

Engineering Research

Technical Reports

Volume 5 – Issue 3 – Article 1

ISSN 2179-7625 (online)

ARBOT: Augmented Reality System for Teaching Logic

Algorithmic

Tiago De Gaspari¹, Antonio Carlos Sementille², João Fernando

Marar³

MAY / 2014

Taubaté, São Paulo, Brazil

¹ Departamento de Computação, Universidade Estadual Paulista "Júlio de Mesquita Filho", Bauru, Brasil, gaspari@fc.unesp.br.

² Departamento de Computação, Universidade Estadual Paulista "Júlio de Mesquita Filho", Bauru, Brasil, semente@fc.unesp.br.

³ Departamento de Computação, Universidade Estadual Paulista "Júlio de Mesquita Filho", Bauru, Brasil, fermarar@fc.unesp.br.

Engineering Research: Technical Reports

Technical Editor: Giorgio Eugenio Oscare Giacaglia, Universidade de Taubaté, Brazil

Associate Technical Editors

Eduardo Hidenori Enari, Universidade de Taubaté, Brazil

Wendell de Queiróz Lamas, Universidade de São Paulo at Lorena, Brazil

Editorial Board

Antonio Faria Neto, Universidade de Taubaté, Brazil

Asfaw Beyene, San Diego State University, USA

Bilal M. Ayyub, University of Maryland, USA

Bob E. Schutz, University of Texas at Austin, USA

Carlos Alberto de Almeida, Pontifícia Universidade Católica do Rio de Janeiro, Brazil

Ciro Morlino, Università degli Studi di Pisa, Italy

Eliane da Silveira Romagnoli Araujo, Universidade de Taubaté, Brazil

Epaminondas Rosa Junior, Illinois State University, USA

Evandro Luís Nohara, Universidade de Taubaté, Brazil

Fernando Manuel Ferreira Lobo Pereira, Universidade do Porto, Portugal

Gilberto Walter Arenas Miranda, Universidade de Taubaté, Brazil

Hubertus F. von Bremen, California State Polytechnic University Pomona, USA

João Bosco Gonçalves, Universidade de Taubaté, Brazil

Jorge Muniz Júnior, Universidade Estadual Paulista at Guaratinguetá, Brazil

José Luz Silveira, Universidade Estadual Paulista at Guaratinguetá, Brazil

José Walter Parquet Bizarria, Universidade de Taubaté, Brazil

María Isabel Sosa, Universidad Nacional de La Plata, Argentina

Ogbonnaya Inya Okoro, University of Nigeria at Nsukka, Nigeria

Paolo Laranci, Università degli Studi di Perugia, Italy

Rolando A. Zanzi Vigouroux, Kungliga Tekniska högskolan, Sweden

Sanaul Huq Chowdhury, Griffith University, Australia

Tomasz Kapitaniak, Politechnika Łódzka, Poland

Zeki Tüfekçioğlu, Ankara Üniversitesi, Turkey

The “Engineering Research” is a publication with purpose of technical and academic knowledge dissemination.

BIOGRAPHIES

Tiago De Gaspari is graduated in Bachelor of Information Systems from Sao Paulo State University (2012). Currently, he is a student of the Master's program in Computer Science from Sao Paulo State University and scholarship by the National Council for Scientific and Technological Development (CNPq). Have experience in development of Augmented Reality systems. Have interest in Virtual and Augmented Reality, Image Processing, Evaluation of video quality and the development of electronic games.

Antonio Carlos Sementille is graduated in Computer Science from Sao Paulo State University (1988), Masters in Computer Science from the Federal University of São Carlos (1994), PhD in Sciences (Computational Physics) from the University of São Paulo (1999) and thesis in Advanced Interfaces UNESP (2010). He is currently assistant professor, exclusive dedication (RDIDP) at Sao Paulo State University (UNESP-Bauru), Coordinator of the Real Time Systems and Chairperson of the Ad Hoc Research Foundation of the State of São Paulo (FAPESP). Since 07/2009 is teaching and guiding the Programme Postgraduate (MSc) in Computer Science - UNESP - São José do Rio Preto, as well as teaching and guiding the Programme Postgraduate (MSc) in Digital TV - FAAC - UNESP / Bauru. He has experience in Computer Science with emphasis on the following topics: Virtual Reality, Augmented Reality, Image Processing, Distributed Virtual Environments, Collaborative Environments, Computer Networks and Design.

João Fernando Marar receives his B.S. in Computer Science from the University of São Paulo (USP - Sao Carlos), in 1985, his M.S. in Computer Science and Computational Mathematics from the University of São Paulo, USP Sao Carlos, (1992), his Ph.D. in Computer Science from the Federal University of Pernambuco, Recife UFPe, in 1997, and his Post Doctoral from University of Lisbon, Department of Informatics, in 2008. He is Productivity CNPq Fellow between 2009-2013. He is currently Adjunct Professor at Sao Paulo State University (UNESP-Bauru), Coordinator of Adaptive Systems and Intelligent Computing and Ad Hoc Advisory of the National Council for Scientific and Technological Development, FAPESP, CAPES and FAPEMIG. He has experience in the area of Computer Science working on the following topics: Neural Networks, Artificial Intelligence, Design and Emotion Processing, Recognition and Classification of Images and Signals, Transforms Wavelets, Design, Materials Selection and Manufacturing Processes

PRESENTATION

This article describes results of the Undergraduate Thesis presented to São Paulo State University in Partial Fulfilment of the Requirements for the Degree of Honours Degree in Bachelor of Information Systems.

TABLE OF CONTENTS

Section	Page
Editorial Board	ii
Biographies	iii
Presentation	iv
Abstract	6
Introduction	6
Related Work	8
ARBOT System	9
Results and Discussions	15
Conclusions	22
References	23

ARBOT: SISTEMA EM REALIDADE AUMENTADA PARA O ENSINO DE LÓGICA ALGORÍTMICA

Este artigo propõe o ARBot, um sistema que tem como objetivo principal, a apresentação de conceitos de lógica para alunos do ensino fundamental e médio. O sistema foi desenvolvido utilizando-se a tecnologia conhecida como Realidade Aumentada (RA), a qual permite complementar o ambiente real, onde está o usuário, por meio da adição de objetos virtuais. Neste cenário criado a partir da RA, uma interface de jogo virtual é utilizada, por intermédio da qual são apresentados desafios cognitivos. Para resolver estes desafios, os usuários devem programar personagens virtuais tridimensionais utilizando uma linguagem visual. Como resultado tem-se que, de forma lúdica, conceitos de algoritmos e programação são assimilados pelos usuários. Além disso, o sistema possibilita a interação de dois usuários em um modo cooperativo de jogo. Neste modo cooperativo, o sistema foca na aprendizagem colaborativa, uma vez que permite que os usuários, em conjunto, resolvam o desafio cognitivo apresentado pelo sistema.

Palavras-chave: aprendizagem colaborativa; ferramenta de ensino; lógica algorítmica; realidade aumentada.

ARBOT: AUGMENTED REALITY SYSTEM FOR TEACHING LOGIC ALGORITHMIC

This article purposes the ARBot, a system that has as main objective the presentation of concepts of logic for students of elementary and secondary education. The system was developed using the technology known as Augmented Reality (AR), which allows complement the actual environment where the user is, by adding virtual objects. In this scenario the RA created from a virtual game interface is used, through which cognitive challenges are presented. To solve these challenges, users must set up three-dimensional virtual characters using visual language. As a result it follows that, in a playful way, concepts of algorithms and programming are assimilated by users. In addition, the system enables two users to interact in a cooperative game mode. In cooperative mode, the system focuses on collaborative learning, since it allows users to jointly solve the cognitive challenge presented by the system.

Keywords: collaborative learning; teaching tool; algorithmic logic; augmented reality.

1. INTRODUÇÃO

Um dos problemas enfrentados pelas ferramentas de ensino é a criação de conteúdo adequado para os alunos. Caso o conteúdo não seja motivador, os alunos podem se sentir entediados ou até mesmo desistir da escola ou do curso em que se matricularam. Outro desafio é fazer com que os alunos gostem de estudar quando estão em suas próprias casas.

Portanto, as ferramentas de ensino, principalmente quando utilizadas pelos alunos apenas em

suas casas, como no caso de cursos de EAD (Ensino à Distância), devem possuir certas características motivacionais.

Para a construção de um ambiente de ensino inovador e motivador, tem-se atualmente a tecnologia da Realidade Aumentada (RA). De acordo com Dede (2008), a RA pode ser considerada como o meio pedagógico da “próxima geração”, no que diz respeito ao aprimoramento da qualidade da aprendizagem.

Um sistema de RA é descrito como um sistema que combina elementos virtuais com o ambiente real, é interativo, tem processamento em tempo real e é concebido em três dimensões (AZUMA, 1997). Com a utilização da RA, é possível criar ambientes ricos e complexos, pois ela constrói cenas onde cenários reais se misturam com objetos virtuais. Para isso, é preciso no mínimo de uma câmera que capture as imagens do mundo real e de um software que crie objetos virtuais e realize a união entre esses dois mundos (CONSULARO *et al.*, 2007).

O avanço da multimídia e da Realidade Virtual (RV), proporcionado pela maior potência dos computadores, permitiu a integração, em tempo real, de vídeo e ambientes virtuais interativos. A RA, enriquecendo o ambiente físico com objetos virtuais, beneficiou-se desse progresso, fazendo com que aplicações dessa tecnologia tanto em plataformas sofisticadas quanto em plataformas populares, se tornassem viáveis (KIRNER; TORI, 2006).

Uma das maiores vantagens do uso da RA é a motivação do usuário. Esta motivação, além de favorecer o aprendizado do aluno e a memorização do conteúdo, é crucial para que a RA e qualquer outra tecnologia seja disseminada. Usuários motivados, principalmente crianças, tendem a comentar com outros alunos, com os familiares e com outras pessoas a experiência que tiveram, fazendo com que o assunto desperte o interesse nesta tecnologia na sociedade de uma maneira mais ampla.

A potencialização de ferramentas didáticas é uma das principais vantagens que pode ser alcançada com esta tecnologia. Muitos já consideram a RA como o futuro da educação, pois quando se tem o interesse e a atenção dos alunos focada em um determinado objetivo, os resultados são extremamente positivos (GONÇALVES *et al.*, 2010).

Aplicações de RA para o ensino de conceitos de lógica algorítmica vêm sendo empregadas nos últimos anos (GASPARI *et al.*, 2012), devido à dificuldade e falta de motivação dos alunos em aprenderem esses assuntos, uma vez que a maneira tradicional normalmente é pouco desafiadora.

Considerado o contexto exposto, o objetivo do sistema ARBot é criar um ambiente como um jogo virtual, onde o usuário deve passar diversos níveis de dificuldade. Esses níveis são montados, de uma maneira que o usuário possa aprender conceitos sobre lógica, vistos em cursos de Introdução à Lógica de Programação, por exemplo. Além disso, a proposta é que a interação com ambiente gerado pelo sistema seja uma atividade lúdica, pois o público-alvo são estudantes do

ensino fundamental e médio. Isso se torna possível ao utilizar o conceito de linguagem visual, onde o programa escrito pelo usuário é constituído apenas de símbolos (ícones). A criação de animações dos objetos 3D, que representam o programa escrito pelo usuário de maneira visual, consegue motivar o usuário a concluir um determinado nível.

Um conceito usado neste sistema, e que combina perfeitamente com o intuito de motivação do aluno, é a aprendizagem colaborativa. Aprendizagem colaborativa é definida como uma técnica por meio da qual os estudantes se ajudam no processo de aprendizagem, atuando como parceiros entre si e com o professor, enquanto trabalham para adquirir conhecimento sobre um dado objeto (SMYSER, 1993). Essa abordagem visa ganhos não só em relação ao próprio processo ensino-aprendizagem, mas também na preparação dos indivíduos para situações futuras, onde cada vez mais, são exigidas pessoas aptas ao trabalho em equipe.

O sistema proposto utiliza o conceito de aprendizagem colaborativa como aliado na criação de desafios para os alunos, que dependerão de seus companheiros para executar determinadas tarefas, para assim alcançar um determinado objetivo em comum, conseguindo também, muitas vezes sem imaginar, aprender com esta experiência.

O objetivo principal deste estudo é explorar as influências de uma simulação apoiada por RA sobre a eficácia de aprendizagem de conceitos de lógica de programação. A principal motivação para o desenvolvimento deste sistema é mostrar como a RA é uma ferramenta extremamente funcional e motivadora no processo de aprendizagem. Outro fator importante a ser avaliado é a utilização de um sistema de ensino que utilize a RA por mais de um usuário simultaneamente, e como a aprendizagem colaborativa pode ser alcançada em um ambiente aumentado.

2. TRABALHOS CORRELATOS

Diversos estudos buscam a integração da RA com sistemas educacionais. A criação de objetos de aprendizagem, a geração de conteúdo motivador ou até mesmo novas formas de ensinar conceitos complexos são buscados por educadores e a RA permite que muitas barreiras entre o professor, o aluno e o conteúdo sejam ultrapassadas.

O Construct 3D (KAUFMANN *et al.*, 2000) é uma ferramenta que utiliza a RA e possibilita os usuários a aprenderem conceitos de geometria espacial e matemática de uma maneira diferente à convencional. Como interface, este sistema utiliza um painel criado em RA, onde são exibidas formas de interação com os modelos virtuais. Para a visualização, as informações são projetadas em capacetes de RV ou em monitores convencionais. O grande diferencial deste sistema é a possibilidade de vários usuários, utilizando diferentes capacetes e diferentes painéis, visualizarem o mesmo objeto no ambiente aumentado e inclusive realizar alterações nos modelos virtuais simultaneamente.

Uma ferramenta flexível e poderosa de prototipagem e desenvolvimento de aplicações distribuídas e colaborativas de RV e RA é o ARTiFICe (MOSSEL *et al.*, 2012). O ARTiFICe permite o desenvolvimento de aplicações em diversas plataformas e com diversos focos diferentes. É facilitada a criação de aplicativos para celular ou utilizando diferentes dispositivos de interface como o Microsoft Kinect, Razer Hydra, Sony Move e SpaceNavigator.

O trabalho apresentado por Kaufmann e Meyer (2008) mostra como conceitos complexos podem ser ensinados de uma forma diferente à convencional. Segundo os autores, muitas vezes os alunos possuem dificuldade em entender conceitos de física e mecânica clássica, uma vez que a teoria ensinada pelos professores não motivam os alunos. Ao contrário de apresentar apenas a teoria, um sistema de RA possibilita a demonstração dos fenômenos mecânicos em tempo real utilizando modelos tridimensionais, ao contrário de apenas figuras em um livro. Além da visualização, o sistema permite a interação dos usuários com o ambiente aumentado, onde características dos modelos tridimensionais podem ser alteradas, como o peso, suas dimensões e tipos de material, causando consequências em tempo real no ambiente. Ao utilizar um ambiente aumentado gerado pela RA, fenômenos mecânicos e físicos podem ser estudados e discutidos em tempo real de uma maneira mais realista e motivadora.

3. SISTEMA ARBOT

Nesta seção é descrita a arquitetura do sistema, detalhes sobre sua implementação e como a cena aumentada foi estruturada. Destacam-se os diagramas UML utilizados para estruturar o projeto e as ferramentas utilizadas para seu desenvolvimento.

3.1. Ferramentas Utilizadas e Arquitetura do Sistema

As versões gratuitas de três bibliotecas foram utilizadas para o desenvolvimento, escritas em C e C++: ARToolKit, OpenSceneGraph e Osgart.

A biblioteca ARToolKit utiliza rastreamento óptico, implementando técnicas de visão computacional para identificar e estimar, em tempo real, a posição e orientação de marcadores fiduciais em relação ao dispositivo de captura vídeo. Assim, é realizado o cálculo da correlação entre os dados estimados do marcador e da sua imagem real, que permite posicionar os objetos virtuais alinhados com a imagem do marcador, como mostra a Figura 1 (KATO, 2000).

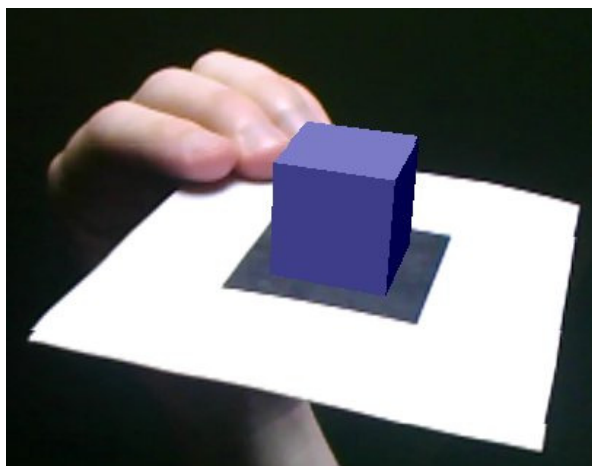


Figura 1 – Exemplo de objeto virtual alinhado com um marcador real

A OpenSceneGraph é uma biblioteca de RV que usa uma estrutura chamada grafo de cena para a construção de cenas virtuais complexas. Ele permite o uso de texturas, iluminação e alterações dinâmicas da cena através do gráfico hierárquico que representa as relações entre todos os objetos e efeitos tridimensionais. Uma de suas características, que é de extrema importância, é o grande número de plug-ins para carregar diferentes modelos 3D, como 3D Studio, Lightwave, Wavefront OBJ, COLLADA e VRML.

Finalmente, a biblioteca OSGART foi utilizada para combinar as características descritas nas bibliotecas anteriores. A OSGART segue os mesmos princípios da ARToolkit, fornecendo uma maneira fácil para desenvolvimento de aplicações de RA com a vantagem da integração simplificada da RA com outros kits de ferramentas de visualização e simulação, como a OpenSceneGraph (LOOSER *et al.*, 2006).

Foram escolhidas outras duas bibliotecas para o desenvolvimento do sistema que não utilizam nenhum conceito de RV ou RA: CXBOXController e TinyXML.

A CXBOXController é uma biblioteca simples, que auxilia o desenvolvimento de aplicações na linguagem C++, permitindo a interação do usuário com o sistema por meio de até quatro controles do console XBOX 360 simultaneamente. Ela foi desenvolvida para ser utilizada em plataformas Windows, pois utiliza funções presentes no XINPUT que é uma API que controla a troca de informações entre o computador e controles do XBOX 360.

A TinyXML é uma biblioteca simples, pequena, porém robusta, que facilita a leitura e escrita em arquivos do tipo XML (*eXtensible Markup Language*) em sistemas desenvolvidos em C++. As bibliotecas utilizadas no desenvolvimento do sistema são organizadas e construídas sobre uma hierarquia que é apresentada na Figura 2.

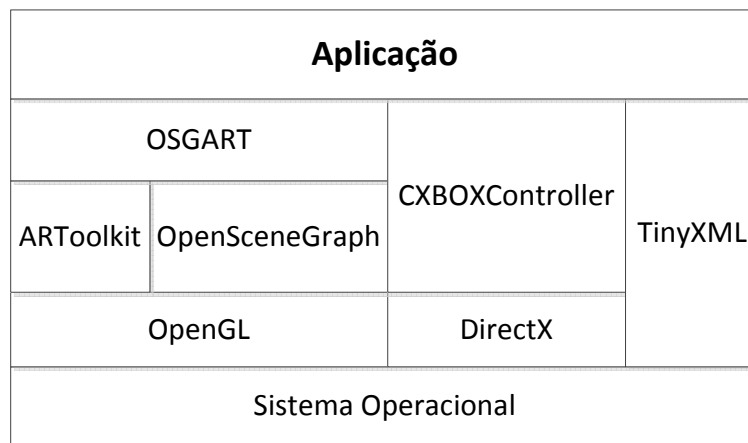


Figura 2 – Estrutura hierárquica das bibliotecas utilizadas pela aplicação

O sistema possui, de maneira geral, dois fluxos de informação.

O primeiro fluxo inicia-se no momento em que o usuário acessa o sistema e é referente ao carregamento dos dados para a geração dos níveis que são exibidos em uma ordem ao usuário no ambiente aumentado. A Figura 3 representa o diagrama de fluxo deste carregamento.

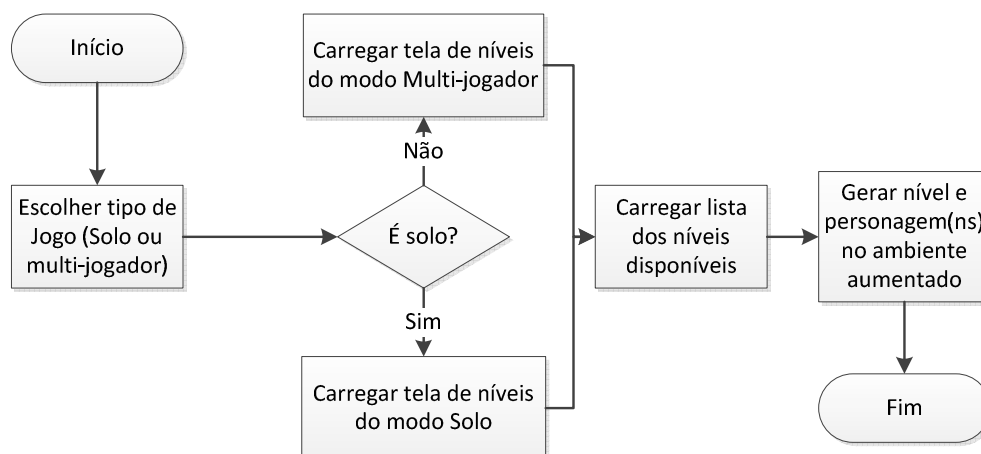


Figura 3 – Diagrama que demonstra os passos no carregamento dos dados para iniciar um nível

Após a escolha do modo de jogo e o carregamento dos arquivos onde estão os níveis, o usuário utiliza o sistema para realizar a programação do personagem para cumprir os desafios cognitivos propostos. O fluxo de informações da interação entre o sistema e o usuário ao decorrer de um nível é apresentado na Figura 4.

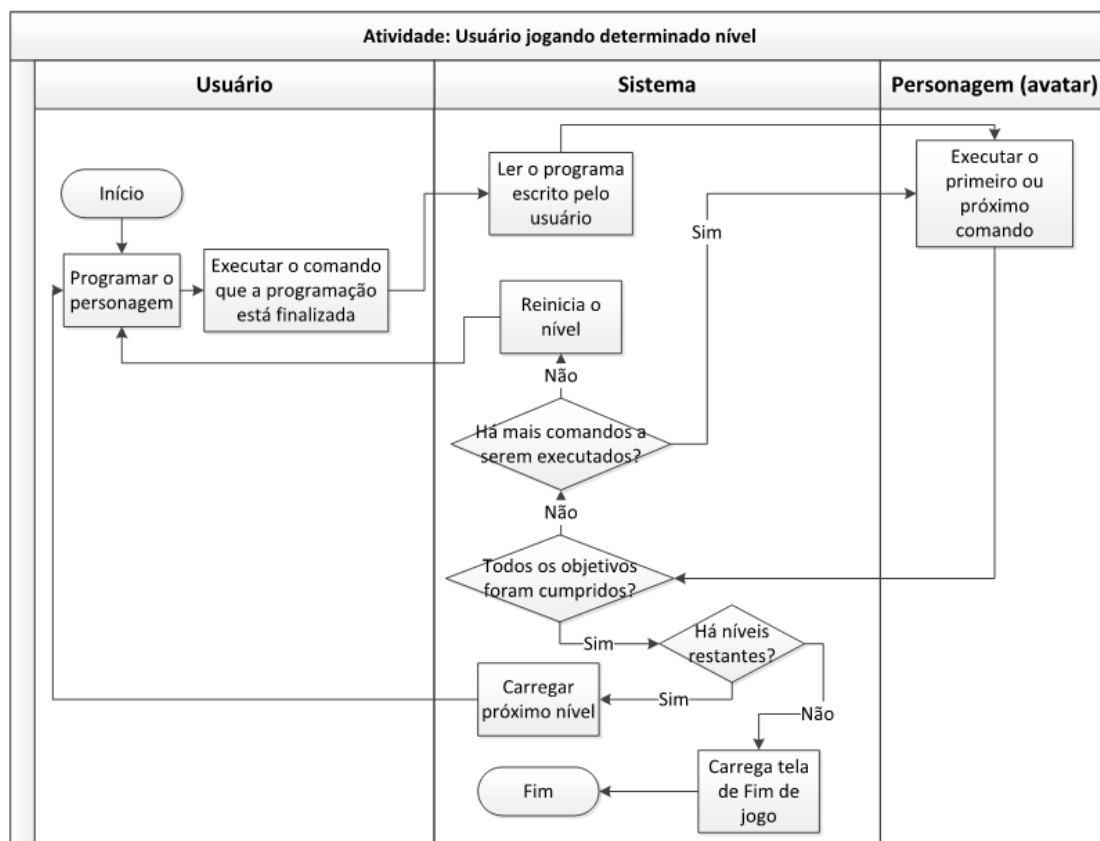


Figura 4 – Diagrama que exhibe os passos no carregamento dos dados para iniciar um nível

O sistema foi concebido para interagir com dois tipos de usuários: alunos e professores. A Figura 5 apresenta o diagrama de casos de uso do sistema.

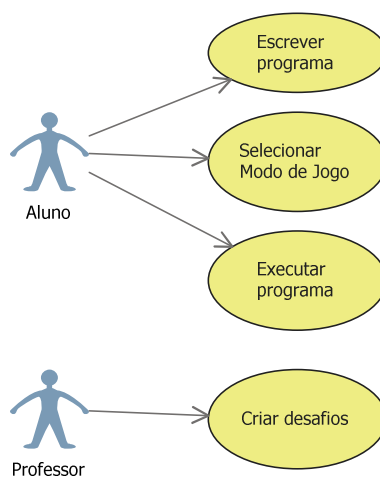


Figura 5 – Diagrama de casos de uso

3.2. Implementação

Para a implementação do sistema foi utilizado o paradigma de programação orientada a objetos, assim a implantação de melhorias no futuro é facilitada.

O sistema possui uma classe chamada **Viewer**. Esta classe é utilizada como um agregador de informações a serem exibidas ao usuário, ou seja, todas as informações visuais são associadas a esta classe.

Além do **Viewer**, o sistema funciona como uma máquina de estados, onde para cada momento, um estado diferente é associado à aplicação. Foram criados dois estados, representados por telas exibidas ao usuário: a tela de seleção de níveis e a tela do nível selecionado.

A tela de seleção de níveis é constituída por uma câmera virtual e um objeto chamado de apontador, que permite ao usuário selecionar os níveis e opções do sistema antes de começar a jogar.

A tela de nível é a tela principal do sistema, nela é exibido o ambiente aumentado utilizado pelo usuário. Para cada objeto da classe que representa a tela de nível são criados todos os objetos que constituem a cena aumentada, e por serem associados ao **Viewer**, são exibidas ao usuário. Neste estado também está presente a classe de apontador, uma vez que o usuário também faz escolhas nesta fase.

A Figura 6 apresenta o diagrama de classes do sistema com as principais classes e seus relacionamentos.

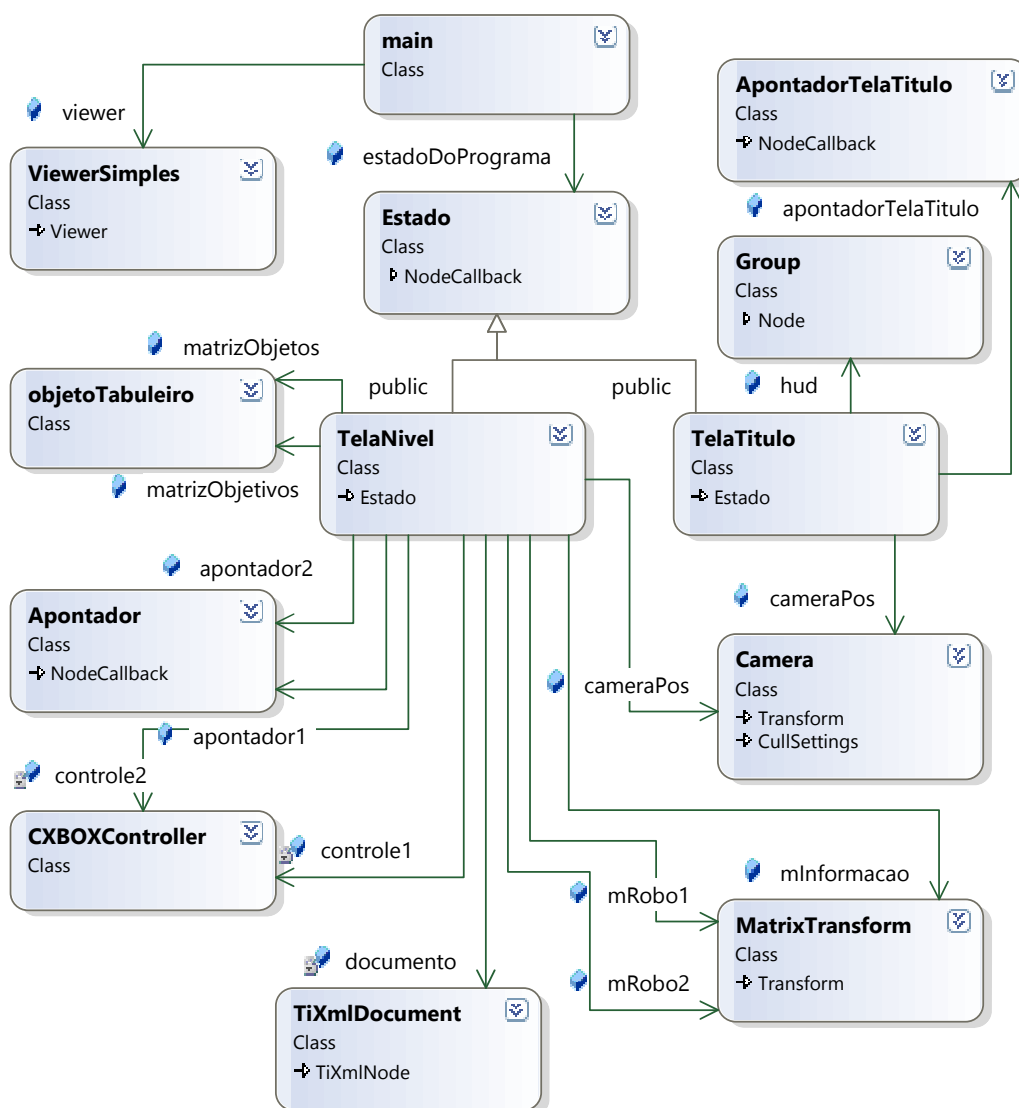


Figura 6 – Diagrama de classes do sistema

Para organização da cena aumentada, foi utilizada uma estrutura de dados chamada de grafo de cena.

Um grafo de cena é uma estrutura de dados geral que define a relação espacial e lógica de uma cena gráfica para a gestão eficiente e renderização de dados gráficos. É tipicamente representado como uma árvore hierárquica e não permite que exista um ciclo fechado ou então um elemento isolado, ou seja, que não possua nenhuma ligação com a estrutura principal (WANG; QIAN, 2010).

A utilização de um grafo de cena faz com que a renderização dos elementos virtuais na cena aumentada seja muito mais rápida, pois dependendo do estado da aplicação, determinados objetos não precisam ser exibidos.

Foi construído um grafo de cena que representa a estrutura da cena aumentada gerada. Foi, também, realizada uma separação entre a estrutura geral do sistema e a cena aumentada gerada para

cada nível carregado no sistema. Na Figura 7 é apresentado o grafo de cena que representa o sistema como um todo.

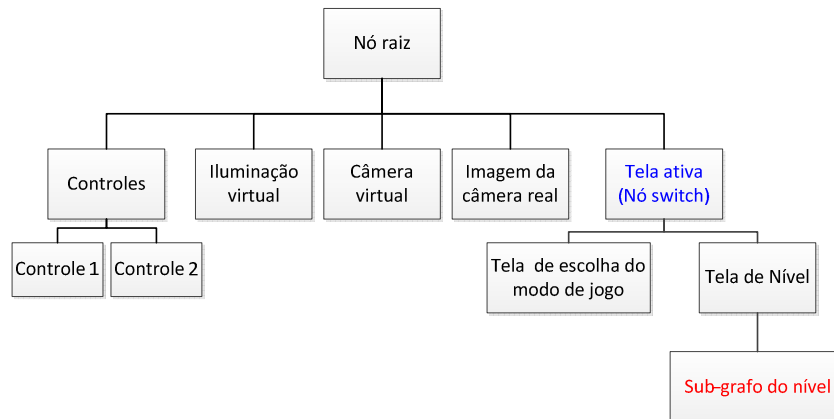


Figura 7 – Grafo de cena geral do sistema

No grafo de cena apresentado na Figura 7 foi criado um nó chamado de “Tela ativa”. Este nó é do tipo switch e permite que o sistema faça o processamento de apenas um dos filhos deste nó, dependendo do estado da aplicação. Este tipo de nó foi utilizado para que fosse possível a alteração entre as telas de escolha do modo de jogo para as telas de níveis que são carregadas.

Outro nó que foi destacado é o nó de “Sub-grafo do nível”. Este nó representa outro grafo ilustrado pela Figura 8. Na Figura 8 é apresentado o sub-grafo do nível, pertencente ao grafo de cena principal do sistema.

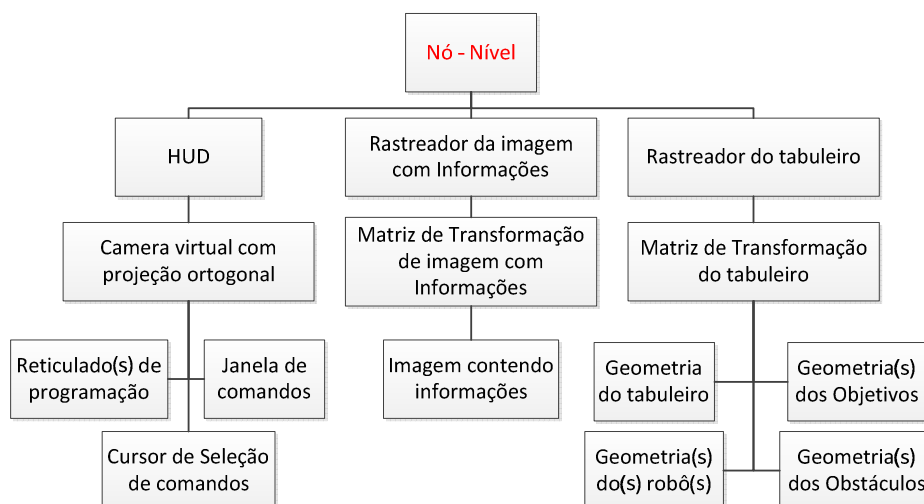


Figura 8 – Sub-grafo que representa o nó de nível do grafo principal do sistema

4. RESULTADOS OBTIDOS

Nesta seção são descritas a geração do ambiente aumentado, a personalização do ambiente gerado por meio de arquivos XML, a interface com o usuário e, por fim, o resultado obtido ao final

do projeto.

4.1. Geração do Ambiente Aumentado

Como mencionado anteriormente, o objetivo principal do sistema é gerar um ambiente aumentado que possibilite aos usuários a programação de personagens (representados por robôs) virtuais para cumprirem objetivos pré-estabelecidos. Para isso foi criado um tabuleiro virtual no formato de uma matriz bidimensional de 10 linhas por 10 colunas.

A forma de um tabuleiro e a escolha por criar a cena aumentada sobre o chão, ou sobre uma mesa, foi escolhida pela similaridade com um tabuleiro real, e devido à familiaridade que a maioria dos usuários possui com este tipo de jogo, pois é construído de forma parecida a jogos de dama e xadrez.

Neste tabuleiro são posicionados um ou dois personagens, assim como os obstáculos e os espaços do tabuleiro que representam os objetivos. É importante frisar que a posição destes personagens, assim como os obstáculos e os objetivos, são criados dinamicamente e de forma diferente para cada nível proposto ao usuário.

Todos estes objetos que compõe o tabuleiro são posicionados sobre um marcador real na cena. Na Figura 9 é apresentado o tabuleiro vazio (sem nenhum personagem, obstáculo ou objetivo) sobre o marcador.

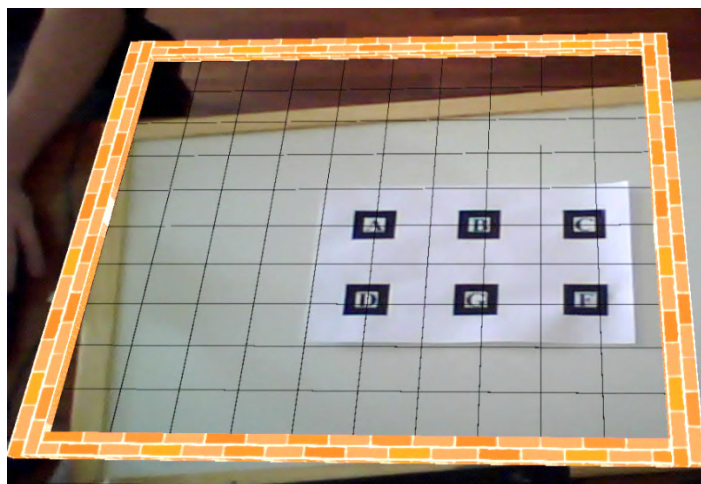


Figura 9 – Tabuleiro virtual vazio posicionado na cena aumentada

Como o objetivo do sistema é gerar um ambiente onde o usuário deve cumprir alguns objetivos, estes foram divididos em diversos níveis como em um jogo virtual. Para cada nível ser gerado de uma maneira diferente, com obstáculos e objetivos diferentes, foi criada uma estrutura baseada em arquivos XML para que o sistema pudesse ser personalizado a partir deles.

4.2. Personalização do Ambiente Aumentado

O sistema consiste de vários níveis que devem ser carregados e exibidos aos usuários. Estes níveis são criados a partir de arquivos do tipo XML. Estes arquivos possuem informações referentes ao posicionamento dos obstáculos, objetivos e personagens que serão colocados sobre o tabuleiro virtual, para cada nível.

Inicialmente, para prova de conceito, foram criados dois tipos de obstáculos, três tipos de objetivos e dois personagens.

Os obstáculos são divididos em duas categorias: obstáculos fixos (representados por uma textura semelhante a pedra) e obstáculos móveis (representados por uma textura semelhante a madeira). No caso do obstáculo fixo, este é utilizado como uma barreira, a qual o personagem não consegue transpor. O obstáculo móvel é parecido com o fixo, porém, o personagem tem a possibilidade de movê-lo utilizando alguns comandos específicos.

Na Figura 10 é apresentada a representação gráfica destes obstáculos, posicionados na cena aumentada.

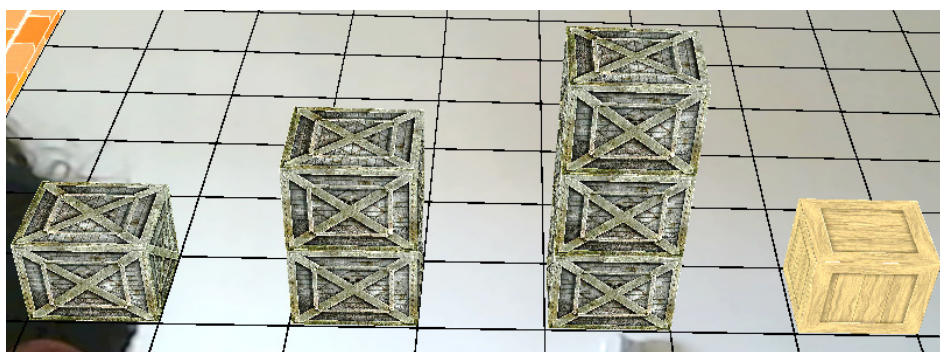


Figura 10 – Representação gráfica dos obstáculos criados

Três diferentes tipos de objetivos também foram criados. Um deles é referente ao obstáculo móvel, outro ao personagem 1, e por fim o obstáculo referente ao personagem 2. A Figura 11 apresenta a representação gráfica destes objetivos, na ordem mencionada.



Figura 11 – Representação gráfica dos objetivos criados

Cada um destes objetivos deve ser acionado pelos personagens controlados pelos usuários por meio de comandos específicos que serão descritos mais adiante.

Por fim, foram criados dois personagens representados por robôs virtuais, onde cada um deles representa um jogador no ambiente aumentado, ou seja, é o seu avatar. A Figura 12 apresenta estes personagens.

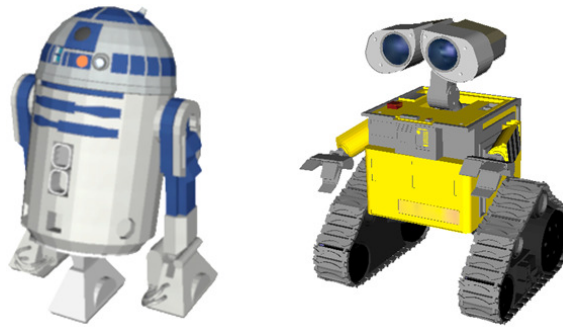


Figura 12 – Modelos tridimensionais dos robôs utilizados como personagens

A Figura 13 apresenta o tabuleiro na cena aumentada já com os personagens, os obstáculos e os objetivos posicionados sobre ele.

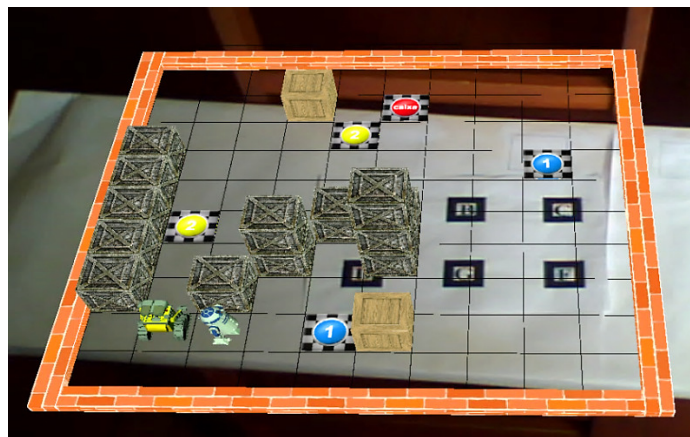


Figura 13 – Tabuleiro preenchido posicionado na cena aumentada

4.3. Interação com os Usuários

A interação do sistema com o usuário se dá por meio de menus do tipo HUD (*Heads Up Display*).

O conceito de HUD se originou a partir da sua utilização em aeronaves, visando fornecer informações ao piloto, sem que este tivesse que desviar os olhos do alvo à frente da aeronave. Atualmente, o HUD é amplamente utilizado em jogos eletrônicos, onde informações são mostradas sobre a imagem criada do ambiente do jogo. No sistema desenvolvido, um HUD foi criado para identificar diferentes escolhas do usuário.

O HUD criado consiste de três reticulados para cada usuário, que são utilizados para montar um programa em linguagem visual. Este programa será lido pelo sistema fazendo com que o

personagem se movimenta de acordo com a programação que foi criada pelo usuário e assim cumpra os objetivos do nível.

Para programar o personagem, o usuário deve selecionar o campo que deseja inserir algum comando, por meio dos botões direcionais do controle ou pelo teclado. O campo selecionado é representado por uma borda mais escura que a cor do respectivo reticulado. Na Figura 14 são apresentados os reticulados e os campos selecionados (segunda linha, segunda coluna da função principal).

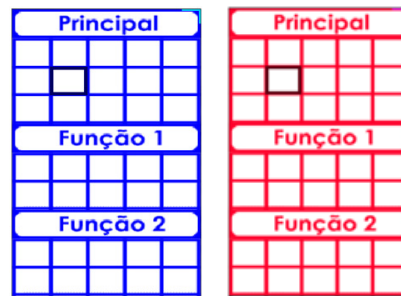


Figura 14 – Reticulados utilizados para a programação dos personagens

Após a escolha do campo pelo usuário, a janela de comandos é exibida sobre o reticulado de programação, com os comandos disponíveis. A Figura 15 apresenta a janela de comandos que é exibida ao selecionar um campo do reticulado de programação.

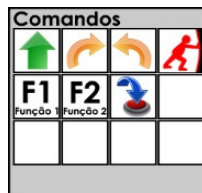


Figura 15 – Janela de comandos

Os comandos que podem ser dados aos personagens criados, assim como a ação que o personagem executa são listados a seguir:

- **Andar:** faz com que ele se desloque um quadrado do tabuleiro à frente;
- **Girar (anti-horário / horário):** faz com que ele gire 90 graus em seu próprio eixo no sentido escolhido;
- **Empurrar:** faz com que ele empurre um objeto móvel e se locomova um espaço à frente, no lugar onde o objeto empurrado se encontrava;
- **Acionar Botão:** faz com que ele salte sobre um objetivo e no caso deste ser um objetivo referente ao personagem que o está acionando, faz com que o objetivo desapareça,

mostrando que foi cumprido;

- **Função 1:** faz com que o fluxo de comandos principal sofra um desvio, para a função 1, fazendo com que o personagem continue executando as ações que se encontram dentro do espaço destinado à programação da função 1; e
- **Função 2:** possui a mesma funcionalidade da Função 1, porém os comandos executados são referentes ao espaço da Função 2.

Na Figura 16 é apresentado um exemplo de programa escrito nesta linguagem visual criada.



Figura 16 – Exemplo de programa escrito com a linguagem visual criada

Foi criado um segundo marcador exclusivamente para exibir uma imagem para cada nível no ambiente aumentado. Assim é possível explicar os objetivos de cada nível ou até mesmo fornecer dicas para o usuário. Na Figura 17 é apresentado o marcador posicionado na cena aumentada com a imagem contendo as informações sobre ele.

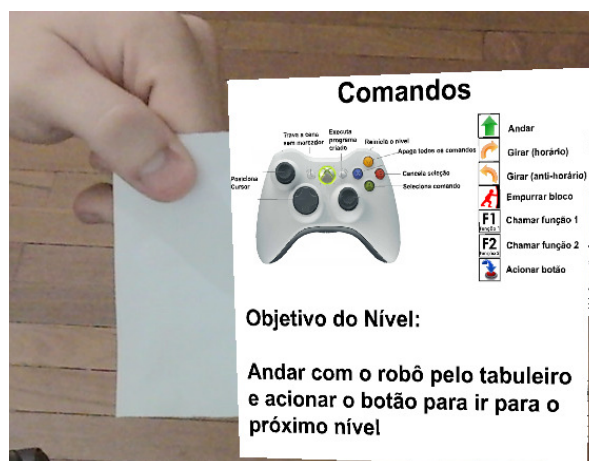


Figura 17 – Marcador com informações sobre o nível exibidas sobre ele

Esta maneira de exibição de informação foi escolhida, pois o usuário pode esconder facilmente as informações apenas retirando o marcador da visão da câmera. Além disso, evita-se que muitas informações sejam exibidas na tela ao mesmo tempo.

4.4. Resultado Final

O resultado obtido ao final do desenvolvimento do sistema é um ambiente aumentado que disponibiliza vários níveis para um ou dois jogadores, a serem cumpridos escrevendo programas em linguagem visual para a movimentação de personagens representados por robôs virtuais. A Figura 18 representa a interface de um desafio simples gerado para apenas um jogador e o programa que resolve este desafio.

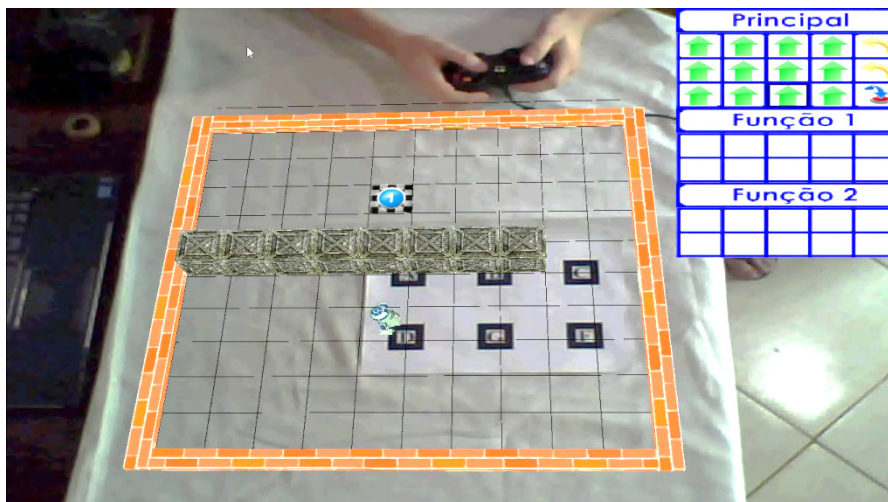


Figura 18 – Desafio simples para um jogador

Esses níveis podem ser cumpridos utilizando comandos simples apenas utilizando o reticulado de programação “Principal”. Outros níveis necessitam o uso das funções 1 e 2.

Quando é preciso muitas repetições dos comandos é preciso utilizar o conceito de recursividade. É importante frisar que mesmo que o programa escrito pelo usuário não possua final, ou seja, seja recursivo e sem uma maneira de sair do fluxo, é possível concluir o nível, cumprindo todos os objetivos mesmo que haja comandos a serem executados.

Há níveis montados especificamente para a aprendizagem colaborativa. Estes níveis possuem caixas móveis que devem ser empurradas por um dos personagens para liberar espaço para que os dois possam cumprir os objetivos. Caso um jogador não colabore com o outro, retirando estes obstáculos, não é possível cumprir os objetivos e avançar de nível. A Figura 19 apresenta um nível onde é preciso utilizar a aprendizagem colaborativa.

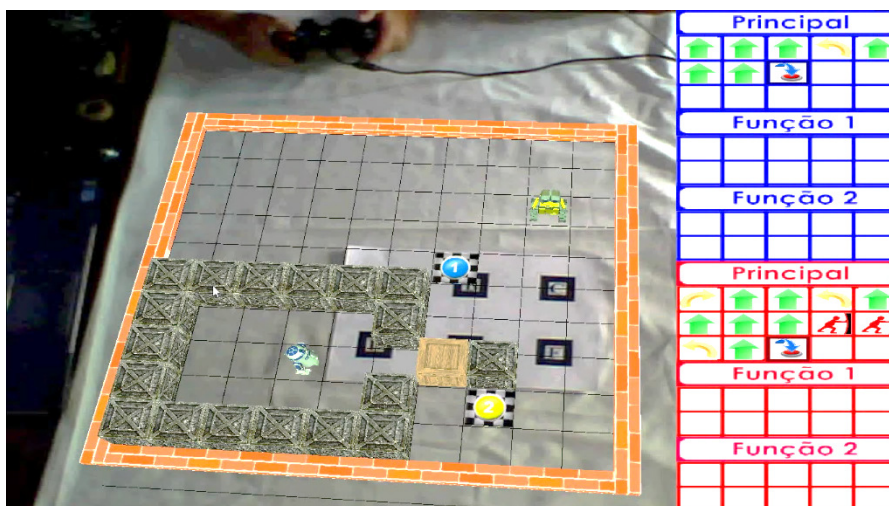


Figura 19 – Nível onde é utilizada a aprendizagem colaborativa

No nível apresentado na Figura 19, para cumprir os objetivos é preciso que o jogador 2 (representado pelo robô amarelo) empurre o obstáculo móvel que está à frente do personagem do jogador 1 (representado pelo robô azul), para que o jogador 1 possa movimentar seu personagem e cumprir os objetivos. Caso o jogador 2 não ajude o jogador 1 e apenas cumpra seus objetivos, indo pelo outro caminho, o nível não é concluído, fazendo com que os jogadores tenham que programar seus personagens novamente. Se os jogadores seguirem os programas criados na Figura 19, e o segundo jogador executar seu programa primeiro, o nível será concluído com sucesso.

5. CONCLUSÕES

O sistema desenvolvido proporcionou uma nova maneira de se lidar com o ensino de lógica algorítmica para alunos de ensino fundamental e médio. Verificou-se que a RA é uma ferramenta poderosa e inovadora, que permite gerar um ambiente motivador ao aluno.

A flexibilização na geração dos desafios cognitivos viabiliza a personalização e adequação dos mesmos ao público alvo, a saber, alunos do ensino fundamental e médio.

Além da personalização dos níveis, a utilização de dois personagens sendo controlados por dois jogadores, aumenta ainda mais as possibilidades de desafios cognitivos a serem criados. As combinações de comandos entre os dois jogadores podem ser complexas, viabilizando a aprendizagem colaborativa.

Utilizando-se bibliotecas de desenvolvimento gratuitas é possível gerar uma cena aumentada rica em detalhes, dinâmica e com ótimo desempenho.

Verificou-se, também, que a interação entre a cena gerada pelo sistema e o usuário por meio de controles de videogame, foi uma boa escolha. Os controles possuem comandos que são executados com baixa latência, não utilizam muitos recursos do computador e são facilmente manuseados pelos alunos, pois são equipamentos que muitos deles já conhecem ou possuem em

suas casas.

Finalmente, como trabalhos futuros, pretende-se:

- Habilitar novos desafios, tanto para o modo de um jogador, quanto para o de dois jogadores;
- Disponibilizar novos comandos para programação os personagens;
- Adicionar novos obstáculos, utilizando-se modelos tridimensionais mais complexos;
- Integrar os robôs virtuais já presentes no sistema, com robôs reais. Essa integração faria com que o usuário pudesse programar tanto robôs virtuais quanto robôs reais, aumentando as possibilidades de geração de desafios cognitivos; e
- Integrar outros objetos reais ao ambiente aumentado. A ideia é fazer com que os usuários adicionem objetos reais na cena aumentada (sobre o tabuleiro virtual), para ajudar os personagens a cumprirem seus desafios. Um exemplo deste tipo de interação poderia ser uma régua real que o usuário posicionasse no ambiente real, mas que no ambiente aumentado se transformaria em uma ponte, através da qual o personagem poderia passar para cumprir determinado desafio.

AGRADECIMENTOS

Os autores agradecem o apoio financeiro da Pró-reitoria de Extensão Universitária (PROEX/UNESP – ID do projeto: 6624 / 6141 / 8376 / 13969).

REFERÊNCIAS

- AZUMA, R. T. A survey of Augmented Reality. In: Teleoperators and Virtual Environments, Vol. 6, No. 4, pp. 930-933, Aug. 1997.
- CONSULARO, L. A., COELHO, R. C., CALONEGO JR, N. Rastreamento Óptico para Sistemas de Realidade Virtual e Aumentada. In: KIRNER, C., SISCOUTTO, R. (Ed.), “Realidade Virtual e Aumentada: Conceitos, Projeto e Aplicações”. Pré-Simpósio IX Symposium on Virtual and Augmented Reality, pp. 22-51, 2007.
- DEDE, C. Theoretical perspectives influencing the use of information technology in teaching and learning. International Handbook of Information Technology in Primary and Secondary Education, Vol. 20, pp. 43-62. 2008.
- GASPARI, T., SEMENTILLE, A. C., KINJO, E. M., AKYAMA, D. T., NAZIMA, F. A., MARAR, J. F. FERASOLI FILHO, H. Augmented Reality and the Teaching of Mobile Robotic: Two Case Studies. In: Robocontrol 2012 - 5th Workshop in Applied Robotics and Automation, 2012.
- GONÇALVES, D. Z. S., LIMA, D. C. CORREIA, V. F. A utilização da Realidade Aumentada em atividades práticas/complementares como motivação para alunos da educação básica. XVI

Congresso Internacional ABED de Educação a Distância, 2010.

KATO, H. ARToolKit version 2.3. 2000. Disponível em: <[http://www.hitl.washington.edu/research/shared space/download/](http://www.hitl.washington.edu/research/shared_space/download/)> Acesso em: 18 nov. 2012.

KAUFMANN, H., SCHMALSTIEG, D., WAGNER, M. Construct3D: A Virtual Reality Application for Mathematics and Geometry Education. In: Education and Information Technologies, Kluwer Academic Publishers, pp. 263 – 276, 2000.

KAUFMANN, H., MEYER, B., Simulating educational physical experiments in augmented reality. In: ACM SIGGRAPH ASIA 2008 educators programme, pp. 3:1–3:8, 2008.

KIRNER, C., TORI, R. Fundamentos de realidade aumentada. In: TORI, R. KIRNER, C. SISCOOTTO, R. (Ed.), “Fundamentos e Tecnologia de Realidade Virtual e Aumentada”. Pré-Simpósio VIII Symposium on Virtual Reality, pp. 20-34, 2006.

LOOSER, J. et al. OSGART -- a pragmatic approach to MR. In: ISMAR 06: 5th IEEE and ACM International Symposium on Mixed and Augmented Reality. Santa Barbara, USA, 2006.

MOSSEL, A., SCHÖNAUER, C., GERSTWEILER, G., KAUFMANN, H. ARTiFICe - Augmented Reality Framework for Distributed Collaboration. In: The International Journal of Virtual Reality, Vol. 11, No. 3, pp. 1 – 7, 2012.

SMYSER, B.M. Active and Cooperative Learning, 1993.

WANG, R., QIAN, X. OpenSceneGraph 3.0: Beginner's Guide. Packt Publishing, Birmingham, England, 2010.