* converte automaticamente para uma resolução mais baixa. Dessa forma, a tela de ajuda possuía uma baixa nitidez, não impedindo de ser lida, mas tendo uma distorção nas letras apertadas que explicavam todas as instruções do jogo. Abaixo descrevo de forma sucinta as configurações de cada nível construído:

- 1) Um número aleatório ao centro da tela. A missão do jogador é inserir o número na máquina.
- 2) O número aleatório maior que 0 de um lado da tela e 5 números zeros bloqueando a passagem pelo centro da tela. A missão do jogador é inserir o número maior que 0 na máquina.
- 3) O número aleatório maior que 0 de um lado da tela e 8 números zeros bloqueando a passagem pelo centro da tela. A missão do jogador é inserir o número maior que 0 na máquina.
- 4) O número aleatório maior que 0 de um lado da tela e 12 números zeros bloqueando algumas passagens. A missão do jogador é inserir o número maior que 0 na máquina.

- 5) O número aleatório maior que 0 de um lado da tela e 17 números zeros bloqueando algumas passagens. A missão do jogador é inserir o número maior que 0 na máquina.
- 6) O número 9 de um lado da tela e 8 números 5 bloqueando todas as passagens. A missão do jogador é inserir o número 9 na máquina.
- 7) Três números 7 de um lado da tela e 18 números diferentes de 7 bloqueando todas as passagens. A missão do jogador é inserir os números 7 na máquina.
- 8) Doze números pares e doze números ímpares espalhados pela tela sem sobreposição. A missão do jogador é inserir os números pares na máquina.
- 9) Dezesesseis números pares e dezesesseis números ímpares espalhados pela tela sem sobreposição. A missão do jogador é inserir os números ímpares na máquina.
- 10) Dezesesseis números zeros e dezesesseis números maiores do que zero espalhados pela tela sem sobreposição. A missão do jogador é inserir os números zeros na máquina.
- 11) Dezesesseis números zeros e dezesesseis números maiores do que zero espalhados pela tela sem sobreposição. A missão do jogador é inserir os números maiores do que zero na máquina.
- 12) Dez números ímpares múltiplos de 3 e dez números que não são ímpares múltiplos de 3 espalhados pela tela. A missão do jogador é inserir os ímpares múltiplos de 3 na máquina.
- 13) Dez números que são múltiplos de 4 ou de 7, e dez números que não são múltiplos nem de 4 e nem de 7 espalhados pela tela. A missão do jogador é inserir todos os múltiplos de 4 e todos os múltiplos de 7 na máquina.
- 14) Dez números que são 27 ou 29, e dez números que não são 27 ou 29 espalhados pela tela. A missão do jogador é inserir todos os números 27 e 29 na máquina.
- 15) Dez números que são 14 ou 15, e dez números que não são 14 ou 15 espalhados pela tela. A missão do jogador é inserir todos os números 14 e 15 na máquina.
- 16) Dez números que são múltiplos de 9 e dez números que não são múltiplos de 9 espalhados pela tela. A missão do jogador é inserir todos os múltiplos de 9 separadamente do maior para o menor na máquina.
- 17) Dez números que são múltiplos de 5 e dez números que não são múltiplos de 5 espalhados pela tela. A missão do jogador é inserir todos os múltiplos de 5 separadamente do maior para o menor na máquina.
- 18) Dez números que são pares e dez números que são ímpares espalhados pela tela. A missão do jogador é inserir todos os números pares separadamente do maior para o menor na máquina.
- 19) Dez números que são pares e dez números que são ímpares espalhados pela tela. A missão do jogador é inserir todos os números ímpares separadamente do maior para o menor na máquina.

20) Vinte números aleatórios espalhados pela tela. A missão do jogador é inserir todos os números separadamente do maior para o menor na máquina.

21) Um número 1 e dez números diferentes de 1 espalhados pela tela. A missão do jogador é inserir o número 1 na máquina.

**Testes:** De setembro a novembro de 2016 realizei os testes mais variados desta narrativa. Em setembro ainda, a mecânica de *gravar* as ações do personagem não teleportava o personagem para a posição inicial no final do processo. Considerava que assim o jogador teria mais opções de como usar as rotinas de repetição, tanto com movimentação iterativa como com a movimentação repetitiva. Bastaria que o jogador movesse o personagem para a mesma posição da sua silhueta marcada na tela ao início do processo de *gravar*. Mas meus próprios colegas que conheciam a fundo o projeto, não conseguiam controlar “milimetricamente” o personagem para a mesma posição da silhueta. Eles consideravam isto algo absurdamente difícil e inútil. Afinal, não haviam desafios que exigiam movimentação iterativa. Logo, era uma mecânica que excedia a necessidade dos desafios.

Já em outubro, estava próximo demais do jogo para enxergar se ele era ou não agradável, um efeito de cegueira que diversos game designers mencionam sofrer. Por mais complexos que fossem as habilidades, para mim e para meus colegas que acompanhavam a produção do jogo, pareciam óbvias e simples de usar. Porém era meu interesse não apenas criar um jogo digital envolvendo tópicos da Matemática, mas também um jogo que pudesse ser utilizado como material auxiliar para a aprendizagem de Matemática.

De forma otimista para não dizer ingênua, tentei mostrar que meus jogadores entravam em estado de fluxo enquanto jogavam. Isso me garantiria que o jogo leva os jogadores ao máximo de suas habilidades durante a realização das missões. De forma indireta, se o jogador entra em fluxo para resolver problemas de natureza Matemática, então o jogo facilitaria o exercício da aprendizagem dos conteúdos Matemáticos envolvidos no jogo. O modo como propus demonstrar isto era filmando os rostos dos jogadores e a tela do computador enquanto jogavam. Depois descreveria cada expressão facial e cruzaria com as ações de dentro do jogo. Se conseguisse mostrar que as ações dos jogadores são relacionáveis a suas expressões faciais e depois que estas expressões variavam constantemente no intervalo entre ansiedade e o tédio, então eles estariam em fluxo.

A dificuldade desta abordagem para não dizer a ingenuidade, diz respeito a complexidade de análises. Pra começar eu não tomei como base as expressões naturais dos jogadores para ansiedade e tédio. Assim, precisei criar minha própria medida a partir das expressões dos mesmos jogando. Outra dificuldade foi não ter que considerar apenas expressões faciais, mas também gestos adaptadores (ações com braços, cabeça, pescoço, mãos, ombros) que expressam reações humanas

ao cansaço ou a ansiedade. Isto tornou este um trabalho manual e sem medidas prévias para auxiliar as minhas justificativas. O ponto final que torna esta uma abordagem muito otimista, é que o jogo realmente não colocava os jogadores em fluxo. O jogo nesta fase de desenvolvimento era tedioso ou frustrante demais, os jogadores em poucas ocasiões se animavam. Descartando estes testes como uma forma de demonstrar o potencial de aprendizagem deste jogo.

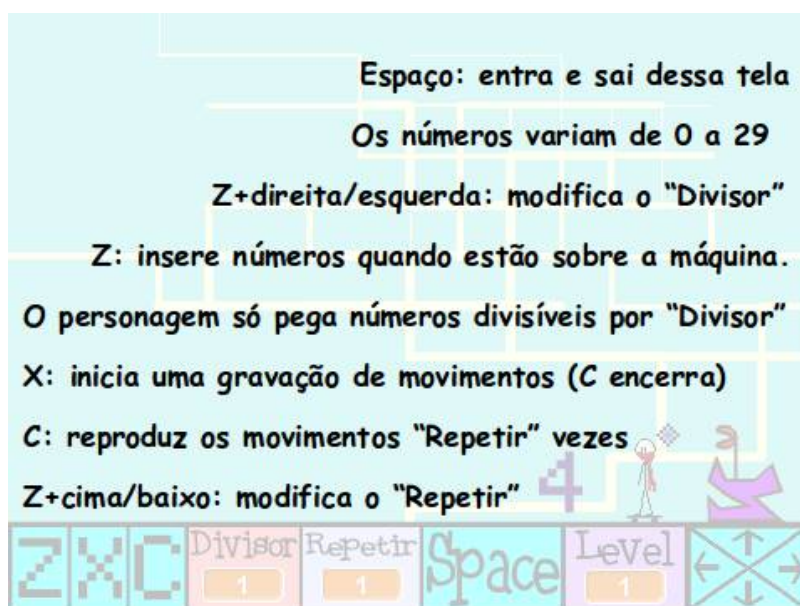
Uma observação curiosa relacionada com o primeiro parágrafo desta seção de testes, é que uma das participantes que teve seu rosto e tela do jogo gravadas jogou de modo muito peculiar. Ela entendeu como pegar números a partir do *Divisor*, porém não entendeu como soltá-los. No nível que precisava agarrar somente os números diferentes de 0, ela utilizou a mecânica de *gravar* para passar pelo 0 sem agarrá-lo ( $Divisor = 0$ ), agarrar o número ( $Divisor = 1$ ) e então encerrava a gravação, fazendo o personagem ser teleportado para a posição inicial carregando o número diferente de 0. Ainda que pensasse já conhecer todas os modos como os jogadores podiam tentar resolver os desafios, esta estratégia me surpreendeu, tanto por ser trabalhosa quanto por utilizar uma mecânica que inseri para fins completamente diferentes de como ela utilizou.

Outra abordagem exaustiva para testar a aprendizagem deste jogo foi colocar jogadores em dupla e gravar o áudio e a tela do jogo. Digo que foi exaustiva pois a análise do diálogo foi feita de forma manual junto com as comparações do que ocorria no jogo em cada período de fala. Percebi com isto que o jogo servia como problema para impulsionar a aprendizagem de como jogar o jogo. Nos diálogos, as falas indicavam que os jogadores apresentavam conceitos errados sobre a Matemática. Como por exemplo, achar de primeira instância que se o *Divisor 2* pegava todos os números pares, o *Divisor 3* pegaria todos os números ímpares. O jogo neste sentido, trouxe aos jogadores problemas de Matemática incomuns ao currículo, que lhes forçava a discutir abordagens diferentes do usual “fazer contas”. Analisando também o intervalo das falas das duplas, pude perceber que os jogadores (que não se conheciam anteriormente) iniciam o jogo falando muito pouco. Mas após enfrentar muitos desafios, estão conversando quase sem intervalos sobre soluções para o jogo. Outro aspecto que vale observar, é a criação de expressões próprias, como por exemplo, uma dupla após se deparar com os números indistinguíveis visualmente, passou a chamá-los de “morceguinhos amassados”, e após algum tempo de jogo, se referia a estes números somente por “morceguinhos”.

No dia 15 de outubro, em comemoração ao dia dos professores, lancei a fã-page deste jogo em uma rede social. Anunciando várias informações de pesquisa relevantes e correlatas aos conteúdos do jogo. E também convidando os usuários, a jogarem, gravarem o vídeo da tela de seus jogos e me enviarem, como forma de fomentar o material desta pesquisa. Recebi retorno de 6 jogadores. Analisando seus jogos, reparei que o principal entrave na época, dizia respeito a forma



como as instruções apareciam. Nesses vídeos os jogadores ficavam vários minutos lendo as instruções e retornando ao jogo para tentar realizar alguma ação. Percebendo que não funcionava, em vez de tentarem outras ações, retornavam para as instruções na tentativa de aprender a jogar corretamente. Isso me fez perceber que as informações auxiliares precisariam ter um cunho muito mais prático e objetivo, ainda que faltassem informações. Assim, era preferível que o jogador tivesse uma dica sucinta de como usar uma habilidade para tentar usá-la na prática e aprender como ela funciona. Do que tentar explicitar em detalhes todo o formalismo Matemático envolvido na jogabilidade. Também percebi que a tela de acesso a todos os níveis do jogo era ignorada, e o menu principal que levava os jogadores a iniciarem estilo **Megaman X**, não trazia uma lição tão essencial que facilitasse o decorrer do jogo, pois no geral todos os jogadores intuitivamente testavam os direcionais do teclado para movimentar o personagem e o levavam até o número que logo percebiam ficar agarrado a ele. Dessa forma, retornei ao contexto simples, onde o jogo inicia com uma tela de instruções, e a primeira das instruções diz respeito a como entrar/sair da tela de instruções. O fundo dessa tela seria o próprio jogo, deixando nítido o personagem, um número, a máquina, o menu de ações em baixo e um espaço em cima para onde o cristal apareceria dizendo a missão daquele nível.



**Figura 32:** Tela de ajuda/Tela inicial de jogo (fonte própria).

Durante o processo de produção, abrindo mão do formalismo que tomava até aquela hora, realizando testes gravados, registrando questionários, entrevistas entre outros, passei a mostrar para algumas pessoas o jogo e sugerir que jogassem livremente na minha presença. Neste período

realizava algumas perguntas sobre o que elas achavam de determinados aspectos do jogo, se entendiam como a habilidade funcionava. Ironicamente estas entrevistas abertas me deram mais indícios do que precisava modificar no jogo do que todos os outros testes que realizei. Quando analisava o jogador ao vivo, de imediato podia perguntar a razão de determinada ação, e logo percebia que uma informação não era lida, ou era mal interpretada e seus motivos.

**Finalização:** A versão foi encerrada em novembro de 2016. Um dos fatores mais evidentes neste jogo era sua falta de envolvimento com os jogadores. Ainda que o cristal azul fosse um personagem que auxiliava o jogador, não havia interação com ele. E o próprio cristal parecia muito distante de causar uma sensibilização nos jogadores. O jogo era mais visto como uma série de desafios intelectuais dentro de uma interface animada com soluções dinâmicas iterativas, do que algo para ser jogado e produzir prazer ao jogador. Analisando de forma geral, levantei os seguintes aspectos positivos e negativos deste jogo:

- a) A mecânica de automação e de alterar o critério do *Divisor* permite resolver uma série de problemas Matemáticos complexos;
- b) O potencial de solucionar problemas automaticamente deixava o jogador (quando aprendia a usar) fascinado. O jogador não queria mais resolver os desafios de forma manual, se esforçando até mais do que o necessário em uma solução manual, apenas para gerar uma estrutura automática que resolva o problema sozinho e observar ela trabalhando;
- c) Eram poucos os níveis que estimulavam o jogador a usar a habilidade de criar rotinas automáticas. Isto fazia com que no decorrer do jogo, a existência dessa habilidade permanecesse confusa e gerasse até falsas hipóteses sobre seu funcionamento;
- d) As instruções eram absolutas para todo o jogo, logo havia mais informações do que um jogador iniciante precisava para jogar, o que deixava o início do jogo o percurso mais difícil de entender;
- e) A falta de envolvimento proporcionado pelo jogo deixava os jogadores apáticos mesmo quando superavam desafios, faltando uma parabenização ou motivação mediante os sucessos;
- f) A falta de envolvimento proporcionado pelo jogo deixava os jogadores frustrados quando falhavam, faltando elementos que desviassem sua atenção, isto somente aumentava a frustração;
- g) A falta de envolvimento proporcionado pelo jogo deixava os jogadores desanimados, as vezes demorando vários minutos analisando o contexto antes de se arriscar em alguma ação de jogo;

## Handles in Scratch 2.0

Esta versão contou com três grandes diferenciais. O aumento nas funções tanto do personagem quanto dos elementos do cenário. A interatividade com o usuário a partir do mecanismo de *feedback* Amanda. A modularização das componentes do jogo.

Conforme percebi na versão predecessora, a habilidade de criar rotinas de repetição aparecia na prática em poucos níveis. Propor mais desafios envolvendo o processo iterativo e repetitivo colocava em risco a trivialidade do jogo. Pois o jogador já teria dominado estas dinâmicas, precisando apenas executá-la e esperar mais tempo para sua conclusão. Por outro lado, se os novos desafios envolvessem algoritmos mais complexos, o jogador poderia não ter um *insight* e preferir resolvê-lo manualmente, deixando a resolução cansativa. A solução encontrada para isto foi investir em desafios iterativos e repetitivos simples, porém utilizando novos elementos de cenário interativo e habilidades do personagem. Assim mesmo simples, o jogador precisaria se desdobrar para aprender como jogar nestas novas circunstâncias, deixando o jogo pouco repetitivo.

**Pré-produção:** A ideia para a interatividade com o mecanismo de *feedback* Amanda surgiu em um curso de animação para jogos 2-D realizado em dezembro de 2016. Nele a professora exemplificou mecanismos de *feedback*. Enquanto assistia percebi que os mecanismos de *feedback* podem ser colocados como variáveis nominais, ordinais, discretas ou contínuas. Assim mecanismos comuns de *feedback* são discretos como placares, pontos de vida, número de corações. Mas muitos jogos também utilizam variáveis do tipo ordinal, como o personagem mancando um pouco quando perde vida, e mau conseguindo andar quando perde muita vida. Mas são poucos os jogos que trabalham com variáveis nominais. Ou seja, variáveis que não podem ser diretamente associadas com um grau de intensidade.

Em todo este percurso de construção, tentava fugir das variáveis discretas como placares, pois eles são irrelevantes com o jogo. Posso dar 1000 pontos ou 1, sem uma escala da qual estes pontos possam ser usados, isto é ilógico. Outro motivo para querer evitar placares é o possível interesse do jogador adquirir um placar mais alto, reiniciando o jogo para não cometer os erros já conhecidos. Também não era minha intenção colocar um limite de erros aos jogadores, senão não faria sentido estimulá-los a se arriscar testando suas hipóteses. Logo, não podia dar um *feedback* ordinal, pois em algum momento chegaria no valor mínimo desta ordem, não podendo piorar mais. E sem o *game over*, esta consequência ordinal também não faria sentido. Afinal, para que se preocupar com o estado de saúde de um personagem se ele não morre e na prática sua saúde não interfere em nada no jogo. Porém uma variável nominal por não possuir uma ordenação natural, me permitiria colocar uma pressão no jogador ao mesmo tempo que parabenizá-lo a altura do feito, sem

ter que utilizar pontos ou medidas. Evitando assim que os jogadores pensem recomeçar o jogo para atingir melhores desempenhos ou se preocupem em arriscar novas ideias para não levar o jogo ao game over.

Pensando em variáveis nominais, imaginei colocar como mecanismo de *feedback* do jogo cenas de personagens batendo a cabeça na mesa quando o jogador errasse. E cenas do mesmo personagem sorrindo frente ao computador quando o jogador obtêm sucesso. O problema desta abordagem foi encontrar imagens nitidamente claras sobre estas ações e que fossem leves no efeito que poderiam provocar no jogador. Uma alternativa que escolhi, foi procurar folhas de expressões humanas desenhadas em mangá ao estilo chibi. Esta escolha se deu por conta de uma amiga que costumava responder as mensagens nas redes sociais não com palavras, mas com imagens das expressões de uma menina desenhada em mangá ao estilo chibi.

Escolhi a folha de expressões que achei mais neutra e com imagens nítidas para usar no jogo. Porém, o site original desta fonte, está em um idioma que sou incapaz de compreender para sequer atribuir-lhe crédito, no máximo, apresento a imagem desta folha e o link de onde eu a encontrei <<https://crazytattoo.ru/enciklopediya-tatuirovok/stili/tatu-v-stile-anime/>> (acessado em 15, jan, 2018).

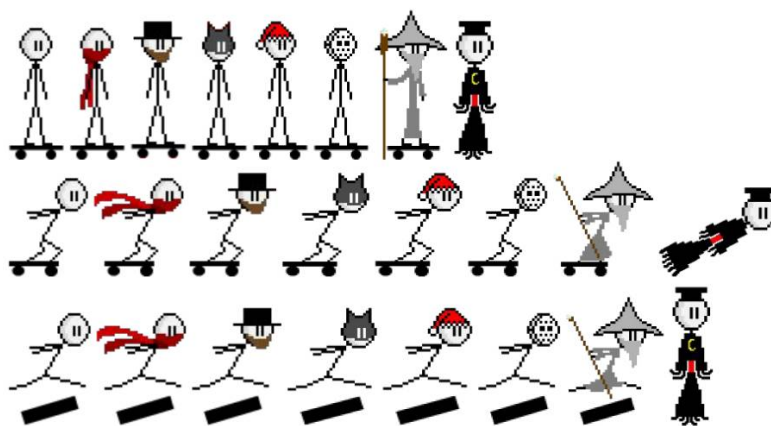


**Figura 33:** Folha de expressões escolhida.

Atribui para este mecanismo de *feedback* o nome Amanda, por ser um nome comum no Brasil, não estar em alta nos noticiários e pelo seguinte jogo de palavras, “Quem manda levar os objetos até a máquina? A Amanda manda”. Esta folha tinha 22 expressões, das quais escolhi 3 grupos de 6 que considerava próximos de formar uma escala para os critérios de tédio, raiva e felicidade.

Em relação a modularidade do código, tomei com atenção cada componente e cada variável utilizada, na tentativa de reduzir chamadas redundantes. Também procurei encurtar o código criando blocos de funções próprias a serem repetidas para as ações que apareciam várias vezes com poucas alterações. Por último, construí um gerador de níveis que gerenciava como cada nível seria montado mediante uma lista de códigos previamente estabelecida. Isto tornou a estrutura do jogo bastante simples, facilitando assim a identificação de falhas, a correção e alteração de elementos do jogo.

Procurando melhorar a identificação do jogador com o personagem, achei apropriado que este pudesse escolher um visual mais agradável. Assim, investi em uma série de novas fantasias para o mesmo personagem desenhado em palitinho. A forma como as produzia era voltada para sua divulgação, desde Abraham Lincoln, o dia dos professores (Batman é o símbolo do curso Licenciatura em Ciências Exatas da USP – São Carlos), natal, sexta-feira 13, Gandalf o Cinzento, e o personagem protagonista do anime Ansatsu Kyoushitsu (que é um professor).



**Figura 34:** Variações do personagem (fonte própria).

Tendo em mente otimizar aquilo que já havia se mostrado eficaz na versão anterior, procurei melhorar as instruções e sensibilização aos jogadores. Entre outras modificações, a barra inferior foi movida para a lateral esquerda, deixando a área de jogo com dimensões mais próximas a um quadrado. Sua variação de cores também foi reduzida para a cor preta, e nela foram colocadas as mesmas instruções da versão anterior, somada pelo mecanismo de *feedback* Amanda, e instruções para acessar a tela de ajuda. Procuramos também relacionar a forma com a função, isto para os novos objetos arrastáveis e dos equipamentos no cenário. Assim uma balança teria espaço para receber os objetos e compará-los enquanto um escâner tem uma lâmpada para indicar o resultado. De modo geral, determinamos os seguintes aspectos visuais como necessários:

**a)** Idem ao anterior (*O jogo será bidimensional com visão de perfil*);

- b)** Idem ao anterior (*Todas as telas de fundo dos níveis serão diferentes porém mantendo o mesmo estilo gráfico*);
- c)** Idem ao anterior (*Quando ativo o comando gravar, a silhueta do personagem permanece na sua posição inicial para orientar o jogador onde ele estava no início deste processo*);
- d)** Idem ao anterior (*Após cumprir a missão, uma tela cinza surge por 1 segundo antes do próximo nível começar*);
- e)** Idem ao anterior (*Os números serão pixelados e terão a altura do personagem*);
- f)** A parte do jogo que recebe os objetos carregados pelo personagem (chamada de máquina) será da cor azul, com formato abstrato, terá 4 luzes piscando e uma antena em constante movimento;
- g)** O protagonista será um boneco de palitinho com diferentes visuais disponíveis ao jogador;
- h)** Haverá uma barra preta do lado esquerdo da tela com informações ativas sobre o jogo;
- i)** Haverão diferentes telas de ajuda, cada uma com informações necessárias para um determinado intervalo de níveis;
- j)** A tela final do jogo possui uma mensagem de parabéns ao jogador, indica o endereço da fã-page do jogo e explica como acessar o código do próprio jogo;
- k)** É aplicado na tela de fundo, máquina, no escâner, na balança, nos números e nos bolinhos o mesmo filtro de cor, que se altera de acordo com o nível do jogo;
- l)** A parte do jogo que recebe os bolinhos, verifica se ao menos um deles valida uma propriedade e devolve o resultado ao jogador (chamado de escâner) será da cor cinza, com formato retangular e terá uma luz que mudará de cor caso a propriedade seja validada;
- m)** As duas partes do jogo que recebem os bolinhos, soma seus pesos e compara uma com a outra para devolver ao jogador qual lado é o mais pesado (chamado de balança) terá seu eixo verde, suas regiões para incluir bolinhos na cor cinza, dois pratos de cada lado onde os bolinhos se posicionarão e um arco redondo em cima de cada região;
- n)** O mecanismo de *feedback* deste jogo é representado por uma personagem de mangá desenhada ao estilo chibi com 18 diferentes expressões faciais que chamamos de Amanda.

O aumento das mecânicas do personagem, a forma como o personagem e os objetos interagiriam com os novos elementos do cenário exigiu a construção de um novo segmento ao jogo. Podendo ser encarado até mesmo como independente dos problemas Matemáticos que o precedem. Entre os aspectos que precisaram de maior atenção, foram aqueles envolvendo a solução direta do problema, pois agora que tratávamos de objetos todos iguais coletados em ordem, precisávamos garantir que os equipamentos funcionariam de modo objetivo e seguro. Pois uma falha não seria mais tão facilmente percebido quanto era com o critério do *Divisor*. Assim, demos prioridade para o

desenvolvimento detalhado destas componentes no plano diretor. Dessa forma, para realizar a construção destes recursos, determinamos os seguintes aspectos lógicos como necessários:

- a)** Idem ao anterior (*O protagonista poderá ser controlado tanto pelos respectivos direcionais do teclado quanto por uma lista de comandos criada pelo jogador*);
- b)** Idem ao anterior (*Os extremos da tela estarão conectados com seus lados opostos, fazendo com que o personagem quando avançar por um extremo, apareça no outro*);
- c)** Idem ao anterior (*O jogador possui um comando para inserir (chamado de insert) na máquina os objetos que estejam tocando nela*);
- d)** Idem ao anterior (*Existem dois tipos de missão, aquelas de conteúdo e aquelas de conteúdo-ordenado*);
- e)** Idem ao anterior (*Enquanto o comando insert estiver pressionado, o personagem não se moverá pelas teclas dos direcionais*);
- f)** Idem ao anterior (*O jogador possui um comando para alterar a condição que faz o personagem agarrar números, isto ocorre segurando o comando insert e pressionando as teclas de direcionais para direita ou para esquerda, este valor será chamado de Divisor*);
- g)** Idem ao anterior (*O jogador possui um comando para alterar a quantidade de vezes que a gravação dos movimentos será realizada, isto ocorre segurando o comando insert e pressionando as teclas de direcionais para cima ou para baixo, este valor será chamado de Repetir*);
- h)** Idem ao anterior (*O jogador possui um comando para criar uma lista de ações para seu personagem realizar sozinho, que será chamado gravar*);
- i)** Idem ao anterior (*O jogador possui um comando para reproduzir de forma autônoma uma lista de ações gravadas anteriormente por Repetir vezes, que será chamado reproduzir*);
- j)** Idem ao anterior (*Uma vez ativo o comando gravar, todas as ações de movimento ou que alterem o valor do Divisor serão registrados nesta lista, isto durará até que uma destas condições seja satisfeita: o nível reinicie, o jogador avance para o próximo nível, o jogador acesse a tela de ajuda, o jogador pressione o comando reproduzir (neste caso o reproduzir somente encerra a gravação e o personagem é teletransportado para a posição inicial)*);
- k)** Idem ao anterior (*Uma vez ativo o comando reproduzir, todas as ações gravadas serão executadas por Repetir vezes e o personagem é teletransportado para a posição inicial, ou até que uma destas condições seja satisfeita: o nível reinicie, o jogador avance para o próximo nível, o jogador acesse a tela de ajuda, o jogador pressione o comando gravar (neste caso o gravar somente encerra a execução)*);
- l)** A lista de ações somente guarda as ações do personagem em movimento ou alterando variáveis;
- m)** O jogo possui 42 níveis;

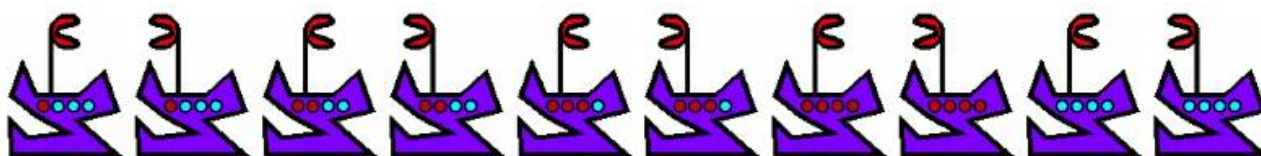
- n)** No início ou reinício dos níveis, a personagem Amanda dirá por 2 segundos a missão, a missão some com qualquer tecla que o jogador pressionar após este tempo;
- o)** O jogador possui um comando para alterar a quantidade de objetos que o personagem consegue carregar, isto ocorre segurando o comando insert e pressionando as teclas de direcionais para direita ou para esquerda, este valor será chamado de *Mochila*.
- p)** A variável *Divisor* pertence aos Inteiros, as variáveis *Repetir* e *Mochila* pertencem aos Naturais;
- q)** Temos dois tipos de objetos agarráveis pelo personagem, bolinhos e números;
- r)** Nas missões de *conteúdo*, se todos os objetos (bolinhos ou números) corretos forem inseridos na máquina, o jogador avança ao próximo nível, mas se ao menos um objeto errado for inserido na máquina, o nível atual se reinicia imediatamente;
- s)** Nas missões de *conteúdo-ordenado*, não basta que somente os objetos corretos sejam inseridos na máquina, isto também precisa ocorrer em uma ordem preestabelecida, se ao menos um objeto errado ou fora de ordem for inserido na máquina, o nível atual se reinicia imediatamente;
- t)** Todos os níveis terão bolinhos visualmente idênticos, ou números entre 0 e 29 escolhidos aleatoriamente ou predeterminados e que também podem ser indistinguíveis visualmente;
- u)** Todos os níveis possuem objetos dispostos na tela de forma aleatória ou predefinida, uma máquina no canto inferior direito, o personagem do jogador localizado a sua esquerda;
- v)** Alguns níveis possuem também um escâner localizado no canto inferior esquerdo ou uma balança centralizada na parte inferior da tela;
- w)** O escâner quando detecta um bolinho com a propriedade determinada, desloca todos os bolinhos que estão em contato para a sua direita;
- x)** A balança é composta por duas partes, seu lado direito e seu lado esquerdo, cada lado possui dois pratos, um próximo ao eixo e outro mais afastado;
- y)** A balança soma o peso de todos os bolinhos colocados em cada lado e compara um lado com o outro, no lado com maior peso os bolinhos são deslocados para o prato mais afastado, no lado com menor peso (ou quando ambos os lados têm peso igual) os bolinhos são deslocados para o prato mais próximo do eixo;
- z)** O jogador possui um comando para ver outras telas de ajuda além daquela considerada adequada ao intervalo atual de níveis;
- aa)** O mecanismo de *feedback* Amanda reagirá ao tempo que o jogador leva para iniciar ou reiniciar um nível da seguinte maneira, ela mudará sua imagem para expressões de tédio ou sonolência em até 6 intensidades;
- ba)** O mecanismo de *feedback* Amanda reagirá aos reinícios sucessivos de um nível pelo jogador da seguinte maneira, ela mudará sua imagem para expressões de raiva em até 6 intensidades;



- ca) O mecanismo de *feedback* Amanda reagirá aos inícios sucessivos de níveis pelo jogador da seguinte maneira, ela mudará sua imagem para expressões de felicidade em até 6 intensidades;
- da) Enquanto o personagem estiver executando automaticamente uma lista de ações, o jogador fica impedido de movimentar o personagem e o comando *insert* fica desabilitado;
- ea) O jogador possui um comando para trocar o visual do seu personagem;
- fa) Alterar a variável *Divisor* somente é possível até o nível 31 do jogo;
- ga) Alterar a variável *Mochila* somente é possível a partir do nível 32 do jogo;
- ha) Na barra preta de informações, está registrado os valores atuais do nível, total de níveis, *Divisor*, *Repetir*, *Mochila*, total de itens que o personagem carrega.
- ia) A partir do nível 22 todos os números ficam indistinguíveis visualmente.

**Produção:** De dezembro de 2016 até março de 2017 trabalhei no mecanismo de *feedback* e na modularização deste jogo. Deixando de lado a ideia de construir as mecânicas de bolinhos, balança e escâner. Pois começava a considerar estes como sendo desafios complexos demais, afinal poucos jogadores conseguiam atingir o nível 31. E como a jogabilidade muda por completo a partir deste nível, achava até desnecessário investir neste segmento.

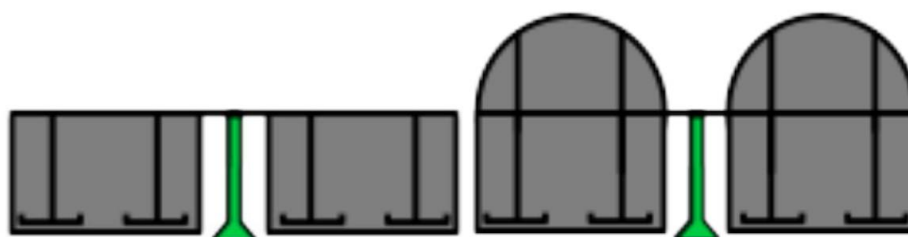
Mas em março de 2017 escrevi um artigo conceitual para uma revista de Matemática propondo este jogo como um recurso para trabalhar métodos numéricos a partir das dinâmicas da balança e do escâner. Tive a felicidade deste artigo ser aceito e para publicá-lo de acordo com o que propunha, precisava habilitar estas mecânicas no jogo. Logo resgatei este segmento ao jogo. Com isto veio também uma repaginada na estrutura das máquinas do jogo, no caso a máquina principal onde os números são inseridos, pensei que ficaria melhor com algumas luzes sequenciais acendendo e apagando. Se antes seu formato já era fácil de identificar, agora seria ainda mais.



**Figura 35:** Máquina com luzes sequenciais (fonte própria).

Reverendo o design da antiga balança, percebi que não precisava de algo tão pomposo, mudando sua posição de acordo com a comparação dos pesos. Somente deveria gerenciar os objetos que estão nela, para que ocupem posições diferentes. A ideia aqui é ter dois pratos, um mais próximo do eixo, e outro mais afastado. O prato mais afastado significa que aquele lado é o mais pesado. Caso ambos em ambos os lados os objetos estejam próximos do eixo, significa que eles

possuem pesos iguais. Então de forma a simplificar a inserção do objeto na balança, substitui ela por um retângulo cinza com espaços para o prato conectados com seu topo. Porém durante os testes de uso automático dela, inserir um objeto era bastante difícil, visto que o personagem precisava tocar na balança no lado certo do prato, e ao afastar-se ele poderia não colocar o objeto no prato ou agarrar de um outro prato sem querer. A solução para este problema técnico foi inserir um arco acima de cada lado da balança. Assim, na hora de criar soluções automáticas, o jogador poderia colocar os bolinhos a partir do arco, que fica longe o bastante dos pratos, não interferindo no sistema.



**Figura 36:** Novos modelos de balança para o jogo (fonte própria).

Também decidi substituir as barreiras de zeros por barreiras de 1 nos níveis iniciais. A escolha por esta mudança tem a ver com desajustes acidentais na variável *Divisor*. Se o jogador sem querer viesse a mudar o *Divisor*, no caso da barreira de zeros, com exceção do *Divisor* igual a 0, toda a barreira continuaria sendo agarrada pelo personagem, porém não há garantia alguma de que o número a ser levado para a máquina também seja. Isto poderia gerar uma tremenda confusão do jogador não entender e se atrapalhar mais do que deveria neste instante inicial do jogo.

Por outro lado, trocando a barreira para números 1. Se o jogador viesse a alterar o *Divisor* (que inicialmente marca 1) para 0, ele passaria por todos os números. Se alterasse para -1 nada mudaria. E se alterasse para qualquer outro número, ele passaria por todos os números 1. Essa mudança sutil poderia acelerar a aprendizagem do jogador sobre o uso do *Divisor*, ainda que de maneira errada. O jogador poderia utilizar o *Divisor* igual ao valor do número que precisa buscar, e ignoraria todas as barreiras de números 1. Substituindo nestes níveis o objetivo de treinar sua coordenação motora para controlar o personagem, por uma introdução a manipulação do *Divisor*. Uma habilidade que no decorrer do jogo será muito mais útil que a coordenação motora avançada. A seguir descrevo de forma sucinta as configurações de cada nível construído:

- 1) Um número aleatório ao centro da tela. A missão do jogador é inserir o número na máquina.
- 2) O número aleatório maior que 1 de um lado da tela e 6 números 1 bloqueando a passagem pelo centro da tela. A missão do jogador é inserir o número maior que 1 na máquina.

- 3) O número aleatório maior que 1 de um lado da tela e 8 números 1 bloqueando a passagem pelo centro da tela. A missão do jogador é inserir o número maior que 1 na máquina.
- 4) O número aleatório maior que 1 de um lado da tela e 13 números zeros bloqueando algumas passagens. A missão do jogador é inserir o número maior que 1 na máquina.
- 5) O número aleatório maior que 1 de um lado da tela e 13 números zeros bloqueando algumas passagens. A missão do jogador é inserir o número maior que 1 na máquina.
- 6) O número aleatório maior que 1 de um lado da tela e 15 números zeros bloqueando algumas passagens. A missão do jogador é inserir o número maior que 1 na máquina.
- 7) O número aleatório maior que 1 de um lado da tela e 23 números zeros bloqueando algumas passagens. A missão do jogador é inserir o número maior que 1 na máquina.
- 8) O número 9 no centro da tela e 8 números 5 bloqueando todas as passagens. A missão do jogador é inserir o número 9 na máquina.
- 9) Seis números 7 no centro da tela e 14 números diferentes de 7 bloqueando todas as passagens. A missão do jogador é inserir os números 7 na máquina.
- 10) Seis números múltiplos de 7 no centro da tela e 14 números que não são múltiplos de 7 bloqueando todas as passagens. A missão do jogador é inserir os números múltiplos de 7 na máquina.
- 11) Seis números múltiplos de 4 no centro da tela e 14 números que não são múltiplos de 4 bloqueando todas as passagens. A missão do jogador é inserir os números múltiplos de 4 na máquina.
- 12) Um número 8 na mesma posição inicial do personagem e um número 5 no centro. A missão do jogador é inserir o número 5 na máquina.
- 13) Um número zero na mesma posição inicial do personagem e um número 3 no centro. A missão do jogador é inserir o número 3 na máquina.
- 14) Doze números 1 e doze números 2 espalhados pela tela sem sobreposição. A missão do jogador é inserir os números 2 na máquina.
- 15) Doze números 1 e doze números 3 espalhados pela tela sem sobreposição. A missão do jogador é inserir os números 3 na máquina.
- 16) Doze números 1 e doze números 2 espalhados pela tela sem sobreposição. A missão do jogador é inserir os números 1 na máquina.
- 17) Doze números 1 e doze números 3 espalhados pela tela sem sobreposição. A missão do jogador é inserir os números 1 na máquina.
- 18) Doze números pares e doze números ímpares espalhados pela tela sem sobreposição. A missão do jogador é inserir os números pares na máquina.

- 19) Dezesesseis números pares e dezesesseis números ímpares espalhados pela tela sem sobreposição. A missão do jogador é inserir os números ímpares na máquina.
- 20) Dezesesseis números zeros e dezesesseis números maiores do que zero espalhados pela tela sem sobreposição. A missão do jogador é inserir os números zeros na máquina.
- 21) Dezesesseis números zeros e dezesesseis números maiores do que zero espalhados pela tela sem sobreposição. A missão do jogador é inserir os números maiores do que zero na máquina.
- 22) Dez números múltiplos de 3 e dez números que não são múltiplos de 3 espalhados pela tela. A missão do jogador é inserir os números múltiplos de 3 na máquina.
- 23) Dez números que são múltiplos de 4 ou de 7, e dez números que não são múltiplos nem de 4 e nem de 7 espalhados pela tela. A missão do jogador é inserir todos os múltiplos de 4 e todos os múltiplos de 7 na máquina.
- 24) Dez números que são 27 ou 29, e dez números que não são 27 ou 29 espalhados pela tela. A missão do jogador é inserir todos os números 27 e 29 na máquina.
- 25) Dez números que são 14 ou 15, e dez números que não são 14 ou 15 espalhados pela tela. A missão do jogador é inserir todos os números 14 e 15 na máquina.
- 26) Dez números que são múltiplos de 9 e dez números que não são múltiplos de 9 espalhados pela tela. A missão do jogador é inserir todos os múltiplos de 9 separadamente do maior para o menor na máquina.
- 27) Dez números que são múltiplos de 5 e dez números que não são múltiplos de 5 espalhados pela tela. A missão do jogador é inserir todos os múltiplos de 5 separadamente do maior para o menor na máquina.
- 28) Dez números que são pares e dez números que são ímpares espalhados pela tela. A missão do jogador é inserir todos os números pares separadamente do maior para o menor na máquina.
- 29) Dez números que são pares e dez números que são ímpares espalhados pela tela. A missão do jogador é inserir todos os números ímpares separadamente do maior para o menor na máquina.
- 30) Vinte números aleatórios espalhados pela tela. A missão do jogador é inserir todos os números separadamente do maior para o menor na máquina.
- 31) Um número 1 e dez números diferentes de 1 espalhados pela tela. A missão do jogador é inserir o número 1 na máquina.
- 32) Dois bolinhos espalhados pela tela e um escâner. A missão do jogador é inserir o bolinho contaminado na máquina.
- 33) Quatro bolinhos espalhados pela tela e um escâner. A missão do jogador é inserir o bolinho contaminado na máquina.

- 34) Oito bolinhos espalhados pela tela e um escâner. A missão do jogador é inserir o bolinho contaminado na máquina.
- 35) Dezesesseis bolinhos espalhados pela tela e um escâner. A missão do jogador é inserir o bolinho contaminado na máquina.
- 36) Dezesesseis bolinhos espalhados pela tela e um escâner. A missão do jogador é inserir os dois bolinhos contaminados na máquina.
- 37) Dezesesseis bolinhos espalhados pela tela e um escâner. A missão do jogador é inserir os três bolinhos contaminados na máquina.
- 38) Três bolinhos espalhados pela tela e uma balança. A missão do jogador é inserir o bolinho mais pesado na máquina.
- 39) Nove bolinhos espalhados pela tela e uma balança. A missão do jogador é inserir o bolinho mais pesado na máquina.
- 40) Vinte e sete bolinhos espalhados pela tela e uma balança. A missão do jogador é inserir o bolinho mais pesado na máquina.
- 41) Vinte e seis bolinhos espalhados pela tela e uma balança. A missão do jogador é inserir os dois bolinhos mais pesados na máquina.
- 42) Vinte e sete bolinhos espalhados pela tela e uma balança. A missão do jogador é inserir os três bolinhos mais pesados na máquina.

**Testes:** De dezembro de 2016 a junho de 2017 realizei testes do jogo com o objetivo de avaliar como os jogadores aprendiam a jogar e se os jogadores aprendiam os conteúdos Matemáticos envolvidos. Os primeiros testes envolviam um questionário para medir se o jogador possuía domínio nos conteúdos Matemáticos envolvidos no jogo. Então colocava o jogador para jogar com um *joystick* conectado ao computador com a tela sendo filmada. Por último realizava uma entrevista estruturada para descobrir se o jogador experimentou alguma das características de quem vivencia o estado de fluxo.

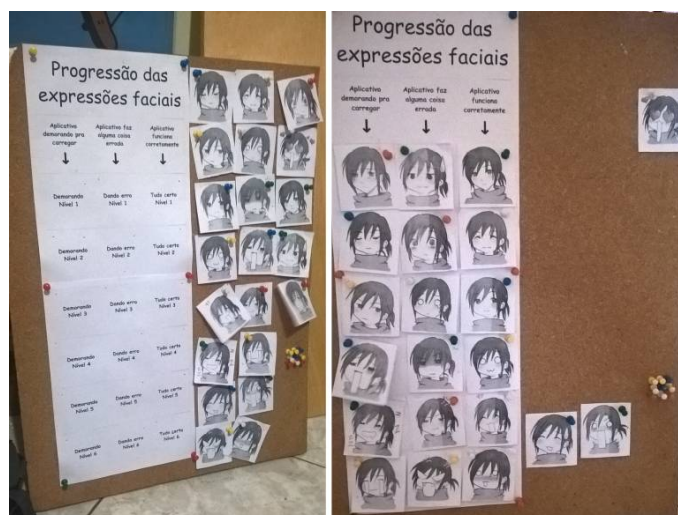
A primeira análise do questionário e da entrevista, já me deu os indicativos sobre seus próprios defeitos, me levando de imediato a adaptação e aplicação de novos, questionário e entrevista. Então analisando-os pude perceber que os participantes dominavam os aspectos da Matemática envolvidos no jogo, mas no decorrer do jogo tinham dificuldades em aplicar aqueles conhecimentos. Também notei que ocorria uma acentuada perda da noção temporal, os participantes ficavam jogando por uma hora mas achavam ter passado cerca de 20 minutos.

Como informação complementar, percebi que o *joystick* facilitava o uso dos comandos se comparado ao teclado do computador. Os jogadores de imediato sabiam como mover o personagem

e não se confundiam sobre a função de seus comandos. Porém ainda apresentavam as mesmas dificuldades e noções Matemáticas errôneas na hora de construir as estratégias dos desafios. Uma observação ao acaso na entrevista trouxe uma hipótese até então não considerada. Uma das participantes se sentiu extremamente pressionada com a presença do mecanismo de *feedback* Amanda. Considerando-a até mesmo estressante. Enquanto outro participante do gênero masculino comentou ter percebido a personagem, porém nem ter reparado que ela mudava suas expressões. Outra hipótese observada nestes testes com *joystick*, foi a dificuldade de um professor de Matemática para resolver os desafios, em comparação com uma psicóloga. Pois enquanto era de se esperar que a dificuldade no jogo fosse resolver seus problemas de Matemática a partir de uma boa base no tema, na verdade isto aparecia de outra maneira, a habilidade em pensar lateralmente e arriscar novas hipóteses parecia auxiliar mais na solução dos desafios do que uma boa base. No caso, o professor de Matemática tentava resolver os desafios mentalmente e aplicar com o personagem no jogo uma solução objetiva, enquanto a psicóloga testava algumas hipóteses e falhava como forma de entender melhor o funcionamento do próprio jogo. O resultado desta análise, é que em um curto espaço de tempo a psicóloga adquiriu um domínio das funções do jogo e conseguia encontrar soluções rapidamente, enquanto o professor de Matemática apresentava uma dificuldade em pensar dentro das dimensões do jogo, onde errar é essencial para aprender.

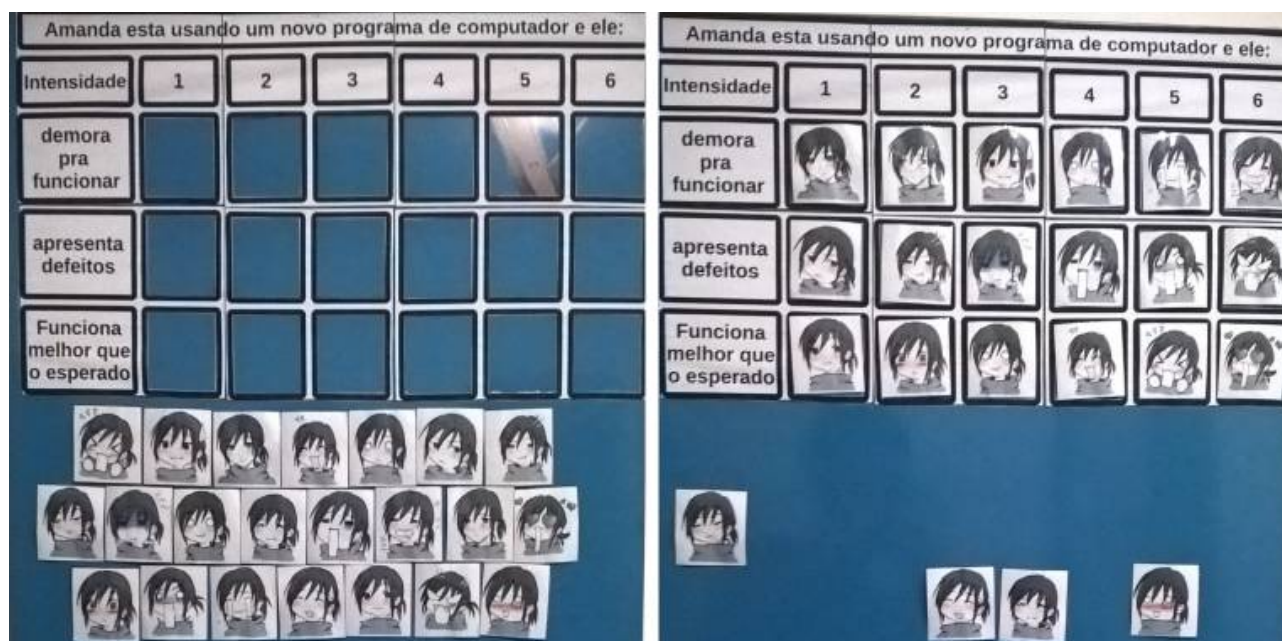
Em relação ao estresse causado pelo mecanismo de *feedback* Amanda, decidi investigar como forma de encontrar uma ordenação para seus rostos que fosse mais facilmente aceita pelos jogadores de ambos os gêneros. Isto significava que o jogador deveria identificar a partir de um rosto qual era a reação que ele estava indicando. Também julguei importante analisar como a interpretação variava de acordo com o gênero de quem via. Visto que o participante do gênero masculino sequer percebeu, logo os rostos escolhidos poderiam não estar de acordo com os significados por ele interpretados. Outro aspecto do qual incentivou esta pesquisa, é a seleção de um padrão de ordenação mais refinado, dado que até então havia aplicado uma ordem de significados baseado somente na minha opinião.

O plano para reunir opiniões era circular a universidade com um painel de tachinhas contendo os 22 rostos e um quadro dividido em 3 categorias e 6 intensidades das emoções. Com este material, pediria aos voluntários que preenchessem movendo os rostos que consideram mais de acordo com cada significado.



**Figura 37:** Painel de tachinhas para identificar rostos da Amanda (fonte própria).

A dificuldade em utilizar o painel de tachinhas estava ligado a movimentação das imagens e eventualmente o vento dos locais abertos. De forma a solucionar este problema, construí um painel semelhante em uma placa de metal e com os 22 rostos plastificados contendo um pequeno ímã em seu verso. Assim eles podiam ser movimentados com uma maior agilidade, tornando o preenchimento do quadro mais rápido e natural.



**Figura 38:** Painel imantado para identificar rostos da Amanda (fonte própria).

Apesar da disposição deste recurso tangível, em virtude de um amigo que quis me ajudar nesta pesquisa de opinião pela internet, construí uma versão simplória onde cada um dos rostos possuía um número associado e lhe enviei uma imagem do quadro para que preenchesse com os

respectivos números. Era este apenas um caso particular onde realizaria esta pesquisa de forma não-presencial. Contudo, outras oportunidades foram surgindo, desde meu orientador que disponibilizou alguns minutos com ambas as turmas que ele ministrava aula, para que eu explicasse aquela pesquisa e pedisse a opinião deles através do preenchimento deste quadro com os números dos rostos associados. Tal foi a comodidade de realizar esta pesquisa a partir do preenchimento de uma tabela, que mais de 50 das respostas foram obtidas de tal maneira.

Assim, com as respostas de 37 homens e 29 mulheres, pedindo que distribuíssem 18 das 22 expressões da Amanda dentro das 3 categorias (tédio, raiva e felicidade) em até 6 graus de intensidade. Seus resultados mostraram que não há muita diferença na forma com que homens e mulheres distribuem as expressões. Contudo, as mulheres aparentemente possuem uma maior concordância sobre o significado de cada expressão, enquanto os homens divergem mais nos significados com intensidades intermediárias. As expressões com maior concordância para ambos os gêneros foram implementadas no jogo como as novas expressões da personagem.

No decorrer da produção, observei cinco pessoas jogando. As perguntas realizadas pelos jogadores foram utilizadas para melhorar as instruções da tela de ajuda ou na clareza das missões. O desempenho alcançado nesta versão pelos jogadores devido a curva de aprendizagem mais suave era nítida. Os jogadores conseguiam aprender a jogar sem me questionar sobre o que era preciso fazer. Curioso também foi observar suas interações com o mecanismo de *feedback* Amanda, pois mesmo falhando sucessivamente ou demorando para resolver algum desafio, achavam graça das reações exageradas desta personagem. Isto parecia quebrar a seriedade de não conseguir resolver os problemas Matemáticos, incentivando os jogadores a se arriscarem mais.

Também recebi de duas pessoas vídeos de suas telas de jogo por meio de redes sociais. Analisando a forma como jogavam, os jogadores conseguiam avançar facilmente desde os níveis iniciais, parando de jogar somente nos níveis mais avançados. Isto me indicava que a aprendizagem do jogo durante o próprio jogo estava propícia aos novos jogadores.

Outro teste realizado envolveu 30 educadores. Neste caso antes de jogarem pedi que preenchessem um questionário sobre as atividades que um computador faz melhor do que um ser humano e o contrário. O objetivo deste questionário é identificar se os jogadores compreendem como o computador e o ser humano podem dividir tarefas de acordo com suas facilidades. Então os educadores jogaram o jogo em computadores com a tela sendo filmada. Após terminarem de jogar, pedia que escrevessem um relatório com críticas e sugestões sobre o jogo.





**Figura 39:** Educadores jogando (fonte própria).

Percebi neste teste que muitos dos educadores possuem uma visão errônea das diferenças entre os computadores e os seres humanos. No geral, aqueles que já tiveram experiências lecionando disciplinas ligadas com as tecnologias digitais compreendem melhor estas diferenças. Entre as sugestões dadas para o jogo, várias delas sugeriam a construção de um guia do professor para ajudá-los a aplicar o jogo em suas escolas. E outras 11 sugestões diziam que o jogo parece propício para ser aplicado no Ensino Fundamental I. Analisando os vídeos dos jogadores, percebi que aqueles com formação em Matemática ou psicologia, ou que tiveram videogames quando criança, se sobressaíam em igual. Outras formações e a ausência da experiência com jogos, proporcionava nos educadores comportamentos de hesitação ao experimentarem a falha dentro do jogo.

Também vale comentar, que o comportamento dos educadores com a personagem Amanda foi fascinante, eles simplesmente conversavam e reclamavam com ela enquanto jogavam, como se a personagem pudesse entender. Durante a aplicação, um sujeito que usava o laboratório de Informática e que não pertencia ao quadro de educadores, mostrou interesse pelo jogo e se colocou a jogar. Ele teve um bom desempenho, sua graduação em Física estava em andamento e ele teve muita experiência com videogames quando criança/adolescente.

O próximo teste foi movido de acordo com as sugestões dos educadores, de que o jogo poderia ser aplicado no Ensino Fundamental I. Assim procurei instituições de ensino e professores para a realização adequada dos procedimentos. E por uma questão de ética exigida pela própria instituição de ensino que me atendeu, eu somente poderia aplicar o jogo e os questionários avaliativos com seus alunos, se estes instrumentos e os procedimentos para a realização, estivessem

de acordo com algum comitê de ética e pesquisa. Dessa forma, com a devida aprovação do comitê de ética e pesquisa da Faculdade de Ciências da UNESP, pude realizar este teste em específico.

Assim com 27 crianças, apliquei um questionário anterior ao jogo contendo questões de Matemática e, outras 10 questões relacionadas a afinidade com jogos digitais e com o estudo curricular. Após jogarem individualmente, as crianças respondiam outro questionário, análogo nas questões Matemáticas, mas com 10 questões sobre a afinidade com este jogo digital em específico. Analisando o desempenho médio das crianças, sua nota passava de 1,6 para 2,2 nos questionários com nota máxima 10. Isto me sugeria que embora as crianças conseguissem jogar, este questionário não era apropriado para avaliar os conceitos de Matemática nos alunos do Ensino Fundamental I. Em compensação nas questões pessoais, os questionários mostraram que estas crianças eram consumidores de jogos digitais, gostavam de Matemática e tiveram uma afinidade com o jogo na versão que conheceram.



**Figura 40:** Crianças do Ensino Fundamental I jogando (fonte própria).

Outro teste realizado, foi graças a professora de uma ETEC do Estado de São Paulo (que compôs o grupo de 30 educadores a testar o jogo) me convidou para realizar testes com 5 turmas do Ensino Médio Técnico. Em sala de aula apliquei um questionário anterior ao jogo. Então eles eram encaminhados até o laboratório de Informática, e pela falta de computadores, os estudantes de cada turma precisavam jogar de 2 a 3 por computador. Então, após jogarem retornávamos para a sala onde aplicava outro questionário.



**Figura 41:** Crianças do Ensino Fundamental I jogando (fonte própria).

Ao final tinha em mãos 142 pares de questionários preenchidos. Analisando o desempenho médio dos alunos, sua nota passava de 4,75 para 5,20 nos questionários com nota máxima 10. Curioso desta análise é que o aumento comum do resultado ocorreu mesmo sem a participação ativa de todos no computador, visto que vários alunos ficavam apenas observando e dando sugestões a quem estava ativo no computador. Nas questões pessoais, os questionários mostraram que a maioria dos alunos consumia jogos digitais, sua preferência de disciplinas era relacionada ao curso técnico que realizavam e tiveram uma afinidade com o jogo na versão que conheceram.

Um resultado curioso deste teste no Ensino Médio Técnico foi com a turma do 2º Ano de Informática, onde três alunos após completarem quase metade dos desafios do jogo, decidem explorar a estrutura do *software*. Após compreenderem seu funcionamento, aproveitam a própria mecânica do jogo para produzir uma versão que critica o atual quadro de corrupção política vivenciado no Brasil. Substituindo o personagem do jogador pela imagem de um político em destaque nos meios de comunicação, os objetos agarráveis por blocos de dinheiro e a máquina (onde os itens agarráveis eram levados) foi substituída pela imagem de outro político em destaque. Analisando as disciplinas favoritas destes alunos, um preferia Geografia e os outros dois, disciplinas relacionadas à formação técnica em Informática. Ao final da atividade, pedi para fazer uma cópia da versão editada do jogo, a qual apresento somente em caráter ilustrativo abaixo.



**Figura 42:** Jogo editado pelos alunos (fonte própria).

**Finalização:** Esta versão foi considerada a definitiva, e concluída em julho de 2017. Trabalhei então na construção de versões similares em outros idiomas (inglês, espanhol, italiano). Os idiomas escolhidos de acordo com o número de projetos no *Scratch* que abordavam o tema Pensamento Computacional, dentro é claro, dos idiomas que eu conseguiria compreender o mínimo para criar estas versões. Também investi na atualização regular da fã-page do jogo, trazendo sugestões aos jogadores mais dedicados, como modos alternativos de jogo, que se baseavam bastante na ideia do *Scratch* permitir que usuários mesmo não cadastrados possam editar o código em uma versão particular do jogo. Este estímulo ao jogador experiente, abrir o código do jogo, aparece na tela final, onde a personagem Amanda, feliz com o trabalho do jogador, instruí sobre como acessar o interior do *software*. Um “ataque final” ao tema Pensamento Computacional, pois nesta ocasião via o jogo todo como um preparatório para editá-lo, assim se conseguisse provar que o jogador frente a todos os desafios estivesse apto a entender como o código do jogo funcionava, então conseguiria dizer que os desafios estimularam de fato o Pensamento Computacional do jogador. Assim, as sugestões para alterar o código do jogo apresentava 5 novos desafios sugeridos aos jogadores.

- 1) **Resto 1:** abrindo o código do jogo modifique no ator “controle” a variável resto para 1. Isso fará com que seu personagem agarre apenas aqueles números que divididos pelo “Divisor” tenham resto 1. Descubra quais estágios se tornaram impossíveis de serem solucionados e quais ficaram mais simples com essa modificação.
- 2) **Mistério:** abrindo o código do jogo modifique no ator “controle” a variável incógnito para 1. Agora todos os números são indistinguíveis visualmente. Veja se algum estágio é impossível.
- 3) **Altere a velocidade:** abrindo o código do jogo modifique no ator “controle” a variável passo para qualquer valor e você verá as diferenças de movimentação.

**4) Altere o tamanho:** abrindo o código do jogo, vá para o ator “o+<”, então vá para os botões da categoria Aparência (botões roxos), puxe para a área de trabalho do jogo o botão “mudar de tamanho [50]%”. Altere esse valor e clique para ser um gigante ou um baixinho.

**5) Modo fantasma:** abrindo o código do jogo, vá para o ator “o+<”, então vá para os botões da categoria Aparência (botões roxos), puxe para a área de trabalho do jogo o botão “mudar efeito de [cor] para [0]”. Altere o [cor] por [fantasma] e o valor [0] por [100], isso deixará seu personagem invisível, valores próximos de [100] o deixarão apenas transparente.

Também pensando em propor desafios dentro do próprio jogo, onde se podem explorar conceitos que estão além dos seus desafios, ou mesmo estratégias degeneradas (estratégias que não são proibidas, porém fogem ao modo esperado de se jogar, como no caso da moça que utilizou o efeito de retorno automático da gravação para pular os números 0's indesejados). No caso, são listados na fã-page os seguintes 3 modos alternativos de jogo como sugestão os jogadores mais interessados:

**1) Speedrun:** tente terminar os 42 estágios da maneira mais rápida possível (meu recorde é 16 minutos), pense em como posso resolver certos estágios tão rapidamente, explore algumas estratégias degeneradas se for preciso.

**2) Multiplicando:** pense em uma multiplicação entre inteiros, agora usando as regras “Divisor” e “Repetir” descubra como calcular seu produto. Exemplo abaixo:  $25 * 25 = 625$ .

**3) Às avessas:** nos estágios 26 até o 30. Acredite que as regras mudaram, que os números agora variam de -29 até 0 e a tarefa agora é levar os números do menor para o maior. Resolva...

Para evitar surpresas ou mesmo preparar os jogadores das próprias condições conhecidas do jogo, pensei também na listagem das estratégias degeneradas conhecidas e mencionadas acima. Assim, o jogo possui 4 *bugs* conhecidos que não estragam os desafios propostos pelo jogo (velocidade; jeito especial de soltar; inserção parcial; teletransporte). Explorar tais estratégias ou outras viabilizadas com modificações no game design é algo bastante complexo e que exige um domínio completo sobre a dinâmica e as mecânicas do jogo. Sendo uma atividade para aqueles jogadores que buscam deixar o jogo mais desafiador do que para quem quer um modo de vencer os desafios sem muito esforço mental. A seguir listamos as 4 conhecidas até o momento:

**1) Velocidade:** assim que você completa uma missão a tela do jogo fica por um segundo com o fundo cinza, se nessa ocasião você acessar a tela de ajuda (apertar b), ao retornar para o jogo sua velocidade estará dobrada nesta instância do jogo (ou seja, até falhar ou passar de nível).

**2) Jeito especial de soltar:** sempre que apertamos Z o personagem solta o objeto, porém os direcionais do teclado (setinhas) são desabilitadas, assim antes que voltemos a mover o

personagem, ocorre a verificação se há espaço na Mochila para bolinhos ou se o número que tocamos é divisível por "Q", fazendo o personagem voltar a segurá-los. Contudo, se após soltar o Z nos movemos e voltamos a segurar o Z, nos afastamos um pouco mais do objeto, repetindo este processo, nos distanciamos até que não mais toquemos no objeto e com isto o soltemos.

**3) Inserção parcial:** quando carregamos números visualmente distinguíveis que são representados com figuras de larguras bem diferentes, conseguimos tocar alguns destes números na máquina sem que os mais finos também toquem, inserindo assim apenas parte do conjunto carregado.

**4) Teletransporte:** quando gravamos os movimentos do personagem, ao encerrar somos teletransportados para a posição inicial da gravação. Com isto, podemos criar uma gravação (sem intenção de executá-la posteriormente), apenas para nos movermos até um ponto da tela, agarrando um objeto, encerrando a gravação para que sejamos teletransportados até a origem.

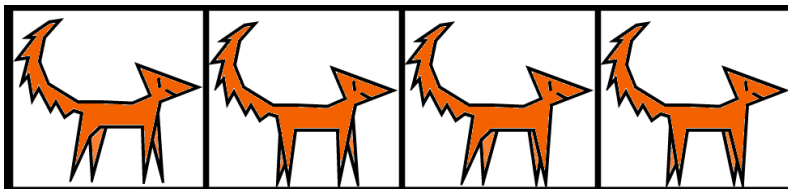
Analisando o jogo todo em sua versão final, considerei os seguintes aspectos positivos e negativos:

- a) O envolvimento emocional gerado pela interação da Amanda com o jogador deixava o jogo leve, agradável e gerava uma história implícita entre os jogadores na tentativa de agradar a personagem;
- b) O aumento crescente na dificuldade dos desafios ensinava os jogadores a jogar;
- c) As dinâmicas deste jogo são simples a ponto de crianças entre 10 e 11 anos conseguirem jogá-lo;
- d) Os problemas Matemáticos deste jogo são tão profundos que alunos do Ensino Médio se entretêm para resolvê-los;
- e) Mesmo os educadores formados em Matemática admiram a maneira com que os problemas do jogo exploram uma Matemática básica em um contexto simples porém incomum;
- f) O tempo necessário para concluir os 42 níveis do jogo dificultam que os jogadores explorem dinâmicas diferentes daqueles que envolvem alterar o critério do *Divisor*;
- g) Alguns jogadores se sentem imensamente pressionados pela figura humana da Amanda como um mecanismo de *feedback*;

## Saúvas in Scratch

A ideia para esta versão surgiu como um susto. Estava em meio a uma conversa corriqueira no mês de dezembro de 2017, e o assunto se voltou para aspectos culturais do Brasil, e como isto é valorizado no exterior. Assim pensei, porque tentamos fazer produtos com características estrangeiras? Não sei porque, porém desse momento em diante adotei um lema, tentar mesclar traços da cultura brasileira nos jogos.

Então peguei logo o **Handles in Scratch 2.0**, que a tarefa de encurtá-lo estava a meses na minha lista de afazeres. E comecei a investir na transformação do visual do jogo para o contexto do cerrado com o qual tenho bastante familiaridade. Tentei de cara colocar o protagonista como um lobo-guará desenhado de forma vetorial no próprio *Scratch*. Não havia problemas para o animal movimentar-se horizontalmente, porém o movimento vertical ficava muito forçado.



**Figura 43:** Sprites do lobo-guará em movimento horizontal (fonte própria).

Substituindo o lobo-guará para uma outra figura bastante comum no cerrado brasileiro, cheguei na formiga saúva. Como estamos acostumados a ver formigas andando pelas paredes, sua movimentação vertical não causaria estranhamento. Assim trabalhei na adaptação dos conceitos e dos elementos do jogo para ações e comandos que uma formiga receberia. O título **Saúvas in Scratch**, faz uma alusão ao título anterior porém tirando o foco do jogador quem manipula os problemas, e colocando-o nas “mãos” de uma formiga saúva colheiteira.

**Pré-produção:** Os gráficos utilizados nesta versão se basearam em uma antiga apresentação que desenvolvi para instruir crianças antes de irmos visitar o cerrado da cidade de São Carlos – SP. Assim, tomei o modelo da formiga saúva e refiz suas linhas e cores em formato vetorial dentro do próprio *Scratch*. Assumi a formiga rainha para ocupar o local da personagem Amanda. E atribui para a última tela um parabéns ao jogador e a mensagem de que com seu trabalho (e de outras formigas) foi possível reunir sementes para todo o inverno. Estes e outros aspectos visuais necessários para construção desta versão estão descritos a seguir:

- a) Idem ao anterior (*O jogo será bidimensional com visão de perfil*);
- b) Os objetos agarráveis serão sementes ovais de diferentes tonalidades e com tamanhos variados;
- c) Por vezes terá um número visível no interior de uma semente;
- d) O protagonista será uma formiga do tipo saúva colheiteira;
- e) Todas as telas de fundo dos níveis serão levemente amareladas de baixo para cima;
- f) A parte do jogo que recebe os objetos carregados pelo personagem (chamada de formigueiro) terá a aparência de um formigueiro marrom;
- g) Haverá uma barra preta do lado esquerdo da tela com informações ativas sobre variando seu conteúdo de acordo com o nível atual do jogador;

- h) A tela final do jogo possui uma mensagem de parabéns ao jogador;
- i) A personagem que diz as missões do jogo se chama Rainha, e é representada por uma grande cabeça de formiga com uma coroa amarela;
- j) Na tela de ajuda haverá uma breve explicação sobre o significado das palavras e imagens utilizadas no decorrer dos níveis, junto com alguns comandos essenciais ao jogador;
- k) A personagem Rainha terá expressões de raiva que aumentam a grossura das sobrancelhas e sua cor passa de um vermelho para um vermelho bem escuro.

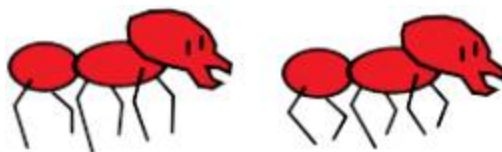
Os aspectos lógicos desta versão não foram tão triviais quanto imaginava. Decidi resgatar uma mecânica do **Handles**, para atribuir uma penalidade de movimento em relação a quantidade de objetos que o personagem carrega. Pois como descartei a mecânica de gerar rotinas de repetição, pude trabalhar com alterações no movimento do personagem sem riscos de interferência. Outro aspecto para resgatar esta mecânica, é no fato dela ser passiva, ou seja, não precisar instruir o jogador sobre como usá-la, pois ela automaticamente afeta o personagem querendo o jogador ou não. Além de ser uma mecânica coerente para o processo de uma formiga carregar sementes.

Outro aspecto mais implícito que inseri, e de difícil percepção foi sua forma de movimento. Como ela moveria as patinhas para cima e para baixo, para um lado e para o outro, a orientação dela precisaria acompanhar estes movimentos. Não podia simplesmente deixá-la subir com as patas apontando para a esquerda e descer com as patas apontando para a direita. Para consertar isto, tratei de cada direção como um par de imagens diferente para a formiga. Porém isto seria um trabalho cruel de se construir no *Scratch*. Eu precisaria criar um verificador que analisa a imagem utilizada pela formiga no estado atual, para então decidir qual das duas imagens dela eu utilizaria em seguida. Erros nesta construção fariam a formiga se movimentar de modo incoerente a sua imagem. Como subindo uma parede estando com as patas apontando para o chão.

A ideia para o truque que utilizei surgiu do tédio, enquanto observava no vidro do carro o movimento das patinhas de formigas visto de baixo. Mesmo prestando atenção, tentando acompanhar, não era possível. Era como se as patinhas ficassem em movimentos consecutivos tão rápidos que não dava para definir se estavam se movendo de forma ordenada ou ao acaso. Então, surgiu a ideia. Em vez de construir um verificador para a imagem atual usada pela formiga, decidi criar um gerador aleatório entre dois números referentes ao número das imagens de cada cena. Assim, se a formiga tivesse que andar na horizontal para a direita, o *software* ficaria sorteando entre uma imagem e outra da formiga no sentido horizontal para a direita. O segredo aqui foi perceber que o jogador realmente não perceberia que os movimentos de suas patas não estão sendo

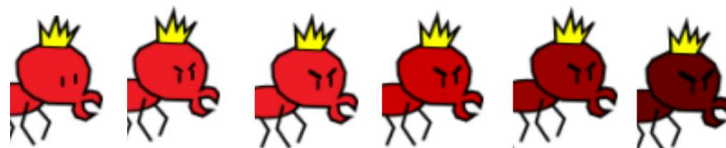


sequenciais, e sim ao acaso. Pois assim como na vida real, quando olhava para uma formiga via apenas patinhas rápidas se mexendo sem que conseguisse definir uma sequência clara.



**Figura 44:** Sprites da movimentação da formiga (fonte própria).

Similar as reações da Amanda, decidi investir em reações para o mecanismo de *feedback* Rainha, que regula esta versão do jogo. Porém, apenas atribui expressões as situações de falha, visto que o reinício do nível em caso de falha por muitas vezes poderia ser confundido com o início de um novo nível. O indicativo da Rainha ficando mais irritada teria como finalidade mostrar ao jogador que ele não fez o que a missão pedia. A edição de sua imagem foi feita no próprio *Scratch 2.0*, a partir de alterações no formato vetorial da formiga saúva que é a protagonista do jogo.



**Figura 45:** Reações da rainha mediante sucessivas falhas (fonte própria).

De forma geral, os aspectos lógicos desta versão foram:

- a) Idem ao anterior (*Os extremos da tela estarão conectados com seus lados opostos, fazendo com que o personagem quando avançar por um extremo, apareça no outro*);
- b) Idem ao anterior (*Enquanto o comando insert estiver pressionado, o personagem não se moverá pelas teclas dos direcionais*);
- c) O jogador possui um comando para inserir (chamado de insert) no formigueiro as sementes que estejam tocando no formigueiro;
- d) O jogador possui um comando para alterar a condição que faz o personagem agarrar sementes, isto ocorre segurando o comando insert e pressionando as teclas de direcionais para direita ou para esquerda, este valor será chamado de Divisor);
- e) O jogador controla o personagem através dos respectivos direcionais do teclado;
- f) O jogo possui 10 níveis;
- g) No início ou reinício dos níveis, a personagem Rainha dirá por 2 segundos a missão, esta mensagem some com qualquer tecla que o jogador pressionar após este tempo;
- h) A variável *Divisor* pertence aos Naturais maiores que 0;

- i) Nas missões, se todas as sementes corretas forem inseridas no formigueiro, o jogador avança ao próximo nível, mas se ao menos uma errada for inserida o nível atual se reinicia imediatamente;
- j) Todos os níveis terão sementes com valores entre 1 e 9 escolhidos aleatoriamente ou predeterminados e que também podem ser indistinguíveis visualmente;
- k) O personagem recebe uma penalidade de movimento diretamente proporcional com a quantidade de sementes que carrega.
- l) Todos os níveis possuem sementes dispostas na tela de forma aleatória ou predefinida, um formigueiro no canto inferior direito, o personagem do jogador localizado a sua esquerda;
- m) Alterar a variável *Divisor* somente é possível a partir do nível 6 do jogo;
- n) A variável *Mochila* registra o total de sementes que existem no nível e a quantidade que o personagem esta carregando;
- o) Na barra preta de informações, está registrado os valores atuais do nível, total de níveis, *Divisor*, *Mochila*.
- p) Do nível 3 ao nível 8 o valor numérico das sementes é visível ao jogador;
- q) O mecanismo de *feedback* Rainha reagirá aos reinícios sucessivos de um nível pelo jogador da seguinte maneira, ela mudará sua imagem para expressões de raiva em até 5 intensidades;
- r) O mecanismo de *feedback* Rainha retorna a sua expressão inicial assim que o jogador avança para um novo nível.

**Produção:** Mantive o foco na construção de poucos níveis, mas que trabalhassem a jogabilidade principal de **Handles in Scratch 2.0**, resolver problemas Matemáticos arrastando e soltando números. Assim, este jogo trabalha durante 10 níveis os mesmos conteúdos explorados em 26 níveis da sua versão predecessora. Durante sua construção foquei na inclusão de uma nova habilidade amparada de um ou dois níveis de desafios triviais para o jogador se acostumar a usá-la. Também acrescentei instruções bastante descritivas no uso das habilidades para facilitar seu entendimento.

Outra variação importante nesta versão, é a ausência dos objetos com valor 0. Pois nas versões **Handles**, **Handles 2.0** e **Handles in Scratch 2.0**, soltar os objetos com valor 0 utilizando o *Divisor* exigia que o *Divisor* fosse igual a 0. Uma estratégia que por muitos jogadores não era natural ou compreensível. Vários pareciam fazer isto por perceberem que no jogo funcionava, mas de fato não compreendiam o motivo do *Divisor* 0 soltar todos os números. Assim, também por uma questão de lógica com o contexto de buscar sementes, as sementes com tempo de germinação zero dias, seriam sementes que já germinaram. Dessa forma, eliminando os valores 0, e restringindo a variação do *Divisor* entre 1 e 9, era possível soltar qualquer número natural entre 1 e 2519.

De forma geral os níveis foram configurados da seguinte maneira:

- 1) Uma semente ao centro da tela. A missão do jogador é inserir a semente no formigueiro.
- 2) Cinco sementes espalhadas na tela. A missão do jogador é inserir as sementes no formigueiro.
- 3) Quatro sementes com valores entre 1 e 3, e quatro sementes com valores entre 4 e 9 espalhadas na tela sem sobreposição. A missão do jogador é inserir as sementes com valores abaixo de 4 no formigueiro.
- 4) Seis sementes com valores 9, e treze sementes com valores abaixo de 9 bloqueando algumas passagens. A missão do jogador é inserir as sementes com valores 9 no formigueiro.
- 5) Uma semente com valor 3 na mesma posição inicial do personagem e uma semente com valor aleatório no centro. A missão do jogador é levar a semente do centro da tela no formigueiro.
- 6) Seis sementes com valores 3, e quatorze sementes com valores abaixo de diferentes de 3 bloqueando todas as passagens. A missão do jogador é inserir as sementes com valores 3 no formigueiro.
- 7) Doze sementes com valores pares e doze sementes com valores ímpares espalhados pela tela sem sobreposição. A missão do jogador é inserir as sementes com valores pares no formigueiro.
- 8) Doze sementes com valores pares e doze sementes com valores ímpares espalhados pela tela sem sobreposição. A missão do jogador é inserir as sementes com valores ímpares no formigueiro.
- 9) Três sementes com valores 3 e sete sementes com valores diferentes de 3 espalhados pela tela. A missão do jogador é inserir as sementes com valores 3 no formigueiro.
- 10) Três sementes com valores 1 e sete sementes com valores diferentes de 1 espalhados pela tela. A missão do jogador é inserir as sementes com valores 1 no formigueiro.

**Testes:** A fase de testes deste jogo ainda está em andamento, não havendo muito o que possa dizer sobre isto. Até agora temos três pessoas envolvidas no teste de funcionalidade deste protótipo, e dez respostas sobre o jogo recebidas por redes sociais de jogadores convidados a jogar. Pelo que foi observado até o momento, o critério do Divisor continua gerando desentendimentos, ainda que apareça instruções de uso à esquerda da tela de jogo.

## 2. Questionário anterior ao jogo

Nome: \_\_\_\_\_ Ano: \_\_\_\_\_ Data de nascimento: \_\_\_\_/\_\_\_\_/\_\_\_\_

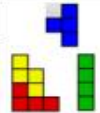










1) Marque com X nos números abaixo que podem ser divididos por [ 4 ] sem deixar resto:	?	2) Marque com X nos números abaixo que podem ser divididos por [ 7 ] sem deixar resto:
[ 2 ] [ 4 ] [ 8 ] [ 16 ] [ 18 ]		[ 0 ] [ 4 ] [ 14 ] [ 15 ] [ 27 ]

3) Marque com X nos números abaixo que podem ser divididos por [ 2 ] sem deixar resto:	?	4) Marque com X nos números abaixo que podem ser divididos por [ 1 ] sem deixar resto:
[ 0 ] [ 13 ] [ 26 ] [ 39 ] [ 52 ]		[ 0 ] [ 1 ] [ 7 ] [ 15 ] [ 23 ]












5) Marque com X nos números abaixo que podem ser divididos por [ 0 ] sem deixar resto:	??	6) Marque com X nos números abaixo que podem ser divididos por [ 72 ] sem deixar resto:
[ 0 ] [ 1 ] [ 5 ] [ 10 ] [ 12 ]		[ 0 ] [ 41 ] [ 17 ] [ 9 ] [ 26 ]

7) Marque com X a resposta correta. Agora são 8 da manhã, daqui a 36 horas, que horas serão?		8) Marque com X a resposta correta. Agora são 13 da tarde, 19 horas atrás, que horas eram?
[ 20:00 ] [ 10:00 ] [ 21:00 ] [ 9:00 ] [ 8:00 ]		[ 20:00 ] [ 6:00 ] [ 18:00 ] [ 12:00 ] [ 0:00 ]

9) Um relógio que adianta 5 minutos a cada hora, agora marca 8 da manhã, que horas marcará após 12 horas?		10) Um relógio que atrasa 5 minutos a cada hora, agora marca 16 da tarde, que horas marcava a 12 horas atrás?
[ 19:00 ] [ 19:30 ] [ 20:00 ] [ 20:30 ] [ 21:00 ]		[ 3:00 ] [ 3:30 ] [ 4:00 ] [ 4:30 ] [ 5:00 ]

11) Gosta de jogos de PC, videogame ou celular?		12) Gosta de conversar com amigos sobre jogos?
    		    

13) Gosta de ir para a escola ou estudar em casa?		14) Sua família deixa você jogar em casa?
    		    

15) Gosta de jogos que te desafiem?		16) Gosta inventar novas estratégias nos jogos?
    		    

17) Jogos digitais favoritos? (podem ser jogos de computador, videogame, tablet ou celular)
18) Por que gosta destes jogos?
19) Disciplina escolar favorita?
20) Algum comentário?


### 3. Questionário posterior ao jogo

Nome: \_\_\_\_\_ Tarefa máxima alcançada: \_\_\_\_\_ Tempo: \_\_\_\_\_ minutos

21) Marque com X nos números abaixo que podem ser divididos por [ 4 ] sem deixar resto:	?	22) Marque com X nos números abaixo que podem ser divididos por [ 7 ] sem deixar resto:
[ 2 ] [ 4 ] [ 8 ] [ 16 ] [ 18 ]		[ 0 ] [ 4 ] [ 14 ] [ 15 ] [ 27 ]












23) Marque com X nos números abaixo que podem ser divididos por [ 2 ] sem deixar resto:	?	24) Marque com X nos números abaixo que podem ser divididos por [ 1 ] sem deixar resto:
[ 0 ] [ 13 ] [ 26 ] [ 39 ] [ 52 ]		[ 0 ] [ 1 ] [ 7 ] [ 15 ] [ 23 ]

25) Marque com X nos números abaixo que podem ser divididos por [ 0 ] sem deixar resto:	??	26) Marque com X nos números abaixo que podem ser divididos por [ 72 ] sem deixar resto:
[ 0 ] [ 1 ] [ 5 ] [ 10 ] [ 12 ]		[ 0 ] [ 41 ] [ 17 ] [ 9 ] [ 26 ]

27) Marque com X a resposta correta. Agora são 8 da manhã, daqui a 36 horas, que horas serão?		28) Marque com X a resposta correta. Agora são 13 da tarde, 19 horas atrás, que horas eram?
[ 20:00 ] [ 10:00 ] [ 21:00 ] [ 9:00 ] [ 8:00 ]		[ 20:00 ] [ 6:00 ] [ 18:00 ] [ 12:00 ] [ 0:00 ]

29) Um relógio que adianta 5 minutos a cada hora, agora marca 8 da manhã, que horas marcará após 12 horas?		30) Um relógio que atrasa 5 minutos a cada hora, agora marca 16 da tarde, que horas marcava a 12 horas atrás?
[ 19:00 ] [ 19:30 ] [ 20:00 ] [ 20:30 ] [ 21:00 ]		[ 3:00 ] [ 3:30 ] [ 4:00 ] [ 4:30 ] [ 5:00 ]

31) Você achou este jogo divertido?		32) Você gostou do jeito de pegar os números??
    		    

33) Você acha que foi bem no jogo?		34) Foi fácil aprender a jogar?
    		    

35) A "Amanda" te incentivou a jogar?		36) Você gostou de deixar a "Amanda" feliz?
    		    

37) O que achou mais legal no jogo?
38) O que achou mais difícil no jogo?
39) O que você mudaria no jogo?
40) Algum comentário?

## 4. Artigo não-publicado / releitura do método de Fellows (1996)

### A Matemática no coração dos jogos digitais:

uma releitura do trabalho de Michael R. Fellows

*Marcos Henrique de Paula Dias da Silva*

**Resumo:** Reconhecendo a importância de se desenvolver nos cidadãos uma apreciação da Matemática tanto por seus aspectos estéticos como recreativos, este trabalho faz releitura de um método sistemático para construir elementos-chave de jogos digitais a partir da combinação de ferramentas e problemas Matemáticos proposto por Michael R. Fellows em 1996. Em virtude do método ser discutido de forma bastante conceitual, este artigo busca retratar cada exemplo e procedimento original, a partir de uma interpretação prática utilizando-se de um mesmo jogo digital de ampla divulgação para construir o esquema proposto pelo autor, levando desde a discretização do jogo Super Bomberman à proposta de um novo jogo que combine sua estrutura reduzida a um problema NP.

**Palavras-chave:** Ensino. Matemática Discreta. Jogos Digitais.

### Introdução

Michael R. Fellows (1996) em seu artigo *The Heart of Puzzling: Mathematics and Computer Games*, descreve como as ideias Matemáticas modernas podem ser exploradas não apenas no aumento do conteúdo dos jogos para ensino de Matemática, mas também incorporando enigmas inovadores e estruturas de interação sem precedentes aos jogos digitais de uso geral. O autor mescla sua experiência em Matemática, Ciências da Computação e *Game Design* para desenvolver um método sistemático capaz de gerar milhares de jogos digitais com perfis de jogabilidade inéditos. Sua discussão é organizada de forma conceitual utilizando diversos termos próprios explicitados no texto, exigindo do leitor uma elevada capacidade de abstração, assim como uma base em Matemática e Teoria da Complexidade Computacional.

A princípio pode parecer ilógico do ponto de vista do Ensino de Matemática, desempenhar um enorme esforço para incorporar em jogos digitais de uso geral uma Matemática que não é visível na forma de conteúdo. Porém, James P. Gee (2005) ataca o seguinte jargão, *a única coisa que se aprende ao jogar um jogo digital, é jogar esse jogo digital*, defendendo que apesar da ironia, um jogo ensinar como ser jogado é um excelente princípio de aprendizagem, mas por vezes ignorado no ambiente escolar, onde aprender os fatos substituem o *brincar com os fatos*. Ou seja, a prioridade do conhecimento fica restrita à reprodução escrita das verdades, e não à sua investigação, o que Howard Gardner (1991) já provou como um método defeituoso, pois alunos quando ensinados desta forma não conseguem aplicar conhecimento para resolver problemas ou para entender os fundamentos conceituais de seu campo de aprendizagem.

Gee (2005) explicita 16 princípios de aprendizagem considerados os mais relevantes encontrados em bons jogos digitais. Dentre eles, o décimo terceiro princípio traduzível como *Explorar, pensar lateralmente, repensar os objetivos*, refere-se ao incentivo dado para os jogadores explorarem detalhes em vez de irem rápido ao prêmio, pensando não apenas de forma linear, usando nessa exploração o pensamento lateral para repensarem seus próprios objetivos de tempos em tempos. Corroborando com a ideia de Fellows (1996) sobre os jogos digitais serem rígidos, pois quando o jogador descobre a receita para derrotar um obstáculo,

este se torna trivial, ou seja, existe uma fórmula confiável e replicável para resolver um problema específico. Assim, se a conclusão do jogo fosse um teorema, seu percurso seria composto por diversos lemas. Pessoas com muita experiência em jogos digitais, aprenderam a fazer suposições razoáveis e se comportar (embora inconscientemente) como pesquisadores Matemáticos.

Reconhecendo a importância de se desenvolver nos cidadãos uma apreciação da Matemática tanto por seus aspectos estéticos como recreativos (NCTM, 2000), este trabalho apresenta uma releitura do método proposto por Fellows (1996), onde vários conceitos de sua discussão original são reinterpretados à luz de um jogo digital muito conhecido e diferente daqueles já tratados no artigo original. Esta abordagem surgiu durante a construção de jogos digitais para outra pesquisa, como forma de exemplificar todos os elementos estruturais deste método a partir de um único contexto.

### **Super Bomberman**

Escolhemos para exemplificar este método, o jogo digital Super Bomberman, lançado em 1993. Este é uma versão de Bomberman, uma série jogos de estratégia baseados em labirintos bidimensionais com visão de topo, desenvolvida por *Hudson Soft* e com a primeira versão publicada em 1983. Sua escolha ocorreu por conveniência, dado que a própria experiência do pesquisador com este jogo proporcionava a partir dele sua principal referência para entender e criar exemplos práticos dos vários elementos e processos presentes no método proposto por Fellows (1996). Podemos considerar este um jogo digital abrangente o bastante para atingir um grande público de leitores, dado que até 2009 existiam cerca de 70 versões desta franquia produzidas das quais, quase 10 milhões de cópias já tinham sido vendidas (Retro Gamer Magazine, 2009).

De forma simples, este jogo apresenta gráficos bidimensionais com uma leve inclinação para dar aspecto de altura aos personagens, objetos e cenário. O jogador controla um robzinho branco e azul que se movimenta na horizontal e na vertical de acordo com comandos diretos do *joystick*. Cada cenário é composto por um tabuleiro de 13x11. Considerando a casa localizada no canto inferior esquerdo como a posição (1, 1), temos que as posições (2n, 2m), com n, m números Naturais, obrigatoriamente correspondem a casas inocupáveis (bloquinhos indestrutíveis), as demais posições podem estar inocupáveis, livres ou obstruídas momentaneamente (bloquinhos destrutíveis).

O modo de desobstruir caminhos deste jogo é a partir das bombas, objetos que quando colocados no campo, obstruem o trajeto até que explodam, eliminando a si mesma, os objetos, personagens e bloquinhos destrutíveis em linhas retas livres ao seu redor dentro de uma distância equivalente a sua potência atual. Inimigos também contam como obstáculos ao personagem, capazes de destruí-lo tanto pelo contato quanto usando bombas. Existem vários aditivos ao personagem que estão escondidos em bloquinhos destrutíveis, estes aditivos podem dar mais velocidade, aumentar a quantidade de bombas para o jogador implantar ao mesmo tempo, seu campo de destruição, entre outros efeitos positivos ou negativos. Cada nível possui um portal para o próximo, que só pode ser utilizado quando todos os inimigos forem eliminados, e se o portal for atingido por uma explosão, gera novos inimigos.

Os níveis possuem também um tempo limite para sua conclusão, caso expire o personagem perde uma vida. Caso o personagem seja atingido por uma explosão ou por um inimigo, também perde uma vida. Ao fim das vidas, o jogo termina. Ganhar muitos pontos (conquistando aditivos e derrotando inimigos) pode conferir ao jogador vidas extras. Além disso, há vários elementos particulares dos cenários como portais, armadilhas, túneis entre outros, além de inimigos com poderes muito específicos (capazes de se esconder no mapa por exemplo) e os *chefões*, que possuem uma interação totalmente diferente do restante do jogo.



Figura 1: Cenas do jogo Super Bomberman (fonte própria)

### Elementos centrais

Fellows (1996) inicia seu método apresentando os *puzzles* e as *estruturas de ação* como elementos centrais a todos os jogos digitais. Ambos são inerentemente questões de natureza Matemática, assim, há muitas maneiras sistemáticas de tomar várias linhas da Matemática para criá-los de forma inovadora. Então, para facilitar a compreensão do leitor sobre o que são estes elementos centrais e como podem ser vistos a partir de sua natureza Matemática, tratamos os seguintes exemplos referentes ao jogo Super Bomberman (1993):

Um *puzzle* presente no jogo, inicia quando todos os inimigos de um cenário são destruídos, mas o portal ainda não foi localizado. Assim, o jogador precisa destruir os bloquinhos restantes para descobrir onde está o portal antes que seu tempo termine. Dessa forma, temos pelo menos duas formas de atacar este problema. Podemos tratá-lo como um problema de rotas, onde o personagem deve *visitar* todos os bloquinhos em uma trajetória para destruí-los. Outro modo de atacar este problema é considerando-o um caso de probabilidade, dado que a posição do portal é aleatória, sendo mais provável encontrá-lo destruindo primeiro os locais com maior concentração de bloquinhos.

Uma *estrutura de ação* do jogo, é plotar bombas no tabuleiro 13x11. As bombas podem ser interpretadas como funções Matemáticas que dado sua posição atual  $(X, Y)$ , sendo  $X$  e  $Y$  números Naturais, sendo  $j$  o alcance máximo da explosão (em casas do tabuleiro), no ato da explosão são atingidas ao mesmo tempo todas as posições disponíveis de  $(X+i, Y)$  e  $(X, Y+i)$ , tal que  $-j-1 < i < j+1$ . Assim, se o bloquinho alvo



ocupar  $(X+i, Y)$  ou  $(X, Y+i)$ , ele será destruído. Por outro lado, no que diz respeito a destruir inimigos, eles podem variar de posição de formas bem particulares, neste caso, acertá-los significa identificar onde ambas as funções (explosão e movimento) se cruzam. Uma alternativa para facilitar este processo, é reduzir o espaço de possibilidades de movimento do inimigo, tornando seu comportamento mais restrito.

O método também é baseado no trabalho de referência *Computers and Intractability: A Guide to the Theory of NP-completeness* de Garey e Johnson (1979). Este livro cataloga os dramas computacionais das mais variadas situações e resume suas respectivas essências Matemáticas. Nestes catálogos, temos os problemas da classe NP, que serão a matéria-prima deste método. Estes são problemas cujo tempo de complexidade da função para resolvê-los é Não-Polinomial (daí o termo NP). Embora não seja intenção neste trabalho aprofundarmos a ideia de problemas NP, para melhor compreendermos o seu efeito no método proposto por Fellows (1996), apresentamos três situações dentro do Super Bomberman, com problemas de complexidade polinomial e não polinomial.

Em um cenário aberto, para o personagem se mover horizontalmente de um lado ao outro do tabuleiro com  $n$  casas na horizontal, ele levará  $n-1$  vezes o tempo de movimentar-se de uma casa para a outra, dado que ele já inicia em uma casa e precisará realizar  $n-1$  movimentos, assim se forem acrescentados ao jogo mais  $x$  casas na horizontal, o tempo necessário aumentaria somente mais  $x$  vezes o movimento de passar uma casa para a outra. Contudo, se o objetivo fosse percorrer todas as casas de um tabuleiro quadrado com  $n^2$  casas, o jogador levaria aproximadamente  $n^2$  vezes o tempo de movimentar-se de uma casa para a outra, assim se aumentássemos seu tamanho, fazendo deste, um tabuleiro com  $(n+x)^2$  casas, o tempo necessário para movimentar-se por elas aumentaria cerca de  $2nx+x^2$  vezes o tempo de movimentar-se de uma casa para a outra. Entretanto, se para destruir todos os bloquinhos restantes (como já mencionamos na explicação sobre *puzzle* neste jogo) determinamos uma rota que leva  $P$  segundos para isso, ao acrescentar um bloquinho a mais, não podemos dizer que a mesma rota continuará válida apenas acrescentando um número determinado de processos, pois o problema de encontrar a menor rota para destruímos todos os bloquinhos envolve uma complexidade de ordem Não Polinomial (embora provar que este problema pertença aos problemas NP seja uma tarefa bem mais difícil e não intuitiva, sua demonstração foge ao escopo deste artigo).

## O Método

Seu método constitui de duas etapas inter-relacionadas e apoiadas em conceitos próprios de Fellows (1996). A primeira delas é identificar uma *manivela para manipulação de informações* (nos referiremos a ela somente por *manivela* no restante do artigo). Podemos entendê-la mascarada em algum contexto como uma caixa de ferramentas disponíveis para tratar um dado problema. Caso Super Bomberman fosse um jogo de corrida onde precisamos chegar em um determinado local antes dos outros personagens, uma *manivela* esperada envolveria alterar a velocidade de deslocamento próprio e uma forma de usá-la seria passando pelas trajetórias lineares com o máximo da velocidade. Contudo com a *manivela* usual, um problema de corrida

possivelmente seria resolvido pelos jogadores mais experientes implantando bombas em posições que eliminem ou atrasem seus oponentes.

Embora conheçamos sua *manivela* mascarada dentro do contexto de bombas, para reaproveitá-la é importante reduzi-la ao mais abstrato cenário, de preferência, um problema de Matemática simples com suas instâncias tratáveis a partir desta *manivela*. No caso, se o desafio do jogo fosse destruir um bloquinho em específico na tela sem a presença de inimigos, uma *manivela* para tratá-lo seria a construção de trajetórias que ligam a posição inicial até o alvo. Assim, no contexto original do Super Bomberman, uma forma para entendermos a *manivela* deste jogo, utiliza alguns conceitos da teoria dos grafos, os quais serão enunciados de forma simplória somente para melhor ilustrar a explicação: vértices são pontos; arestas são conexões entre dois vértices; faces são regiões fechadas por arestas.

Analisando o cenário do jogo, as regiões com espaços indestrutíveis serão consideradas vértices pretos. Todo vértice preto adjacente a outro vértice preto, será conectado por uma aresta preta. Vértices e arestas pretas são indestrutíveis. Isto corresponde tanto aos bloquinhos indestrutíveis, como a fronteira externa do cenário, a qual não pode ser destruída, logo será tratada como bloquinho indestrutível também. Já os bloquinhos que podem ser destruídos, serão considerados vértices rosas. Todo vértice rosa adjacente a outro vértice preto ou rosa, será conectado por uma aresta rosa. O personagem protagonista será considerado um vértice verde, e os vértices pretos e rosas adjacentes a ele, serão conectados com arestas verdes. Toda face fechada por, pelo menos, quatro arestas é uma região ocupável por um novo vértice. Pela própria distribuição obrigatória dos bloquinhos indestrutíveis, é impossível que exista algum vértice rosa ou verde que não esteja conectado por nenhuma aresta a um outro vértice preto. Assim, sempre que um vértice é removido, suas arestas são eliminadas, e unindo novas faces quando estas eram desconexas e estavam separadas por arestas rosas.

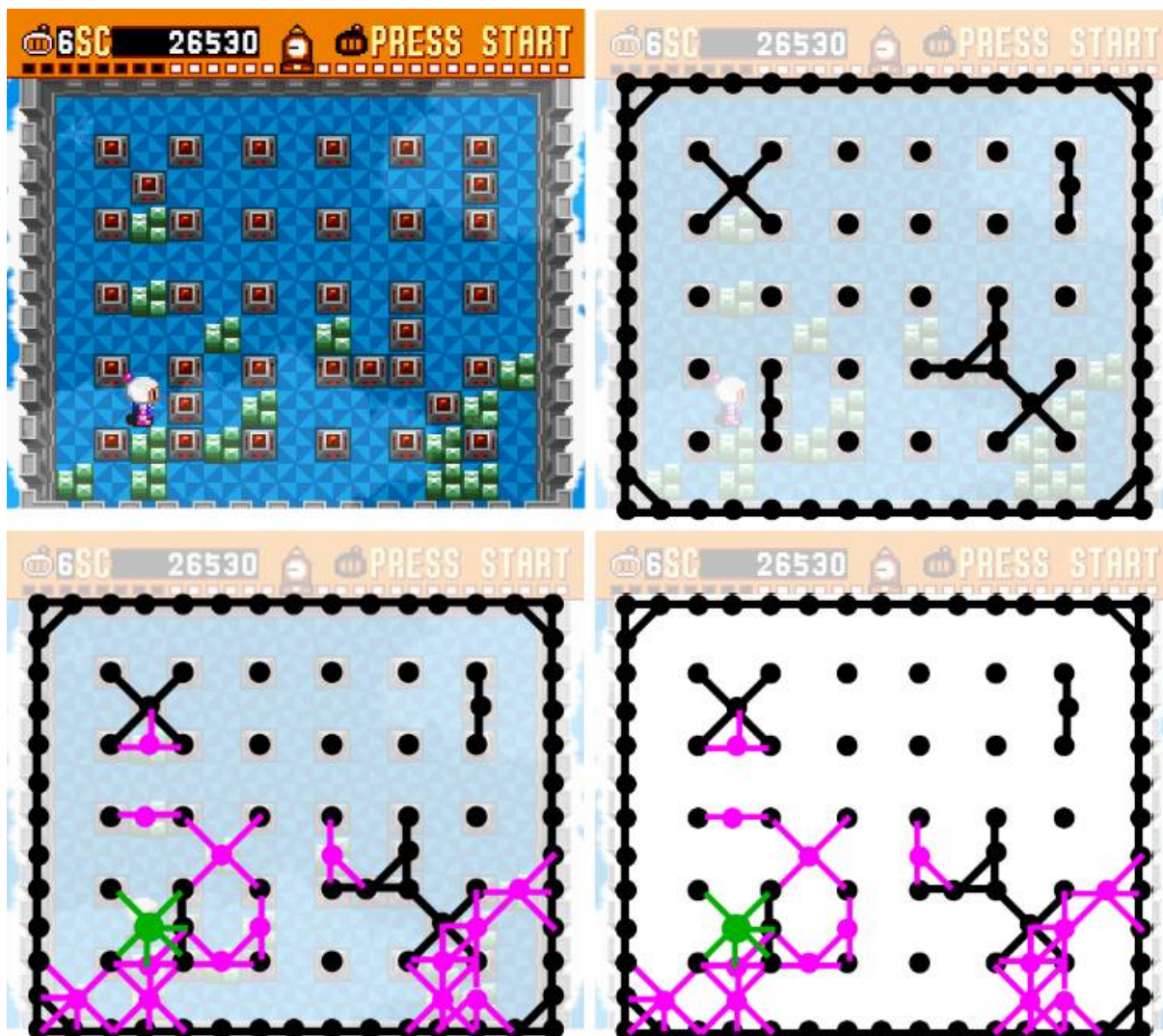


Figura 2: Redução do Super Bomberman a um estado de vértices, arestas e faces.

No passo seguinte do método, Fellows (1996) propõe que a *manivela* seja anexada ao catálogo de problemas NP de Garey e Johnson (1979), e que devemos *girá-la* quantas vezes desejarmos. Para entendermos esta instrução, é essencial termos em mãos o que é a *manivela* e qual a intenção do autor em dizer para *girá-la*. Sendo a *manivela* um instrumento para tratar determinados problemas, anexá-la no catálogo NP significa tomar um dado problema e ver como ele reage com esta ferramenta. Se é possível lidar com as informações do problema a partir destes meios, ainda que não seja considerado adequado. Caso a *manivela* consiga *girar* um problema do catálogo, então devemos manter o esforço em *girá-la* até obtermos uma boa combinação do problema com a *manivela*. Isto formará um *puzzle* ou uma *estrutura de ação*.

Para exemplificar este passo tomamos o problema de encontrar o menor número de passos que solucione uma generalização do *jogo dos 15*, este, um famoso quebra-cabeças de quinze peças composto por uma placa oca de metal com quinze quadrados que trocam de lugar, todos gravados com números, letras ou figuras. Seu desafio é arranjar as peças ordenadamente da esquerda para a direita, de cima a baixo. Goldreich (1984) provou que este é de fato um problema de categoria NP.

Assim, representaremos as quinze peças quadradas do *jogo dos 15* no contexto do Super Bomberman generalizado para vértices, arestas e faces. Podemos considerar cada peça como um vértice capaz de se mover, eliminando suas próprias arestas e gerando novas de acordo com os vértices adjacentes a ele (semelhante ao vértice do personagem). Representaremos estes novos vértices como quadradinhos azuis com um número de 1 a 15 em seu centro. Embora cada um deles tenha arestas conectando-os com os vértices adjacentes, estas são arestas equivalentes à do personagem, ou seja, que se eliminam e surgem sozinhas a medida que se movimentam. Então, para simplificar a representação, ilustraremos o exemplo na figura 3, omitindo-as. Neste contexto, o jogador precisaria solucionar o usual *jogo dos 15*, que embora possua um espaço maior para trabalhar, também possui restrições maiores, como os usuais vértices e arestas indestrutíveis, assim como a existência de faces isoladas.

Se aplicarmos esta *manivela* de forma flexível e formos rigorosos com os critérios de solução do problema nesta instância, teremos criado um *puzzle* onde o jogador deverá em um intervalo de tempo ou movimentos, resolver o problema de ordenar as peças da esquerda para a direita, de cima a baixo (considerando é claro os espaços impossíveis de se ocupar) contando com o aditivo de eliminar vértices e arestas rosas. Contudo, se restringirmos o uso da *manivela* a limitadas eliminações de vértices e arestas rosas, e formos mais flexíveis sobre os critérios de solução, criaremos uma *estrutura de ação* onde o jogador movimentava as peças e elimina somente as arestas que julga essenciais para resolver parcialmente o desafio.

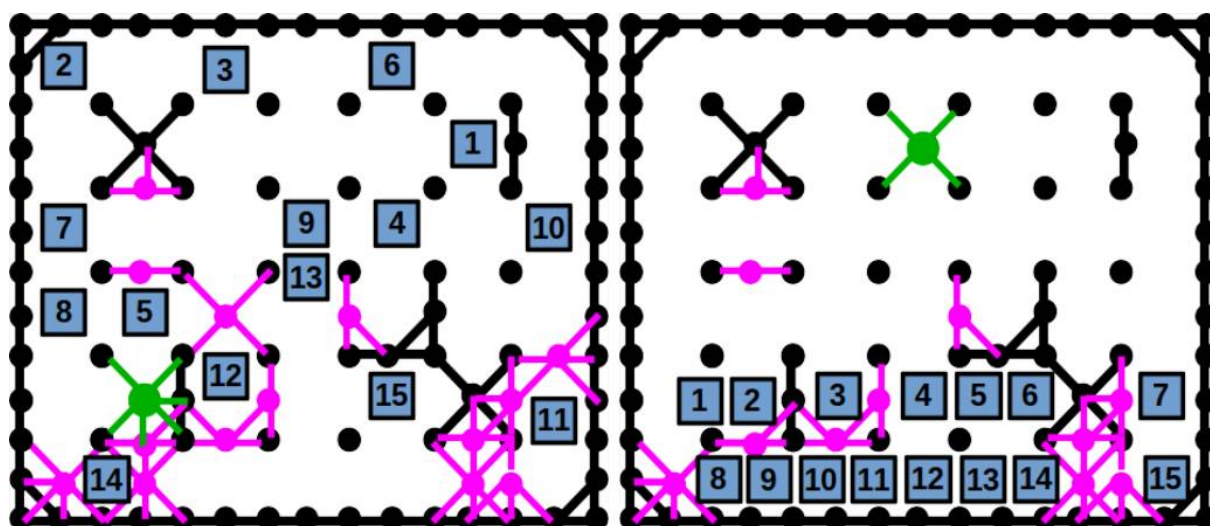


Figura 3: jogo dos 15 combinado com a *manivela* do Super Bomberman.

Fellows (1996) encerra o método dizendo que os jogos digitais não devem ser isolados e totalmente abstratos, precisam também de envolvimento com uma história ao jogador. Para isso, sugere anexá-los a fonte do próprio problema NP que foi trabalhado pela *manivela*, já que a maioria destes problemas foi formulado por uma necessidade de solução real.

## Considerações finais

Talvez as principais dificuldades em aplicar este método seja identificar o funcionamento das *manivelas*, compreender os problemas NP e realizar neles o efeito de *girar a manivela*. Contudo, uma vantagem nítida desta abordagem, é sua capacidade de trabalhar com elementos centrais da estrutura de um jogo, sua combinação na geração de *puzzles* ou *estruturas de ação*, sem em nenhum momento fixar-se a algum contexto concreto. É como se pudéssemos trabalhar com a ideia de um jogo dentro de um campo quase totalmente abstrato, onde estruturas tão diferentes dificilmente seriam pensadas naturalmente, senão na forma de proposições Matemáticas. O método trabalha em mão dupla, a trajetória de discretizar um jogo digital a suas componentes Matemáticas, reduzindo-o para um contexto abstrato, combiná-lo com um difícil mas intuitivo problema, e então transformá-los na estrutura de um novo jogo, seja pela forma de um *puzzle* ou como uma *estrutura de ação*.

A tarefa de explicar o funcionamento deste método a partir de exemplos de um único jogo digital popular, foi um tanto complexa. Pois embora conheça com alguma profundidade o artigo original, a forma como ele foi feito, apoia-se em diversos exemplos pouco conexos entre si. Identificar ao certo o que é cada elemento central de um jogo na visão de Fellows (1996), envolveu entre diversas interpretações e associações de exemplos, também o envio de e-mails com dúvidas para o mesmo. Das quais, mediante algumas discussões com o autor, pude consolidar dentro do jogo Super Bomberman uma base para explicar esta teoria.

## Agradecimentos

Gostaria de agradecer ao professor da Universidade de Bergen na Noruega, Michael Ralph Fellows, pela atenção e dedicação em responder minhas perguntas a medida que estudava seu método.

## Referências

FELLOWS, M. R. The Heart of Puzzling: Mathematics and Computer Games. In: **Proceedings of the 1996 Computer Games Developers Conference**, pp. 109–120, 1996.

GARDNER, H. **The Unschooled Mind: How Children Think and How Schools Should Teach**. New York: Basic Books, 1991.

GAREY, M.; JOHNSON, D. **Computers and Intractability: A Guide to the Theory of NP-completeness**. San Francisco, W.H. Freeman, 1979.

GEE, J. P. Good video games and good learning. **Phi Kappa Phi Forum**, p. 33–37, 2005.

GOLDREICH, O. (1984). Finding the shortest move-sequence in the graph-generalized 15-puzzle is NP-hard. **Laboratory for Computer Science**, MIT, unpublished manuscript. Disponível em <<http://www.wisdom.weizmann.ac.il/~oded/COL/puzzle.pdf>> acessado 14 abr. 2018.

NCTM, National Council of Teachers of Mathematics. **Principles and Standards for School Mathematics**. Reston, VA, 2000.

RETRO GAMER MAGAZINE. **Fell the Burn! Bomberman.** Ip imagine publishing, v. 67, 2009, p. 27–34.