UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE CIÊNCIAS EXATAS E DA TERRA
DEPARTAMENTO DE INFORMÁTICA E MATEMÁTICA APLICADA
PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS E COMPUTAÇÃO

Um Framework para criação de jogos voltados para o ensino de lógica de programação

Tainá Jesus Medeiros

Natal – RN
Agosto/2014

Tainá Jesus Medeiros

Um Framework para criação de jogos voltados para o ensino de lógica de programação

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Sistemas e Computação do Departamento de Informática e Matemática Aplicada da Universidade Federal do Rio Grande do Norte como requisito parcial para a obtenção do grau de Mestre em Sistemas e Computação.

Linha de pesquisa:
Engenharia de Software

Orientador
Prof. Dr. Eduardo Henrique da Silva Aranha

PPgSC – PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS E COMPUTAÇÃO
DIMAp – DEPARTAMENTO DE INFORMÁTICA E MATEMÁTICA APLICADA
CCET – CENTRO DE CIÊNCIAS EXATAS E DA TERRA
UFRN – UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE

Natal – RN
Agosto/2014

Catalogação da Publicação na Fonte
Universidade Federal do Rio Grande do Norte - UFRN

Medeiros, Tainá Jesus.
Um framework para criação de jogos voltados para o ensino de lógica de programação / Tainá Jesus Medeiros. - Natal, 2014.
79f: il.

Orientador: Prof. Dr. Eduardo Aranha.
Dissertação (Mestrado) - Universidade Federal do Rio Grande do Norte. Centro de Ciências Exatas e da Terra. Programa de Pós-Graduação em Sistemas e Computação.

1. Ensino de programação - Dissertação. 2. Framework - Engenharia de software - Dissertação. 3. Jogos digitais - Dissertação. I. Aranha, Eduardo. II. Título.

RN/UF/BSE04 CDU 004.41

Dissertação de Mestrado sob o título *Um Framework para criação de jogos voltados para o ensino de lógica de programação* apresentada por Tainá Jesus Medeiros e aceita pelo Programa de Pós-Graduação em Sistemas e Computação do Departamento de Informática e Matemática Aplicada da Universidade Federal do Rio Grande do Norte, sendo aprovada por todos os membros da banca examinadora abaixo especificada:

Prof. Dr. Eduardo Henrique da Silva Aranha
DIMAp – Departamento de Informática e Matemática Aplicada
UFRN – Universidade Federal do Rio Grande do Norte

Prof. Dr. Andre Mauricio Cunha Campos
DIMAp – Departamento de Informática e Matemática Aplicada
UFRN – Universidade Federal do Rio Grande do Norte

Prof. Dr. Leonardo Cunha de Miranda
DIMAp – Departamento de Informática e Matemática Aplicada
UFRN – Universidade Federal do Rio Grande do Norte

Prof. Dr. André Luís de Medeiros Santos
CIn – Centro de Informática
UFPE – Universidade Federal de Pernambuco

Natal, 29 de Agosto de 2014

Agradecimentos

Aos meus pais, Ednaldo (in memorian) e Gloria que, com muito carinho e apoio, não mediram esforços para que eu chegasse até esta etapa de minha vida.

Aos meus irmãos, Ednaldo Filho e Felipo, pelo carinho e força que me dão, por estarmos sempre juntos nos momentos mais importantes.

Aos meus amados sobrinhos Gabriel, Sophia, Gisele, Samuel e Guilherme por me darem uma completa alegria com suas energias e brincadeiras tirando qualquer estresse durante esse processo. Vocês são os meus tesouros.

Ao meu orientador, Professor Dr..Eduardo Aranha, pelo apoio, ensinamentos, incentivo, confiança, dedicação e orientação deste trabalho, o que me conduziu a um grande enriquecimento profissional e intelectual.

A todos os professores e a coordenação do DIMAp, que colaboraram de maneira direta ou indireta, para o meu crescimento acadêmico e profissional, ao longo destes dois anos.

Aos meus amigos da Hissatsu que me acolheram quando cheguei em Natal, me proporcionando uma grande família que sempre estava ali me apoiando e brigando quando necessário.

Ao laboratório da GamEdu que me proporcionou oportunidades de aprender e contribuir com vários projetos e pessoas.

Aos amigos, em especial, Pedrina, Moacir, Rafaella e Ana Gabriela, pelo incentivo e pelo apoio constante em qualquer etapa da minha vida.

Por fim, a meu marido Handerson, que sempre esteve ao meu lado me mantendo sempre firme e no foco nos momentos mais difíceis. Obrigada por toda sua paciência, carinho, apoio, incentivo e amizade. Simplesmente, te amo.

"Que os vossos esforços desafiem as impossibilidades, lembrai-vos de que as grandes coisas do homem foram conquistadas do que parecia impossível."

Charles Chaplin

Um Framework para criação de jogos voltados para o ensino de lógica de programação

Autora: Tainá Jesus Medeiros
Orientador: Prof. Dr. Eduardo Henrique da Silva Aranha

RESUMO

Os jogos digitais vêm sendo utilizados como ferramenta de auxilio à transmissão de conhecimento, permitindo difundir o conteúdo mais rapidamente. Utilizar essa estratégia para disseminar o desenvolvimento do raciocínio lógico para crianças do nível básico pode ser a engrenagem motivadora que ajuda no processo de aprendizagem para qualquer área. Pensando nisso, diversos jogos podem ser criados e disponibilizados para o uso do professor e aluno, porém, a complexidade de construção desses jogos torna-se um percalço que inviabiliza, muitas das vezes, sua construção. Neste contexto, este artigo apresenta um framework para criação de jogos voltados para o ensino de lógica de programação, apresentando a concepção, desenvolvimento através da integração do ambiente visual de programação Blockly com cenários criados em HTML5.

Palavras-chave: Jogos Digitais, Ensino de Programação, Framework.

A Framework for creating games aimed at teaching logic programming

Authoress: Tainá Jesus Medeiros
Supervisor: Prof. Dr. Eduardo Henrique da Silva Aranha Aranha

ABSTRACT

Digital games have been used as aiding tool for transmission of knowledge, allowing faster dissemination of content. Using this strategy of disseminating logical reasoning development for basic school children can be the motivating gear that helps in the learning process for any area. In this context, many games can be created and provided for the use of teacher and student. However, the complexity of construction of these games becomes a obstacle which can, often, prevent their construction. Thus, this paper presents a framework for creating games, which teach programming logic, presenting from their conception to their integration with the visual programming environment (Blockly) and scenarios created in HTML5.

*Keyword:*Digital Games, Teaching Programming, Framework.

Lista de figuras

Lista de tabelas

Lista de abreviaturas e siglas

API – Application Programming Interface

CSS – Cascading Style Sheets

DIMAp – Departamento de Informática e Matemática Aplicada

FURB – Universidade Regional de Blumenau

GML – Game Maker Language

GQM – Goal / Question / Metric

HTML 5 – Hypertext Markup Language, versão 5

IBCD – Índice Brasscom de Convergência Digital

IESPB – Instituto de Educação Superior da Paraíba

JS – JavaScript

KDE – K Desktop Environment

LEGO – Leg Godt

MIT – Massachusetts Institute of Technology

PUC/PR – Pontifícia Universidade Católica do Paraná

RBIE – Revista Brasileira de Informática na Educação

RENOTE – Revista Novas Tecnologias na Educação

RITA – Revista Informática na Educação: teoria e prática

RSL – Revisão Sistemática da Literatura

SBIE – Simpósio Brasileiro de Informática na Educação

UFF – Universidade Federal Fluminense

UFPB – Universidade Federal da Paraíba

UFRJ – Universidade Federal do Rio de Janeiro

UFRN – Universidade Federal do Rio Grande do Norte

UFRPE – Universidade Federal Rural de Pernambuco

ULBRA – Universidade Luterana do Brasil

UPE – Universidade de Pernambuco

USP – Universidade de São Paulo

WEI – Workshop de Educação em Computação

WIE – Workshop de Informática na Escola

WYSIWYG – What You See Is What You Get

XML – Extensible Markup Language

Sumário

1 Introdução

Professores de diversas disciplinas vêm demonstrando interesse pelo uso dos jogos na escola, pois esses recursos têm o potencial de trazer benefícios para os processos de ensino e aprendizagem (Johnson, 2006; Prensky, 2006, 2007; Gee, 2007). Os jogos conseguem provocar o interesse e a motivação dos estudantes com desafios, curiosidade, interação e fantasia (Balasubramanian e Wilson, 2006). Proporcionam uma experiência estética visual e espacial rica e, com isso, são capazes de seduzir os jogadores e atraí-los para dentro de mundos fictícios que despertam sentimentos de aventura e prazer (Mitchell e Savillsmith, 2004). Assim, o estudante pode assumir um papel em um jogo, enfrentando os problemas reais da vida desse profissional. Facilitando também a assimilação do conteúdo de diversas disciplinas.

Neste cenário as tecnologias digitais podem ser importantes aliadas na busca de soluções para o problema da aprendizagem nas Escolas, inclusive no ensino de programação. Aprender a programar ensina as pessoas a como pensar por meio do pensamento lógico e criativo. Segundo a comunidade empresarial (Giroto, 2014), "O pensamento criativo é a parte mais importante desse aprendizado. Em segundo lugar vem a parte de raciocinar de forma sistemática, e depois como trabalhar de forma colaborativa. Essas três habilidades faltam muito nos profissionais de qualquer área e a programação ensina tudo isso."

Por estimular o pensamento lógico, a criatividade e a capacidade de resolução de problemas, o estudo da programação vem sendo incentivado por diversas iniciativas internacionais, como a Code.org. Alguns especialistas discutem as vantagens de ensinar código a crianças, e defende o ensino da ciência da computação ao lado de outros cursos elementares, como matemática, biologia, física e química (Code.org, 2013).

Com a popularização dos computadores, tablets e smartphones, as crianças tendem a estar em contato com esses tipos de equipamentos a partir de seus primeiros anos de vida. Eles aprendem rápido como usá-los, mas raramente aprendem como eles funcionam. Há uma crescente ideia entre os educadores que defendem que esta lacuna precisa ser preenchida ensinando-se programação às crianças. Isto porque aprender programação é uma forma de conseguir se relacionar melhor com as máquinas (Giroto,

2014). A grande questão é que aprender a codificar não é uma fácil tarefa, especialmente para as crianças.

Nesse ponto, as tecnologias digitais podem ser importantes aliadas, especialmente os jogos digitais, que inserem o estudante em cenários lúdicos que muitas vezes simulam problemas reais, sendo assim recursos mais atrativos para os estudantes. Segundo Medeiros (2013), a proposta de utilizar jogos digitais para o ensino de programação deve ser focada no ensino médio.

De fato, a Sociedade Brasileira de Computação (SBC) entende que os conceitos da Computação devem ser ensinados a partir do ensino básico e incentivam ações dessa natureza, na premissa de que essa área apresente princípios e habilidades que, se trabalhados com os estudantes desde cedo, podem contribuir para o exercício da lógica e resolução de problemas, assim como fomentar o interesse pela área, aumentando o número de profissionais no país (Melo, et.al, 2013). O mercado, por outro lado, já apresenta sinais dessa tendência mais presente em países como os Estados Unidos (Aguiar, 2014).

Sabendo desse déficit e da importância de se aprender lógica de programação, precisamos elaborar técnicas e metodologias que façam com que crianças se interessem em estudar essas áreas. Nesta perspectiva este trabalho apresenta um framework que facilita o desenvolvimento de jogos que ensinem lógica de programação, visando contribuir para o ensino de programação nas escolas do ensino básico, de uma forma contextualizada e divertida.

1.1 Justificativa

Várias são as dificuldades enfrentadas pelos alunos durante o processo de ensino-aprendizagem de programação. Existem várias possibilidades de origem destas dificuldades, como destaca Raabe e Silva (2005), seja pela exigência lógico-matemático predominante na disciplina, ou mesmo pela dificuldade de apreensão, por parte do professor ou até mesmo pelo ritmo de aprendizagem de cada aluno.

A utilização de jogos digitais como ferramenta de auxílio à transmissão de conhecimento permite difundir o conteúdo mais rapidamente. Pensando nisso, diversos

jogos podem ser criados e disponibilizados para o uso do professor e aluno, porém, a complexidade de construção desses ambientes torna-se um percalço que inviabiliza, muitas das vezes, sua construção.

E nesta perspectiva de que existe a necessidade da sociedade de ter jogos educativos que ensinem lógica de programação, este trabalho apresenta o ensino de lógica de programação através de jogos digitais, ou seja, um ensino contextualizado, lúdico e atrativo. Para isto, foi desenvolvido um framework para criação de jogos com essas características fazendo uso de dois ambientes, o Construct 2 e o Blockly.

1.2 Objetivos

1.2.1 Objetivo geral

Este trabalho tem por finalidade desenvolver um framework para criação de jogos digitais voltados para o ensino de lógica de programação.

1.2.2 Objetivos Específicos

Podemos citar como objetivos específicos desse trabalho:

- Pesquisar e apresentar conceitos e características de jogos voltados para o ensino de lógica de programação;
- Compreender algumas ferramentas mais utilizadas no mercado para desenvolver jogos digitais;
- Realizar uma revisão sistemática sobre o ensino de programação através de jogos digitais;
- Desenvolver um mecanismo para criar jogos digitais voltados para o ensino de lógica de programação;
- Permitir o desenvolvimento de jogos, novas pesquisas e novos trabalhos científicos através do framework;

- Fornecer respostas sobre o uso de jogos digitais voltados para a educação;

1.3 Estrutura do Trabalho

O presente trabalho está organizado conforme se descreve a seguir.

Esta introdurabalho está organizado conforme se descreve a seguir. ao trabalho e descrever seu objetivo geral e especcreve a

O caprabalho e descrever seu escri geral e especcreve ar seu objetivo objetivo os atrav apeccreve aork; o Conslsinam balprogramae ho

O Capamae ho e descrever seu escri geral e especcreve ar seu objetivo objetivo os atrav apeccreve aork; o Construct 2 mpo e no

O Capamae ho e descrever seu e resultados de uma revisar seu objetivo objetivo os atrav apeccreve aork; o Construct 2

O capamae ho e descrever seu e resultados de uma revisar seu objetivo objetivo os atrav apeccreve aork; o Construct 2 mpo e no

No capitulo seis serever seu e resulultados obtidos atravseu objetivo objetivo

No capitulo sete apresenta alguns trabalhos relacionados.

Logo após, no oitavo e último capítulo será fornecido a conclusõonclusós, no oitavo e , os artigos publicados através deste trabalho, como tambs a equipe que colaborou com o desenvolvimento do framework.

2 Jogos educacionais voltados para o ensino de programação

Os jogos digitais voltados para o ensino são atividades inovadoras onde as características do processo de ensino-aprendizagem apoiado no computador e as estratégias de jogo são integradas a fim de alcançar um objetivo educacional determinado.

Todo jogo acontece em um tempo e espaço. Ele é o meio que auxilia na concretização de determinados objetivos e promove o domínio do conhecimento. Os jogos consistem numa simples assimilação funcional, num exercício de ações individuais já aprendidas. Geram ainda sentimento de prazer e domínio das ações. (Rosa, 2011)

Esta estratégia, num jogo planejado adequadamente, promove o interesse e a motivação que por sua vez, aumentam a atenção do aluno e criam a sensação de que aprender é divertido, proporcionando ao jogador desenvolver a capacidade de processar fatos e fazer inferências lógicas durante a resolução de um problema (Morati, 2003).

Como benefícios que os jogos digitais educacionais podem trazer aos processos de ensino e aprendizagem, temos: efeito motivador, facilitador do aprendizado, desenvolvimento de habilidades cognitivas, aprendizado por descoberta, experiência de novas identidades, socialização, coordenação motora e comportamento expert.

Mas para serem utilizados com fins educacionais os jogos precisam ter objetivos de aprendizagem bem definidos e ensinar conteúdos das disciplinas aos usuários, ou então, promover o desenvolvimento de estratégias ou habilidades importantes para ampliar a capacidade cognitiva e intelectual dos alunos (Gros, 2003).

Porém, muitos jogos são lançados com o intuito de ter um foco educacional, e por tal fato possivelmente serem incluídos nas escolas. O resultado é que muitas vezes os pais, educadores, diretores e dentre tantos outros “personagens” do contexto escolar, saem à procura de games que possam realmente educar (Medeiros, 2014).

Nessa procura, muitos acabam se perguntando por que não pode existir um jogo educacional que seja tão divertido como um Counter Strike, Call Of Duty ou um World Of Warcraft. Simplesmente pelo fato de que, muito dos jogos tidos como educacionais que temos hoje em dia são chatos e tediosos. (Mendes, 2012)

Sobre esse processo de inserção de jogos no contexto educacional, Gilson Schwartz, líder do grupo de pesquisa da Cidade do Conhecimento, na USP, em São Paulo, e dono da empresa Iconomia Produções Culturais, relata que para que a iniciativa dê certo, os jogos educativos precisam agradar principalmente seu público-alvo, no caso, os alunos. "O game não pode ser chato. Existem muitos jogos educacionais que são chatos porque repete a mesma lógica do professor dentro da sala de aula, cobrando do aluno uma resposta".

Vimos que os jogos têm poder educacional e são fortes aliados na construção do conhecimento. Mas considerar a aplicação dos games que os jovens gostam, inclusive os agressivos, em salas de aula ainda é uma questão recente e polêmica no mundo pedagógico. Segundo Roger Tavares, especialista no assunto, todos os tipos de jogos podem ser trabalhados para atingir um objetivo didático. "Quanto mais os educadores utilizarem ferramentas educacionais presentes no cotidiano dos alunos, melhores serão os resultados obtidos", explica Tavares que, além de transformar e criar jogos, também pesquisa os velhos e bons tabuleiros para tentar adaptá-los ao universo tecnológico (Senac, 2014).

Mas nem sempre os games e escolas conseguem trabalhar juntos. Alguns professores recriminam os jogos eletrônicos nas aulas, pois eles não apresentam elementos que podem ser utilizados como ferramentas pedagógicas. Mas o fortalecimento dos "serious games", os jogos sérios e educativos, pode finalmente levar os games para a escola e, claro, complementar o ensino dos estudantes (Petró, 2010). Pois sabemos que os jogos digitais podem ser utilizados em diferentes níveis de ensino, indo desde a préescola até cursos de graduação, especializações e cursos corporativos.

2.1 Exemplos de jogos que ensinam lógica de programação

No mercado existem alguns jogos que tentam ensinar de forma lúdica, sobre lógica de programação. Entenda-se lógica de programação como sendo programação nada mais é do que a organização coerente das instruções do programa para que seu objetivo seja alcançado.

Podemos ilustrar algumas desses jogos, como por exemplo o Light-bot, um jogo de desafios de programação: um jogo de desafios que usa a mecânica do jogo fortemente baseada em conceitos de programação.

O Light-bot possibilita que os jogadores ganhem prática e entendimento dos conceitos básicos de controle de fluxo, tais como procedimentos, laços e condições, apenas dirigindo um robô com comandos para acender os ladrilhos do chão, como visto na Figura 1.

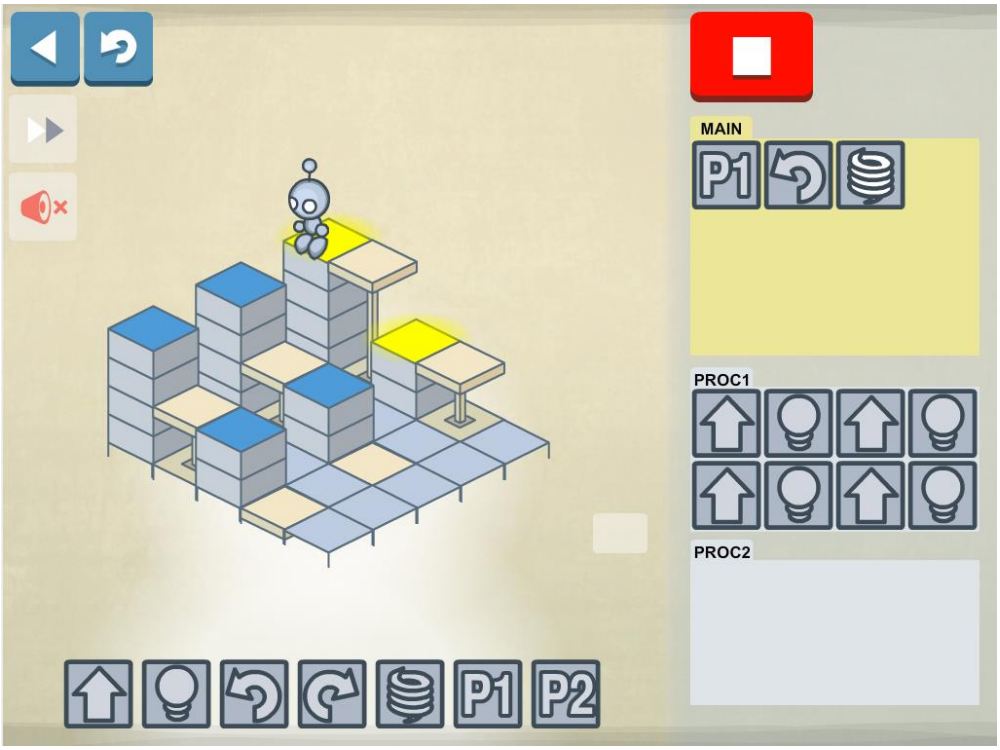

Figura 1: Interface do Light-bot.

A Microsoft criou um jogo para navegadores que tem o objetivo de ensinar programação. A premissa do game, chamado Code Hunt, requer que o jogador escreva os códigos necessários corretamente para avançar no jogo.

O game introduz o jogador às linguagens Java e C#. Para tentar ensinar a arte da programação, o Code Hunt usa quebra-cabeças. O jogador precisa explorá-los por meio de pistas, e entender o código necessário para "capturá-lo". A pontuação é oferecida com base na elegância da solução encontrada.

O game apresenta um código defeituoso e algumas orientações de qual deve ser o resultado daquele programa quando funcionando corretamente. Cabe ao jogador corrigi-lo adequadamente, usando o menor número de linhas, para aumentar a pontuação. Assim, o jogo tenta ensinar programação para solução de problemas. (Figura 2)

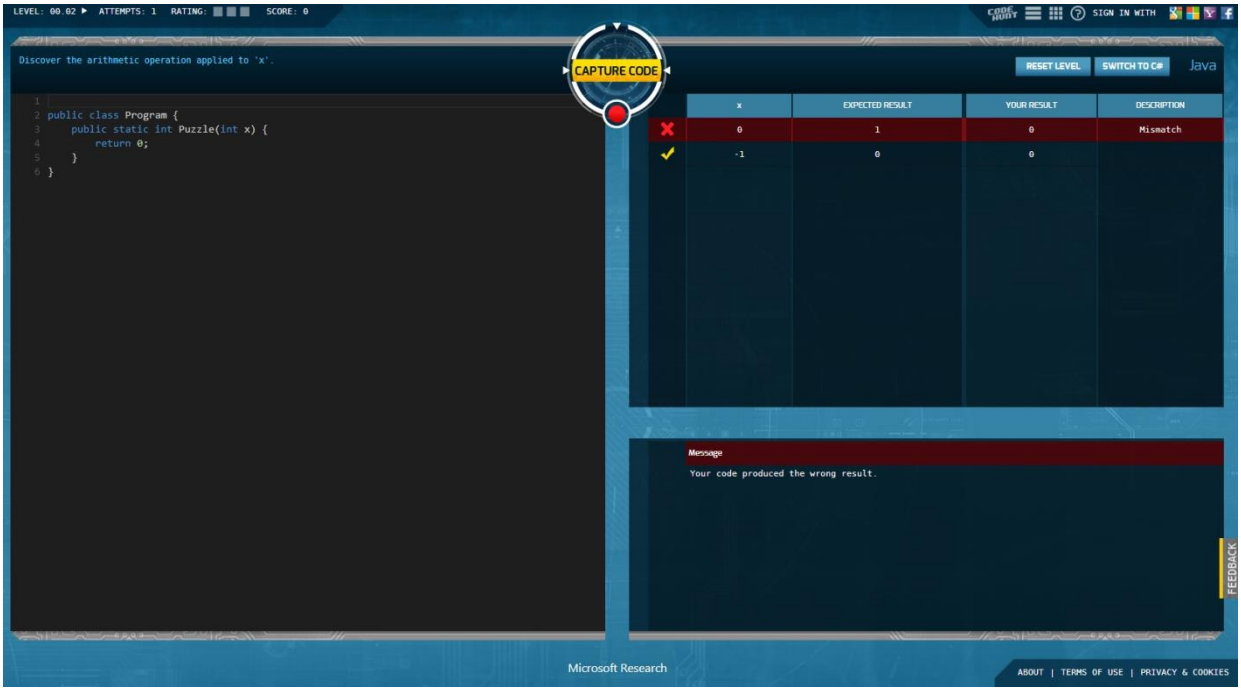

Figura 2: Interface do Code Hunt.

Ainda podemos citar o jogo KLogo-Turtle que é um interpretador da linguagem LOGO para KDE. O KLogo-Turtle é uma útil ferramenta para o ensinamento de geometria e os princípios básicos de programação de computadores (Figura 3).

No Klogo-Turtle podemos movimentar um cursor no campo gráfico para gerar desenhos. Os movimentos são gerados com comandos da linguagem de programação e descritos no campo de comandos. Existem basicamente quatro comandos para fazer a movimentação de tartaruga, que são:

- Frente e atrás – fazem a tartaruga andar
- Direita e esquerda – fazem a tartaruga girar em torno de seu próprio eixo.

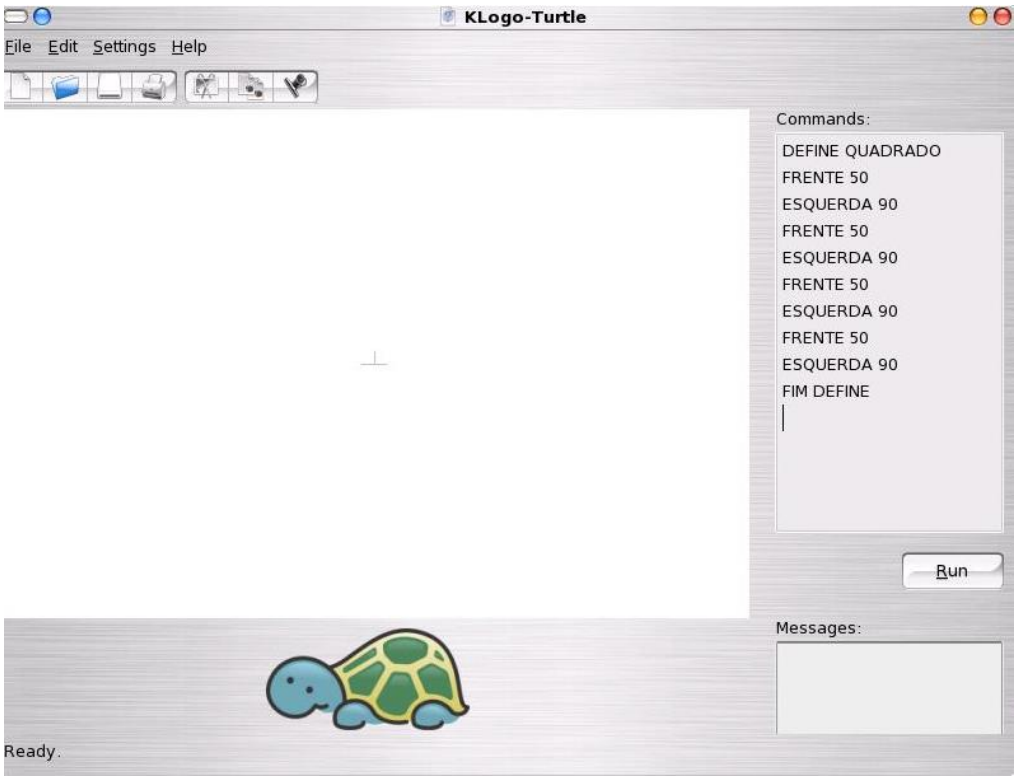

Figura 3: Interface do Klogo-Turtle.

Por fim, temos o Blockly, um ambiente baseado na web para edição de programação de uma forma gráfica. Os usuários podem arrastar os blocos juntos para construir um aplicativo, assim como se arrasta blocos de montar (LEGO). O Blockly é um de vários ambientes de programação visuais crescentes no mercado. Muitos desses ambientes têm sua criação no MIT, tendo uma aparência e comportamento semelhantes em diferentes produtos.

O blockly funciona como um quebra-cabeças, é só ir encaixando as "peças" que realizarão as funções desejadas, como pode ser visto na Figura 4 e Figura 5.

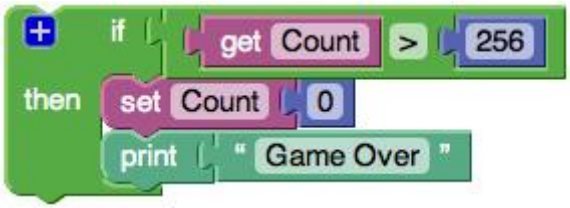

Figura 4: Bloco de comandos do Blockly.

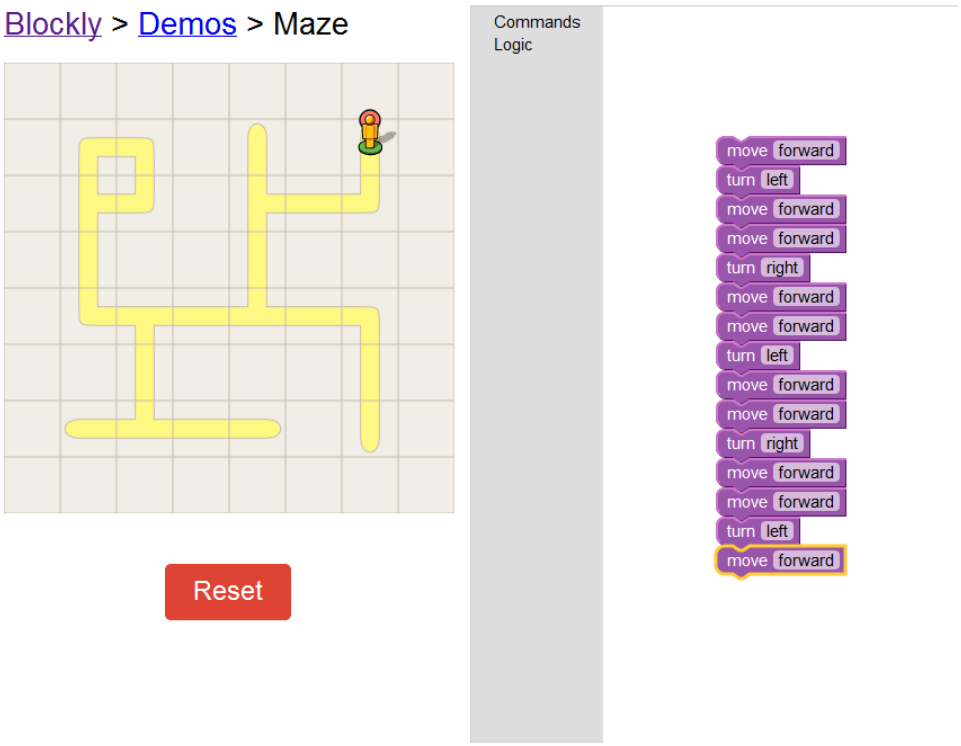

Figura 5: Exemplo do funcionamento do Blockly

As características principais Blockly são:

- Executar em um web-browser. Não há downloads ou plugins necessários.
- Código exportável. Os usuários podem extrair seus programas como JavaScript, Dart, Python ou outra linguagem, para que quando eles superarem o Blockly eles podem continuar a aprender, como pode ser visto pela Figura 6
- É possível baixar o código em XML e abrir novamente este código parametrizado.
- Open source. Tudo sobre Blockly está aberta: você pode modificar, baixar, e usar em seus próprios sites.
- Altamente capaz. Com a capacidade de calcular o desvio padrão usando um único bloco, o Blockly não é um brinquedo.

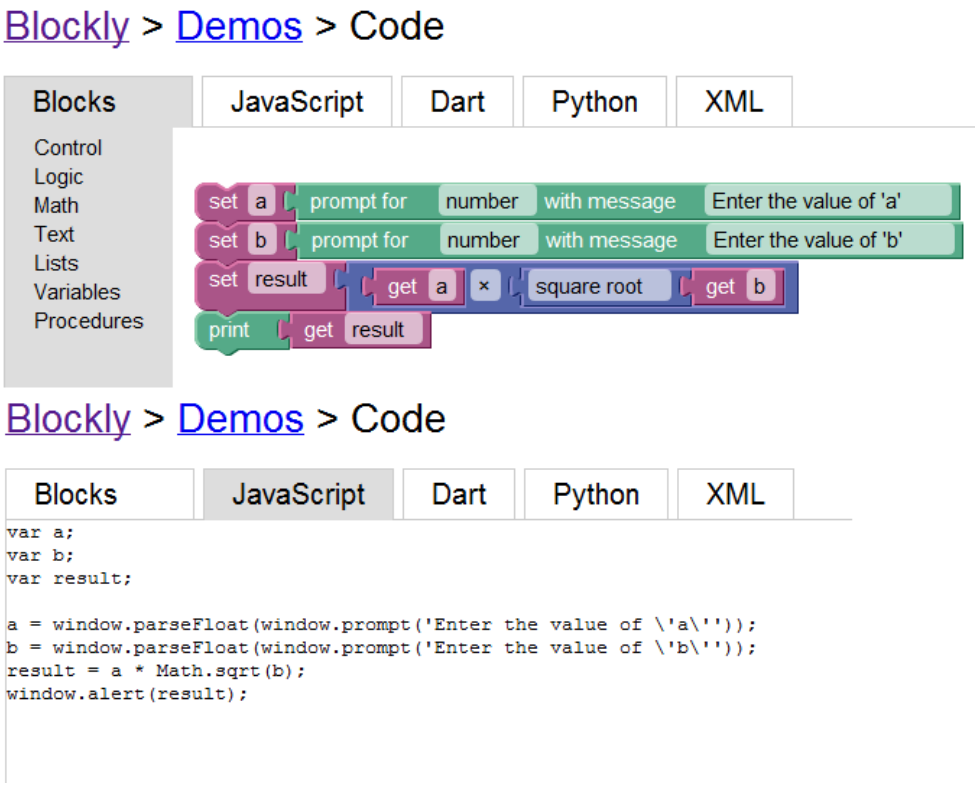

Figura 6: Código gerado pelo Blockly

O Blockly em si não é uma plataforma educacional. Ele é um editor que podem ser utilizados como parte de uma plataforma educacional, ou como parte de um conjunto de negócios, ou como parte de um sistema de jogos.

2.2 Considerações Finais

Neste capitulo vimos que a utilização alguns jogos existentes no mercado que podem servir de forma inovadora e incentivadora de ensino. Podendo ter uma forma de avaliar o aluno através dos jogos e o quanto isso pode ser uma ferramenta de estimulo para o aluno testar seus conhecimentos e melhora-lo.

No próximo capitulo veremos as ferramentas de criação de jogos mais utilizadas no mercado e que foram estudadas como possível ferramenta para fazer parte do framework que foi desenvolvido.

3 Tecnologias de desenvolvimento de jogos

Cada vez mais ferramentas disponíveis para o desenvolvimento de jogos vêm surgindo. A distribuição de jogos deixou de ser apenas privilégios de grandes empresas, hoje é uma opção para qualquer pessoa que queira empenhar em um projeto de jogo.

Esse capítulo apresenta algumas tecnologias mais simples e que estão sendo utilizadas pela academia, usadas para a construção de jogos digitais.

3.1 Construct 2

O Construct é um programa criado pela Sicrra que ajuda na criação de jogos digitais. Não é necessário codificar, pois tudo é feito de forma automática. O usuário apenas irá criar o enredo do jogo, o restante da aplicação é feita praticamente através do mouse, com a ação de arrastar e soltar os objetos no cenário principal. O Construct 2 é uma ferramenta de criação de jogos que exporta para HTML5, e possui um interface bem interessante.

A vantagem de utilizar o Construct 2 é que ele tem o conceito WYSIWYG (What You See Is What You Get - O que você vê é o que você tem). Você visualiza cada fase do jogo e consegue enxergar como ficará o resultado final em tempo de desenvolvimento, sem a necessidade de fazer códigos.

A programação é visual e utiliza o conceito de eventos, onde é possível adicionar condições, sub-condições e ações sequenciais para executar. Ele já vem nativamente preparado para interações com mouse, touch e teclado, basta apenas adicionar o recurso ao projeto e criar eventos para eles. A programação é simplificada e objetiva, permitindo integração com Facebook, Android e iPhone. É rápido, permite muitas possibilidades, possui documentação bem escrita, é estável e gera um produto final com tamanho pequeno, dependendo mais do tamanho das imagens utilizadas e da quantidade de fases criadas (Dadrix, 2012).

3.2 Scratch

Scratch é uma linguagem de programação desenvolvida por Lifelong Kindergarten Group no Media Lab, MIT. Este aplicativo possibilita a criação de histórias, animações, jogos e outras produções, bem como o compartilhamento das criações na Web. Tudo pode ser feito a partir de comandos que devem ser agrupados de modo lógico. A programação é efetuada através da criação de sequencias de comandos simples, que correspondem a blocos de várias categorias, encaixados e encadeados de forma a produzirem as ações desejadas. (MIT Media Lab, 2013)

As principais estruturas lógicas que o Scratch trabalha são as estruturas de controle. Por exemplo, com essas estruturas é possível determinar se um Sprite deve repetir um número de vezes a ação, ou ainda, verificar se algo for verdadeiro fazer uma rotina, senão fazer uma outra rotina. É possível também, criar uma ação para a tecla quando for pressionada (MIT Media Lab, 2013). A figura 7 apresenta essas estruturas de controle.

Figura 7: Estruturas de controle do Scratch

No scratch é possível mudar, mover ou movimentar um Sprite, para isso acontecer basta utilizar as estruturas de movimento. A figura 8 ilustra as estruturas de movimento.

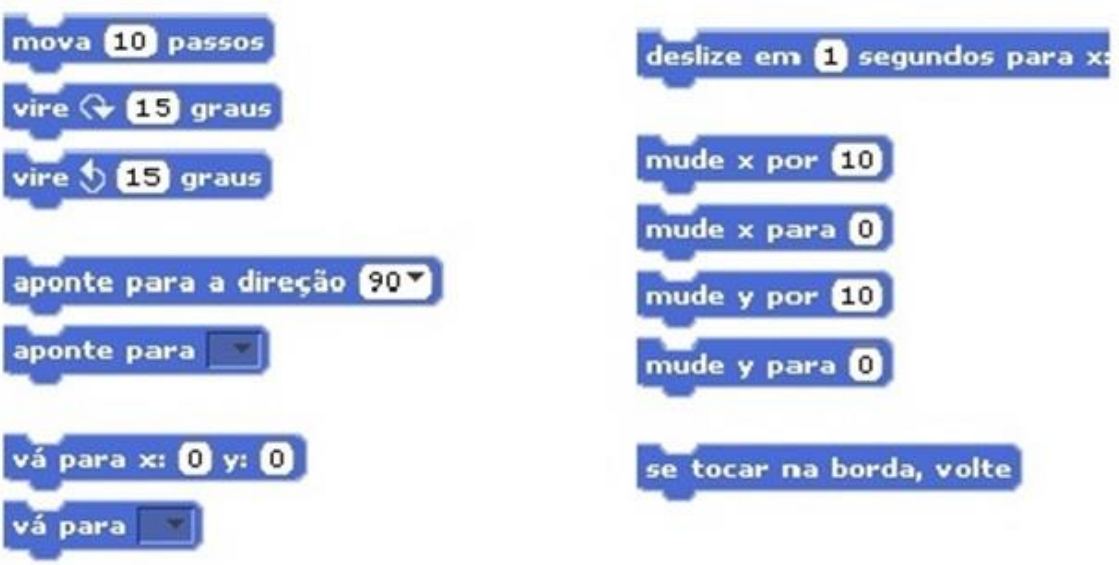

Figura 8: Estruturas de movimento do Scratch

O Scratch é muito mais acessível que outras linguagens de programação, por se utilizar de uma interface gráfica que permite que programas sejam montados como blocos de montar, lembrando o brinquedo LEGO. Utiliza uma sintaxe comum a muitas linguagens de programação. E diferente de outras linguagens, não tem nenhum tipo de pontuação obscura. Cada bloco da linguagem contém um comando em separado, que podem ser agrupados livremente caso se encaixem. E os comandos podem ser modificados através de menus (MIT Media Lab, 2013).

3.3 Alice

O Alice é um ambiente de programação que proporciona a usuários com pouca ou nenhuma experiência de computação possam programar caracteres ou objetos em um mundo virtual, de forma muito parecida a um moderno filme ou a um videogame. Assim como no mundo real, o mundo virtual do Alice é tridimensional em função do tempo, e cada objeto é dotado de propriedades da mesma forma que objetos físicos, tais como cor, tamanho, localização e assim por diante. Dentre os seus objetos, este ambiente de programação possui um diferenciado, uma câmera, que permite exibir o mundo virtual na tela do computador, oportunizando visualizar o que está acontecendo no mundo virtual quando o programa está em execução (Shelly et al., 2007).

Na interface interativa do Alice, estudantes arrastam tópicos gráficos, onde existem instruções correspondentes a comandos padrões na produção de programação orientada a objetos, tal como ocorre com Java, C++ ou C# (Figura 9). Dessa forma, o aplicativo Alice permite que os estudantes vejam imediatamente como os comandos de seus programas de animação funcionam, habilitando-os facilmente a compreenderem as relações entre os comandos de um programa e o comportamento dos objetos nas suas animações. Pela manipulação dos objetos em seu mundo virtual, os estudantes adquirem experiência com todas as típicas construções de programação ensinadas em um curso de programação, dominando suas aplicações e compreendendo suas potencialidades (Barros et al., 2004).

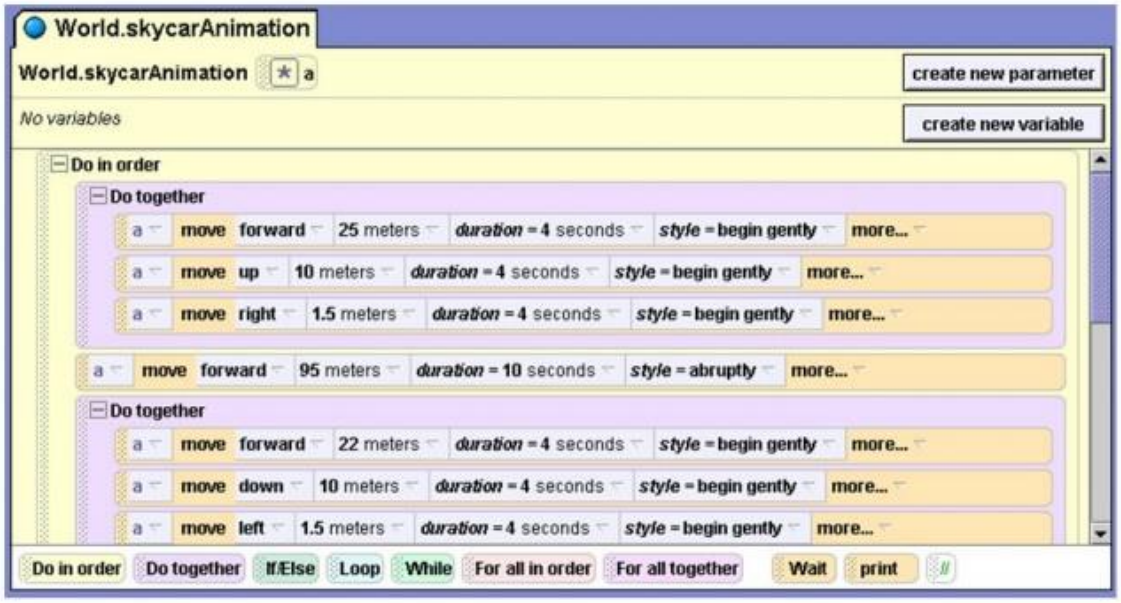

Figura 9: Edição de um texto fonte na interface do ALICE

3.4 Game Maker

O Game Maker é um software voltado à criação de jogos escrito em Delphi. Originalmente o projeto se chamava "Animo", uma ferramenta para criação gráfica que posteriormente evoluiria para um motor de criação de jogos direcionado à desenvolvedores inexperientes (Figura 10).

O Game Maker é um motor (Engine) desenvolvido para facilitar a criação de jogos de qualquer tipo, com um sistema de desenvolvimento "Drag and Drop" (arraste e solte), que consiste em arranjar ícones pré-definidos em uma ordem lógica a fim de

executar uma tarefa. Este motor também permite a criação de jogos 3D simples, porém para criar jogos mais complexos é necessário dominar a linguagem do motor, denominada GML (Game Maker Language) (Yoyogames, 2013).

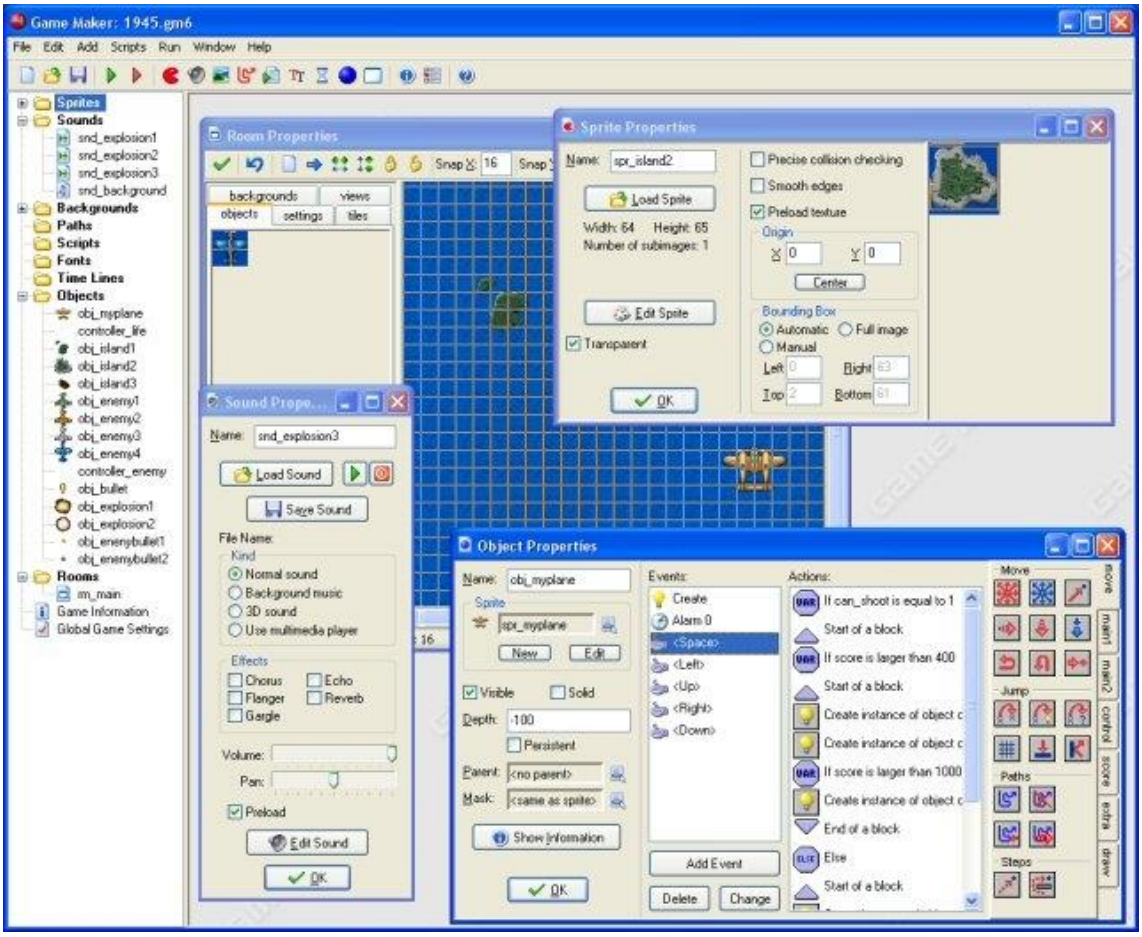

Figura 10: Edição de um texto fonte na interface do GAME MAKER

3.5 Considerações Finais

Após o estudo e análise dessas ferramentas, optamos por usar a ferramenta Contruct 2 para ser integrada junto ao ambiente de programação visual Blockly que foi descrito no capitulo anterior.

O construct 2 por ser mais simples e por poder exporta os projetos para a linguagem HTML5. E a escolha do Blockly se deu pela simplicidade e potencial de extensão da ferramenta.

No próximo capitulo vamos apresentar uma revisão sistemática realizada com o objetivo de avaliar como está o processo de ensino-aprendizagem de programação através de jogos digitais.

4 Revisão Sistemática

Há uma clara necessidade de se analisar a produção científica nacional sobre o processo de ensino-aprendizagem de programação através de jogos digitais. Para alcançar este objetivo, neste trabalho iremos apresentar os resultados de uma Revisão Sistemática da Literatura (RSL) que contou com a análise dos artigos sobre o processo de ensino-aprendizagem de programação com a utilização de jogos digitais publicados no Brasil nos últimos 5 anos (2008-2012) em três importantes eventos nacionais na área de informática na educação, o Simpósio Brasileiro de Informática na Educação (SBIE), o Workshop de Informática na Escola (WIE), Workshop de Educação em Computação (WEI) e também em três relevantes revistas nacionais na área, a Revista Brasileira de Informática na Educação (RBIE), a Revista Novas Tecnologias na Educação (RENOTE) e a Revista Informática na educação: teoria e prática (RITA).

4.1 Método Utilizado

Segundo Kitchenham (2007), uma RSL emprega um processo metódico para identificar, avaliar e interpretar todas as evidências científicas disponíveis e relevantes relacionadas a um tema específico de interesse. O protocolo usado para realizar esta RSL foi baseado no trabalho de Kitchenham (2007).

4.1.1 Questões de Pesquisa

Esta RSL tem como questão central de pesquisa a seguinte pergunta: Qual o panorama atual do uso de jogos digitais no contexto do ensino de programação? Para responder a essa questão, foram definidas as seguintes questões de pesquisa:

- QP1: Como os jogos digitais estão sendo aplicados no ensino de programação?
- QP2: Qual público alvo está sendo focado nos trabalhos de pesquisa?

- QP3: Quais as ferramentas estão sendo empregadas?
- QP4: Quais são os efeitos observados nos alunos com a introdução de atividades de programação baseadas em jogos digitais?
- QP5: Quais teorias pedagógicas estão sendo utilizadas?
- QP6: Quais são as instituições de pesquisas envolvidas na área e como elas estão sendo distribuídas no Brasil?

4.1.2 Fontes de busca, processo de seleção e critério de inclusão e exclusão

A pesquisa foi realizada através de busca manual nos anais dos eventos SBIE, WIE e WEI nas revistas RENOTE, RBIE e RITA nos últimos 5 anos. Esta busca foi realizada em duas fases. A pré-seleção dos artigos consistiu em verificar os anais dos referidos eventos e revistas e acessar manualmente todos os artigos completos, lendo os seus títulos, resumos e palavras-chave, estabelecendo a primeira fase. Na segunda fase, todos os artigos pré-selecionados foram analisados e a cada um deles foram aplicados os critérios de inclusão e exclusão que são apresentados na Tabela 1.

Tabela 1: Critérios de exclusão e de inclusão.

Critérios de Inclusão	Critérios de Exclusão
Artigos completos que abordam o processo de ensino-aprendizagem de programação através de jogos digitais; e Artigos de ensino e programação baseados em robótica.	Artigos duplicados; Artigos que abordam o processo de ensino-aprendizagem de programação, sem a utilização de jogos; Artigos resumidos; e Artigos não relevantes (excluídos pelo título, resumo, palavras-chave não relacionados aos objetivos desta RSL).

A robótica educacional passou a ser uma ferramenta auxiliar para as metodologias de ensino que visam à construção da aprendizagem. Por isso, o uso de robótica para o ensino de programação está incluído nesta RSL.

4.1.3 Procedimento de distribuição e análise dos artigos

A partir da lista de publicações identificadas, os artigos são atribuídos de forma aleatória pelo pesquisador Doutor (R3) para dois outros pesquisadores, um aluno de mestrado (R1) e um aluno de doutorado (R2). Cada pesquisador avalia individualmente seus artigos e os resultados de R1 e R2 são integrados por R3 na tabela de Concordância/Discordância. Por fim, R3 julga e resolve as discordâncias na tabela, numa lista final de estudos avaliados.

4.1.4 Processo de extração de dados

Cada questão de pesquisa motiva a extração dos dados (Tabela 2). Além disso, foram extraídas informações gerais, como: ano da publicação, universidade, fonte da publicação, nome da ferramenta/jogo e quem usou. A extração de dados foi realizada através de planilhas contendo formas de extrair as informações dos artigos selecionados.

Tabela 2: Extração de dados.

Questões de Pesquisa	Atributo	Dados
QP1 e QP5	Processo	Etapas apoiadas pelo estudo.
QP5	Características	Teorema de ensino-aprendizagem para auxiliar no processo de aprendizagem de programação.
QP3	Ferramenta	Título do estudo, nome da ferramenta, onde foi utilizada (escola ou universidade).
QP2 e QP6	Mapeamento	Fonte de publicação, estado do primeiro autor e público alvo do estudo.

4.1.5 Avaliação da Qualidade

As questões de avaliação foram analisadas para obter uma pontuação final para cada artigo. O procedimento de pontuação das questões de avaliação foi de Y (sim) = 1, P (parcialmente) = 0.5 e N (não) = 0, conforme Kitchenham (2007). Os artigos com pontuação 0 foram retirados do estudo. Na RSL utilizamos seis questões de avaliação (QA), são elas:

- QA1. Foi utilizada alguma teoria pedagógica para auxiliar no processo de aprendizagem de programação?
- QA2. Foi utilizada alguma ferramenta existente no mercado para auxiliar no processo de ensino-aprendizagem de programação?
- QA3. Foi realizado algum tipo de experimento controlado ou estudo de caso para avaliação da metodologia proposta?
- QA4. O estudo avaliado apresenta uma breve comparação com outras metodologias relacionadas existentes?
- QA5. O estudo avaliado apresenta o endereço (URL) na Internet onde o jogo digital utilizado esteja disponível para uso ou para cópia (download)?
- QA6. O estudo avaliado foi aplicado em alguma escola/universidade?

4.2 Resultados

A busca manual resultou 46 artigos, dos quais 17 foram incluídos. A Tabela 3 apresenta o resultado geral da busca realizada. Para chegarmos nos 46 resultados iniciais, a etapa da seleção de estudo se deu pela leitura dos títulos e resumos. Posteriormente, foram analisados os critérios de inclusa e exclusão, resultando em 36 artigos, que foram completamente lidos, a fim de descartar os artigos irrelevantes para a nossa revisão. Após esta etapa da seleção de estudo, o número final de artigos relevantes resultou em 17 artigos.

Tabela 3: Resultado geral da busca.

Evento/Revista	Artigos pré-selecionados	Artigos incluídos	Total de artigos	Artigos incluídos/total de artigos (%)
SBIE	18	7	434	4,37
WIE	8	4	200	2
WEI	13	6	122	7,32
RBIE	4	0	96	0
RENOTE	3	0	584	0
RITA	0	0	0	0
TOTAL	**46**	**17**	**1505**	**13,69**

4.2.1Avaliação da qualidade da RL

A busca manual resultou 46 artigos, dos quais 17 foram incluídos. A Tabela 4 apresenta o resultado geral da busca realizada.

QP1: Como os jogos digitais estão sendo aplicados no ensino de programação?

É fundamental compreender o nível de apoio (estudo de caso e/ou experimento) realizado por cada estudo para validar o processo de ensino de programação através de jogos digitais. Os estudos selecionados apresentam oficinas de jogos e experimentos de curto e longo período entre escolas e universidades. Sete artigos apresentaram oficinas como estudo de caso, dos quais apenas um deles foi elaborado em universidade, o restante em escolas públicas (Id.01. Id.03, Id.05, Id.06, Id.13, Id.14 e Id.15). Cinco artigos foram realizados em universidades durante o período letivo dos cursos de computação (Id.02, Id.08, Id.09, Id.10 e Id.12). Três artigos ofereceram um curso fora das disciplinas para alunos da graduação em computação e da escola (Id.07, Id.11 e Id.17). Apenas dois artigos não informaram como foi realizada a avaliação do estudo de caso (Id.04 e Id.16).

QP2: Qual público alvo está sendo focado nos trabalhos de pesquisa?

Notamos que oito estudos foram realizados em escolas públicas, ou seja, acredita-se que o ensino de programação deve-se começar no ensino médio (Id.01, Id.03, Id.05. Id.11, Id.13, Id.14 e Id.15). Oito estudos foram realizados em universidades, de forma que o ensino de programação com jogos fosse uma alternativa de aprendizado para que o índice de reprovação de alunos em disciplinas de algoritmo seja menor (Id.02, Id.06, Id.07, Id.08, Id.09, Id.10, Id.12 e Id.17). Apenas dois artigos não informaram como foi realizada a avaliação do estudo de caso (Id.04 e Id.16).

QP3: Quais as ferramentas estão sendo empregadas?

Foram encontrados 11 ferramentas que auxiliam o ensino de programação: PyGame, RoboMind, Lego Mindstorms, Takkou, Scratch, Alice, iVprog, Escracho, Kodu, Game Maker e Contruct 2. Na Tabela 4 pode ser visualizar para quais estudos foram utilizadas as ferramentas.

A ferramenta PyGame é um módulo Python que fornece a API da biblioteca SDL (feita em C) e mais algumas funcionalidades para a programação gráfica, em

especial jogos. Essa ferramenta baseia-se na ideia de que as tarefas mais intensivas a nível computacional em um jogo podem ser abstraídas separadamente da lógica principal (Id.01 e Id.09).

O RoboMind é um ambiente de programação educacional simples com a sua própria linguagem de script que permite que os iniciantes aprendam as noções básicas de informática através da programação de um robô simulado (Id.02).

O Lego Mindstorms é uma linha do brinquedo LEGO voltada para a educação tecnológica. É utilizada para função lúdica e didática em instituições de ensino tecnológico abordando a teoria e a prática de conteúdos direcionados para a introdução à robótica, permitindo o desenvolvimento de projetos de pequeno e médio porte, estimulando a criatividade e a solução de problemas do cotidiano por parte dos alunos (Id.03, Id.08 e Id.14).

O Takkou é uma ferramenta que auxilia o aluno no exercício do raciocínio lógico através de um jogo suscitando maior motivação com base em aspectos pedagógicos inspirados na teoria de David Ausubel (Id.04).

O Scratch é uma linguagem gráfica de programação, inspirada no Logo, que possibilita a criação de histórias interativas, animações, simulações, jogos e músicas, e a partilha dessas criações na Web (Id.05 e Id.13).

O Alice é um ambiente de programação tridimensional que a permite a criação de animações e pequenos jogos por meio de um ambiente gráfico interativo, bastando clicar e arrastar as instruções para criar um programa (Id.06).

O iVprog é um sistema para uso no ensino-aprendizagem de tópicos relativos a algoritmos, com aplicação direta em páginas Web. Permite aos alunos construir seus algoritmos interagindo com elementos visuais (Id.07).

O Escracho é um programa do qual parte da linguagem é representada por elementos gráficos que podem ser arrastados e encaixados entre si, desde que sejam respeitadas algumas restrições sintáticas (Id.10).

O Kodu permite que as crianças criem jogos no PC e Xbox através de uma linguagem simples de programação visual (Id.11).

O Game Maker é um motor de jogo escrito em Delphi, que permite um desenvolvimento rápido e possui recursos que garantem ao jogo uma boa usabilidade, como: áudio, vídeo e acesso em rede (Id.12).

O Construct 2 é uma ferramenta que não é necessário codificar, pois tudo é feito de forma automática. O usuário apenas irá criar o enredo do jogo, o restante da aplicação é feita praticamente através do mouse, com a ação de arrastar e soltar os objetos no cenário principal (Id.16).

Tabela 4: Artigos Incluídos na RSL.

ID	Título	Ano	Ferramenta	Universidade	Fonte de Publicação
1	Atraindo Alunos do Ensino Médio para a Computação	2011	PyGame	UFPB	WIE
2	O uso do Lego Mindstorms no apoio ao Ensino de Programação de Computadores	2009	RoboMind	FURB	WIE
3	Ensino de Algoritmos a Nível Médio Utilizando Música e Robótica - Uma Abordagem Lúdica	2011	Lego Mindstorms	UFRPE	WIE
4	Takkou - Uma Ferramenta Proposta ao Ensino de Algoritmos	2011	Takkou	UFRN	WIE
5	Ensino de Ciência da Computação na Educação Básica Experiências Desafios e Possibilidades	2012	Scratch	UPE	WIE
6	Limitações da Utilização do Alice no Ensino de Programação para Alunos de Graduação	2012	Alice	PUC/PR	SBIE
7	Uma visão do cenário Nacional do Ensino de Algoritmos e Programação - uma proposta baseada no Paradigma de Programação Visual	2012	iVprog	USP	SBIE
8	A Robótica como Ferramenta de Apoio ao Ensino de Disciplinas de Programação em Cursos de Computação e Engenharia	2011	Lego Mindstorms	UFF	WIE
9	Aprendendo a Ensinar Programação Combinando Jogos e Python	2010	PyGame	UFPB	SBIE
10	Avaliação Empírica da Utilização de um Jogo para Auxiliar a Aprendizagem de Programação	2010	Escracho	UNIVALE	SBIE
11	Kodu Game Labs: Estimulando o Raciocínio Lógico através de Jogos	2012	Kodu	UFRJ	SBIE
12	ProGame: um jogo para o ensino de algoritmos e programação	2010	Game Maker	IESP/UFPB	SBIE
13	Programação no Ensino Médio: Uma Abordagem de Ensino Orientado ao Design com Scratch	2012	Scratch	UFPB	WIE
14	Proposta Metodológica para a Inserção ao Ensino de Lógica de Programação com Logo e Lego Mindstorms	2012	Lego Mindstorms	ULBRA	SBIE

15	RoboEduc: Um Software para Programação em Níveis	2010	*	UFRN	WIE
16	Um jogo para o ensino de programação em Python baseado na taxonomia de Bloom	2012	Construct 2	UFPB	WIE
17	Uma abordagem lúdica para a aprendizagem de programação de computadores	2010	*	PUC/PR	WIE

QP4: Quais são os efeitos observados nos alunos com a introdução de atividades de programação baseadas em jogos digitais?

Os resultados dos estudos de caso dos artigos foram avaliados de forma positiva para o ensino de programa com jogos digitais. Apenas um artigo relata que o uso de jogos digitais desenvolvidos com a ferramenta Alice, de acordo com os testes realizados pelos pesquisadores do artigo (Id.06), não contribui para um melhor rendimento dos alunos em disciplinas de programação. Com os resultados obtidos podemos afirmar que o ensino de programação com jogos digitais foi de aceitação geral entre os alunos de escolas públicas e de graduação.

QP5: Quais teorias pedagógicas estão sendo utilizadas?

Esta questão tem como objetivo apontar as principais teorias pedagógicas usadas para ensino de programação com jogos digitais. As teorias encontradas são apresentadas na Tabela 5.

Tabela 5: Teorias pedagógicas utilizadas.

Teoria Pedagógica	Descrição
Teoria de aprendizagem significativa	Propõe que os conhecimentos prévios dos alunos sejam valorizados, para que possam construir estruturas mentais utilizando como meio, mapas conceituais que permitem descobrir e redescobrir outros conhecimentos, caracterizando, assim, uma aprendizagem prazerosa e eficaz (Id.04).
Teoria pedagógica do construtivismo	Propõe que o aluno participe ativamente do próprio aprendizado, mediante a experimentação, a pesquisa em grupo, o estimulo a dúvida e o desenvolvimento do raciocínio, entre outros procedimentos (Id.02, Id.08, Id.11 e Id.14).
Taxonomia de Bloom	Tem por finalidade auxiliar a identificação e a declaração dos objetivos ligados ao desenvolvimento cognitivo que engloba a aquisição do conhecimento, competência e atitudes, visando facilitar o planejamento do processo de ensino-aprendizagem (Id.10, Id.12 e Id.16)

Nove artigos não apresentaram a utilização de alguma teoria pedagógica para elaborar sua metodologia de ensino de programação.

QP6: Quais são as instituições de pesquisas envolvidas na área e como elas estão sendo distribuídas no Brasil?

Como pode ser visualizada na Tabela 6 a revisão encontrou 12 instituições de ensino diferentes que desenvolveram estudos sobre o ensino de programação através de jogos. Dentre elas, podemos ressaltar a UFPB com cinco estudos desenvolvidos, sendo um em conjunto com outra instituição da Paraíba a IESP. Temos também a UFRN e a PUC/PR com dois estudos, a demais instituições tiveram apenas um artigo.

Tabela 6: Quantidade de artigos por instituição de pesquisa.

Instituição	Artigos	Instituição	Artigos
UFPB	4	UPE	1
UFRN	2	USP	1
PUC/PR	1	UFF	1
FURB	1	UFRJ	1
UFRPE	1	ULBRA	1

Pode-se notar, através do gráfico ilustrado na Figura 11 que a região Nordeste se sobressai em relação às demais, pois obtiveram maior quantidade de estudos em ensino de programação com jogos digitais, totalizando 53%.

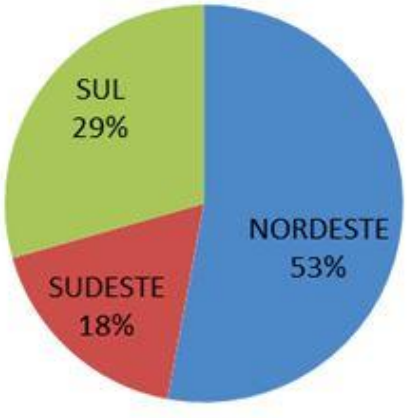

Figura 15: Porcentagem de artigos por região.

4.3 Análise de Resultados

Após a realização da RSL foram identificados alguns pontos fracos e oportunidades para melhorias futuras. Nesta seção, apresentamos estas novas perspectivas, baseadas nos resultados do mapeamento sistemático.

4.3.1 Ausência de jogos que ensinam programação

Através dos resultados obtidos pela RSL, observamos que nenhum artigo realizou o estudo de caso com jogos que ensinem programação. Todos, sem exceção, utilizaram mecanismos que ensinam programação criando um jogo ou criando um robô. O uso de um jogo que ensina programação pode ser importante, pois se torna um meio ainda mais atrativo ao aluno, mantendo assim a criança motivada e consequentemente, aprendendo mais sobre programação.

4.3.2 Temática de Robótica no contexto de jogos digitais

Pode-se analisar que grande parte dos estudos foi realizada com a utilização da robótica. O esforço para executar e gerenciar estudos com robótica às vezes faz com que não seja possível utilizar essa metodologia em escolas para o ensino médio. Notamos que os experimentos de robótica foram utilizados apenas nas universidades.

4.3.3 Impacto de ensino de programação para as escolas

A proposta de utilizar jogos digitais para o ensino de programação deve ser focada no ensino médio. Especialistas em tecnologia, educadores e engenheiros defendem a inserção do ensino da programação nas escolas como uma maneira de compreender o que está por trás de todas as tecnologias que temos acesso, além de contextualizar o aprendizado adquirido na escola. O problema é como disseminar essa

cultura no país, o que exige treinar professores e dar infraestrutura. Segundo a pesquisa da Fundação Pensamento Digital de 2010, em todo o país há 79 cursos de licenciatura em computação, que preparam educadores para lecionar sobre esse tema. Mas eles formam menos de 700 profissionais por ano. Entretanto, uma das coisas que assusta educadores e gestores de escolas quando se fala em robótica e programação são os custos para implementar essas disciplinas.

4.3.4 Instituições de pesquisa mais atuantes na área

A distribuição dos artigos mostrou que mais da metade dos artigos foram publicados de 2010 a 2012. Nos anos de 2010 e 2011 as pesquisas concentraram-se na região Nordeste, com destaque para as instituições UFPB e UFRN, neste mesmo período, um único artigo contou com a colaboração de outra instituição, foi uma pesquisa realizada entre a UFPB e a IESPB. No ano de 2012 as pesquisas concentram-se na região Sul e Sudestes. A distribuição das instituições de pesquisa mostrou que a maior parte delas está localizada nas regiões Nordeste, Sul e Sudeste do Brasil. Com destaque para a região Nordeste, correspondendo mais da metade das pesquisas nesta área, ou seja, a 53% do total de artigos.

4.3.5 Limitações da RSL

A limitação deste trabalho diz respeito aos anais do WEI de 2008 aos quais não tivemos acesso, fazendo com que esse ano não fosse incluído na revisão. Apesar de eles representarem 6\% do total de artigos acessados, isso pode significar que a revisão aqui apresentada não tenha contemplado todos os artigos importantes na área de jogos digitais para o ensino-aprendizagem de programação no contexto do WEI.

Vimos nesse capitulo os resultados de uma RSL sobre o processo de ensino-aprendizagem de programação através de jogos digitais publicados nos últimos cinco anos, em relevantes eventos e revistas na área. Notamos que os jogos digitais de ensino

de programação para universitários estão voltado para melhorar o rendimento dos alunos em disciplinas de programação e segundo os resultados da RSL, essa didática ajuda com o desempenho dos alunos.

Percebemos com a RSL a utilização frequente de algumas ferramentas de criação de jogos digitais. No próximo capitulo vamos visualizar como é o processo de criação de jogos para implementar no framework e como o framework funciona nesse processo.

5 Framework para criação de jogos voltados para o ensino de lógica de programação

Para realizar a construção de um Framework se faz necessário detalhar o processo utilizado na construção, definir conceitos e mecanismos de pesquisa e escolha de trabalhos científicos que suportam a construção desse framework.

Framework é uma abstração que une códigos comuns entre vários projetos de software provendo uma funcionalidade genérica. Um framework pode atingir uma funcionalidade específica, por configuração, durante a programação de uma aplicação. (Muller, 2008). Este trabalho propõe a integração de duas ferramentas (Blockly e Construct 2) que facilitam o aprendizado dos conceitos de programação. Com o auxilio do Blockly os estudantes desenvolvem sua habilidade de pensar logicamente para a resolução de problemas. Os jogos produzidos pelo framework possibilita o aprendizado de programação de forma inovadora, interativa e visual e já alcança um grande número de usuários.

Diante disso, o framework foi projetado para integrar essas ferramentas, de modo que as capacidades delas possam ser potencializadas pelo uso complementar de suas funcionalidades, buscando uma maior eficiência do estudante em aprender os fundamentos de programação.

5.1 Detalhamento do Framework

No framework, foram criados contêiner para apresentar os componentes de um jogo, do Blockly e o ativador de joystick do jogo, conforme ilustrado na Figura 12.

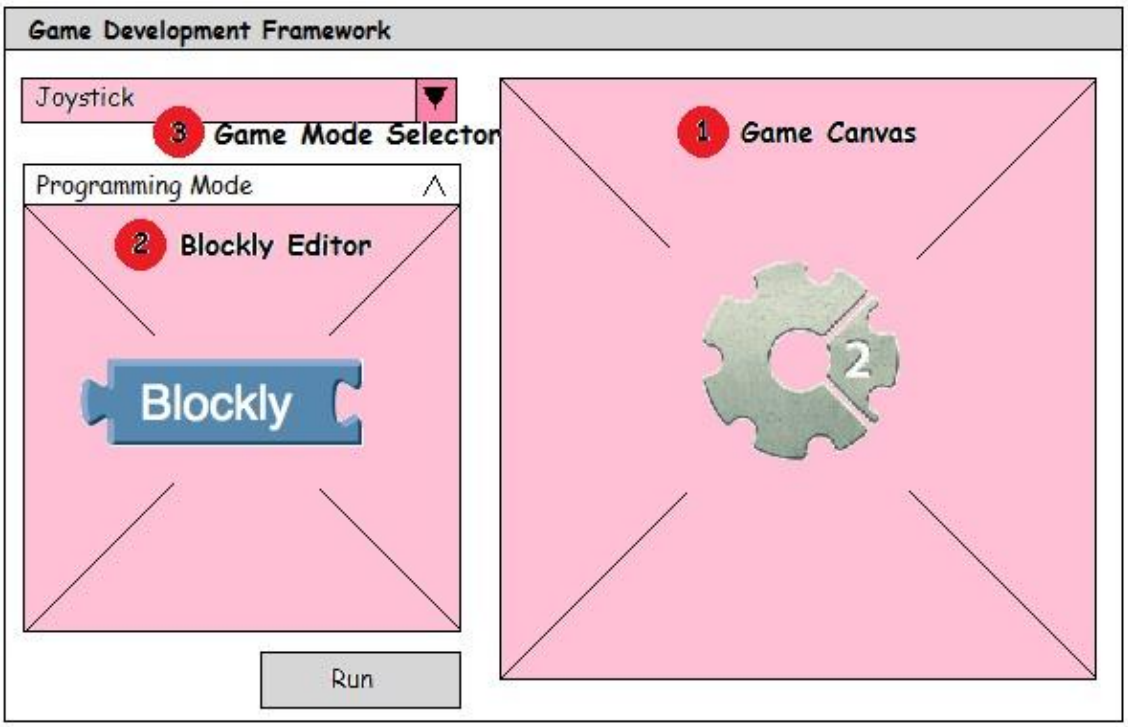

Figura 12: Game Development Framework

No contêiner 1, Game Canvas, um jogo em HTML5 é dinamicamente carregado no espaço delimitado por ele. Da mesma forma ocorre com o Blockly, que é carregado no contêiner 2 – Blockly Editor. Já o contêiner 3 tem como objetivo permitir selecionar o modo de jogo (joystick ou blockly). Isto porque opcionalmente os jogos podem ser inicialmente jogados de forma convencional, através de teclas do teclado, mouse, etc., para que o jogador explore e entenda mais rapidamente o cenário do jogo, criando então uma estratégia que possa ser programada no Blockly e executada através do botão Run.

A Figura 13 apresenta a estrutura de diretórios do framework.

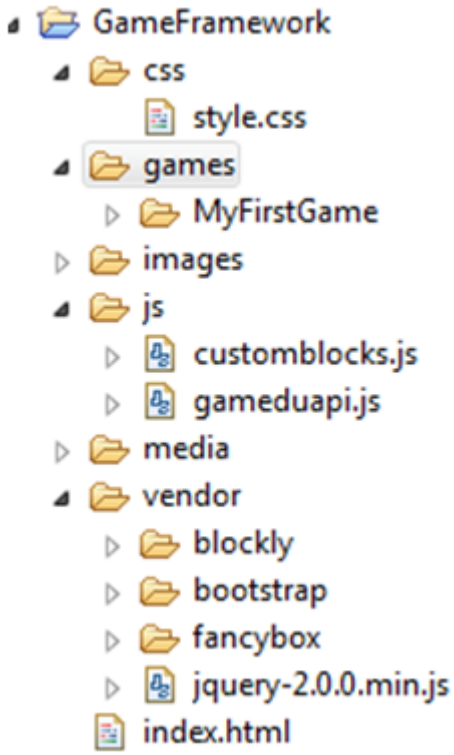

Figura 13: Diretórios do Framework

Compondo essa estrutura estão os seguintes diretórios e arquivos chaves:

- css: Diretório responsável por armazenar as folhas de estilo padrões aplicadas quando executado um jogo.
- games: Diretório responsável por armazenar o(s) código(s) gerado(s) pelo Construct 2 na exportação de um jogo para HTML. É recomendado que para cada jogo seja criado diretório com seu nome, e nele depositado o(s) código(s) criado(s) pelo Construct 2.
- images: Neste diretório são armazenados os artefatos gráficos do(s) jogo(s) HTML exportado pelo Construct 2.
- js: Diretório responsável por armazenar os códigos cruciais para a integração do Blockly com os jogos HTML exportados pelo Construct 2.
- js/gameduapi.js: Neste arquivo é construída a lógica necessária para integração do Blockly com um jogo HTML exportado pelo Construct 2.
- js/customblocks.js: Neste arquivo são definidos novos blocos a ser adicionados ao editor Blockly, que, por padrão, já possui uma coleção de blocos predefinidos.

- media: Diretório onde são armazenados os artefatos de mídia padrões utilizados pelo editor Blockly.
- vendor: Diretório onde são armazenadas os arquivos padrões das bibliotecas e tecnologias utilizadas pelo framework (Blockly, Bootstrap, Fancybox e jQuery).
- index.html: Neste arquivo são carregados os artefatos apresentados durante execução do(s) jogo(s).

5.1.1 Estrutura do Game

O jogo a ser carregado no contêiner 1 (Game Canvas) deve seguir certas regras para que sua integração ocorra com o Editor Blockly carregado no contêiner 2, conforme mostrado na Figura 14.

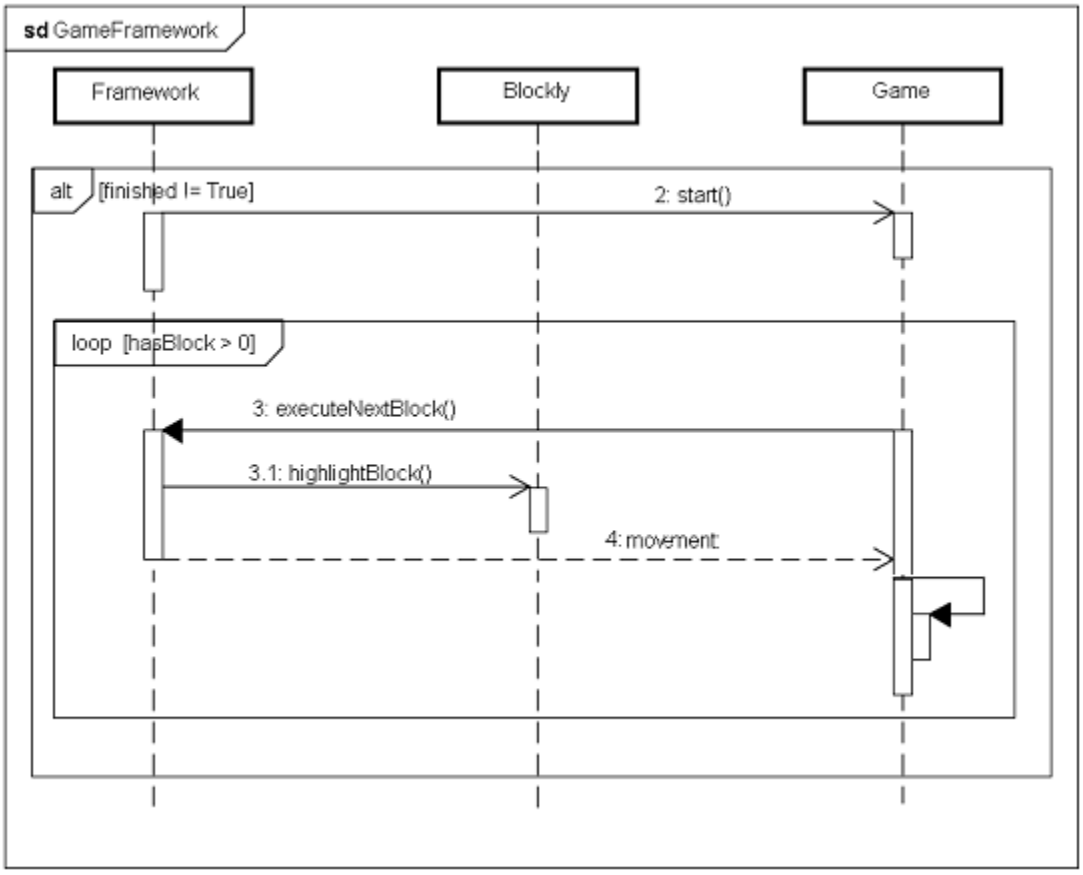

Figura 14: Diagrama de Sequência

Cada bloco utilizado no Blockly representa um movimento ou conjunto de movimentos a serem executados em um jogo. Ao chamar a execução do Blockly através do método execute(), Figura 19, ele converte cada bloco presente no seu workspace em

linguagem de programação javascript pelo método workspaceToCode() e cada movimento é armazenado em uma fila de execução do framework. Criada a fila, uma rotina de assinatura start() deve ser implementada no jogo, esse método é chamado automaticamente, após o execução do método execute() (Figura 15 e Figura 16).

```
execute: function(){
    Blockly.mainWorkspace.traceOn(true);
    var code = Blockly.Generator.workspaceToCode('JavaScript');
    try {
        //Executa Comandos do Blockly, colocando-os em path
        eval(code);
    } catch (e) {
        if (e !== null) {
            alert(e);
        }
    }
},
```

Figura 15: Método execute

```
Gamedu.prototype = {
    game : new Game(),
    blockly: new GameBlockly(),
    startGame : function() {
        return this.game;
    },
    getHasBlocks : function() {
        return gamedugame.blockly.hasBlock;
    },
    getExecuteNextBlock: function() {
        return this.blockly.executeNextBlock();
    },
    setGameStats : function(params) {
        try {
            this.game.gameStats.victory = params[0];
            this.game.gameStats.tries = params[1];
            this.game.gameStats.time = params[2];
            console.debug(this.game);
        } catch(e) {
            console.debug("falhou");
        }
        return this.game;
    }
};
```

Figura 16: método start

Após o "start()", o jogo começa a buscar na fila um movimento a ser executado em seu cenário, através do método "executeNextBlock()" do framework. Chamado esse método, ele retorna ao jogo o movimento que ele deve cumprir e depois chama o método "highlightBlock()" do Blockly para que o bloco do respectivo

movimento seja posto em ênfase no workspace. Após essa execução, e havendo ainda comandos na fila, o jogo continua realizando chamadas ao método "executeNextBlock()", conforme ilustrado na Figura 17.

```
executeNextBlock: function(){
    var tuple = this.path.shift();
    console.log(this.hasBlock);
    console.log(tuple);
    try{
        Blockly.mainWorkspace.highlightBlock(tuple[1]);
        this.hasBlock -= 1;
        return [tuple[0],tuple[2]];
    }catch (e) {
        return "";
    }
},
```

Figura 17: método executeNextBlockly

5.1.2 Usando e estendendo o Blockly

Assim como o jogo, algumas especificações devem ser cumpridas pelo Blockly para sua integração.

Assumindo que não existem blocos que executem os movimentos do jogo a ser programado, para todo movimento do jogo, deve ser inserido um bloco que o adicione na fila de execução.

A definição de novos blocos deve ser realizada no arquivo "js/customblocks.js". Os blocos definidos nesse arquivo são adicionados automaticamente ao Blockly.

A Figura 18 mostra um trecho de código em que um novo bloco ("shootMovement") é criado no Blockly. Esse bloco quando convertido em linguagem de programação (Javascript), executa o método "addToPath()" passando, como parâmetro, o movimento de jogo a ser adicionado a fila de execução.

```
//Block to add shoot movement in the running path
Blockly.Language.shootMovement = {
    init : function() {
        //Block color
        this.setColour(290);
        //Block label
        this.appendDummyInput().appendTitle(Blockly.MSG_SHOOT);
        //Block accept previous statement
        this.setPreviousStatement(true);
        //Block accept next statement
        this.setNextStatement(true);
        //Block shows everything in line
        this.setInputsInline(true);
        //Block tip
        this.setTooltip('Shoot');
    }
};

//
Blockly.JavaScript.shoot = function() {

    return 'addToPath(shoot)';
};


function addToPath(movement){
    framework.path.push(movement);
    hasblocks += 1;
}
```

Figura 18: Código criado para adicionar blocos de movimentos do jogo no Blockly

5.2 Desenvolvimento de jogos usando o framework

Para exemplificar o uso do framework segue desenvolvido um jogo exemplo que através do framework se integra com o Blockly.

Para exemplificar o uso do framework apresentado, foi desenvolvido um jogo de ensino de lógica de programação em cenários inspirados em situações do mundo real. Nesse jogo, o jogador é desafiado a transportar um container de um navio para um caminhão.

O desafio para o jogador, neste exemplo, será o de programar o comportamento dos objetos passiveis de programação (gancho e caminhão) de modo que siga os seguintes passos: mover o gancho para o lado direito e depois para baixo até alcançar o contêiner. Alcançando o contêiner o gancho o segura e então o jogador já

pode mover o caminhão para a direita até o lugar adequado. Dessa forma o gancho poderá soltar o contêiner em cima do caminhão. O jogo finaliza com a partida do caminhão para o lado esquerdo ate o mesmo sair do cenário.

A Figura 19 apresenta uma visão geral desse cenário do jogo sendo solucionado.

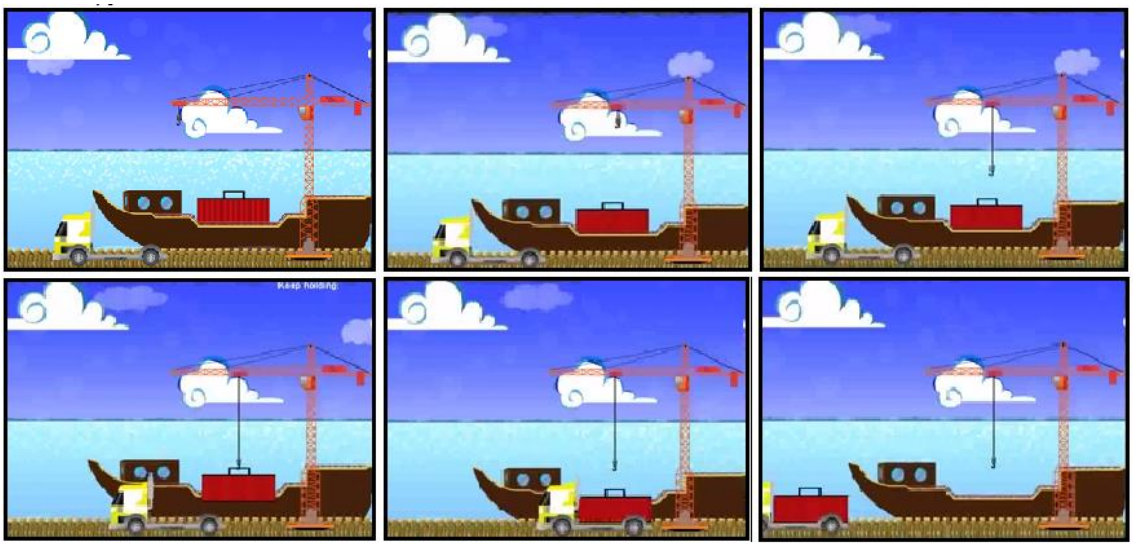

Figura 19: Desafio de transportar contêiner do navio para o caminhão

Seguindo as especificações do framework foi criado o método "start()", conforme ilustra a Figura 20.

Na implementação desse método é verificado a quantidade de blocos da fila de execução. Essa quantidade é armazenada na variável global "qtdBlocks" do jogo. Uma vez que a quantidade de blocos seja maior que zero e o jogo não esteja executando um movimento, é feita uma chamada ao método "executeNextBlock()" do framework. Quando executado, esse método retorna uma String com o nome do próximo movimento a ser realizado. Conhecido o movimento, é feita uma chamada a sua função.

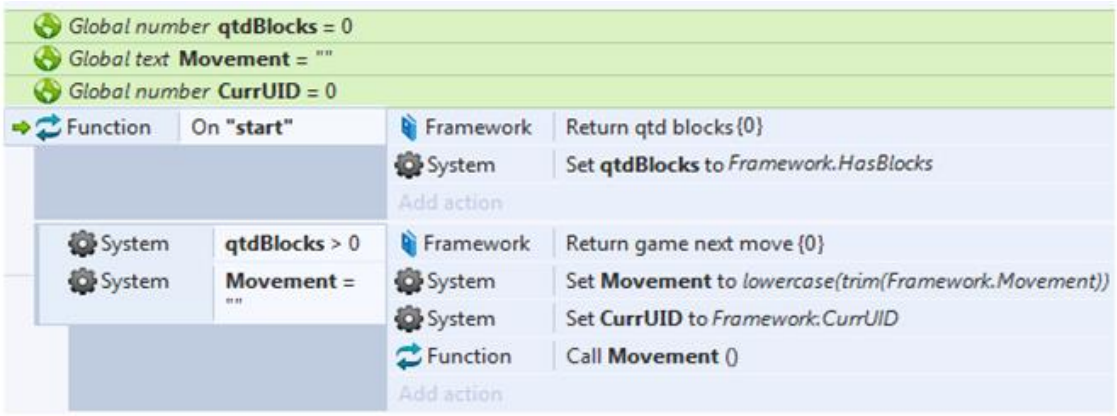

Figura 20: Método start() implementado no Construct 2

Para todo movimento passível de integração com o Blockly, deve ser criada uma função, no Construct 2, de mesmo nome que realize a ação ou ações desejadas. A Figura 21 exibe um exemplo de movimento e sua função implementadora.

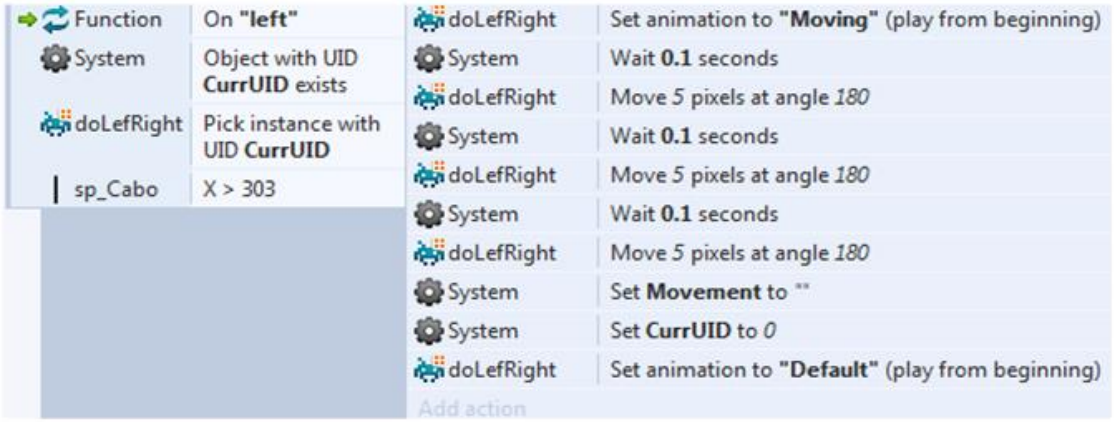

Figura 21: Exemplo de função implementadora de movimento

Nesse exemplo, quando o jogo percebe que o próximo movimento a ser executado é o "left", é feita uma chamada a função de nome igual ao movimento requerido. Assim deverá ser feito para todos os movimentos do jogo.

Finalizado a criação do jogo, ele deve ser exportado para HTML. O Construct 2 se encarrega de gerar os artefatos de código e mídia necessários para execução do jogo.

Para concluir a integração, se faz necessária a criação dos blocos do Blockly que adicionam os movimentos do jogo na fila de execução do framework. A Figura 22 ilustra um exemplo de bloco adicionando a fila de execução do framework o movimento “left” criado para este jogo.

```
Blockly.Language.leftMovement = {
    // Block to add left movement to the running path
    init : function() {
        this.setColour(290);
        this.appendDummyInput()
            .appendTitle(Blockly.MSG_LEFT);
        this.setPreviousStatement(true);
        this.setNextStatement(true);
        this.setInputsInline(true);
        this.setTooltip('This will move an object 10px to the left');
    }
};

Blockly.JavaScript.leftMovement = function() {
    return 'addToPath (\''+ 'left'+'\');';
};
```

Figura 22: Criação de bloco que adicionada a fila de execução do framework o movimento left

Esse código foi escrito no arquivo js/customblocks.js. A Figura 23 ilustra uma visão do Blockly e blocos utilizados por este protótipo.

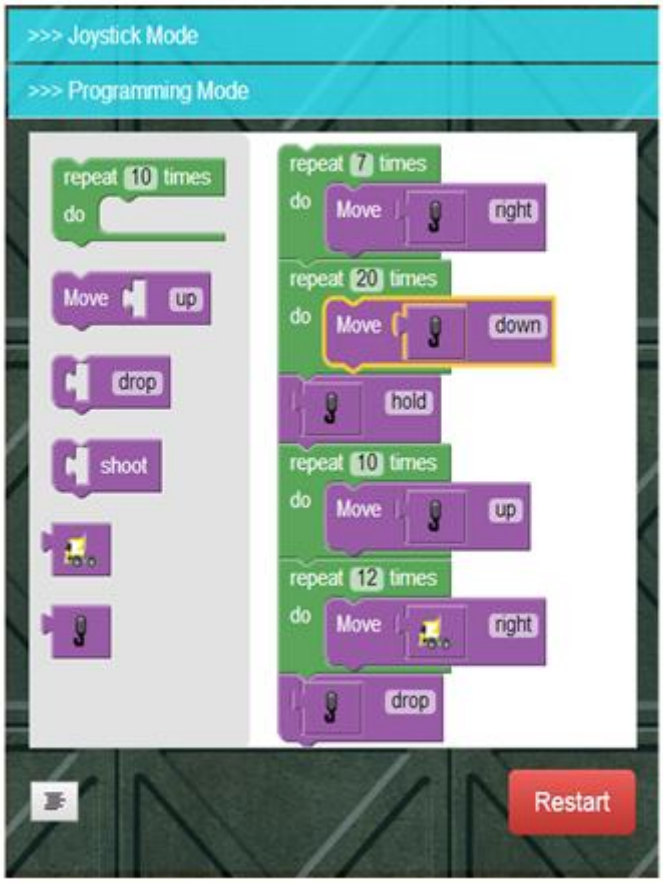

Figura 23: Exemplo de jogo desenvolvido

Na Figura 23 os blocos são utilizados de forma a representar os movimentos e sequência de execução necessária para alcançar o objetivo do jogo.

A Figura 24 apresenta uma visão do jogo em sua versão final, já integrado ao Blockly através do framework.

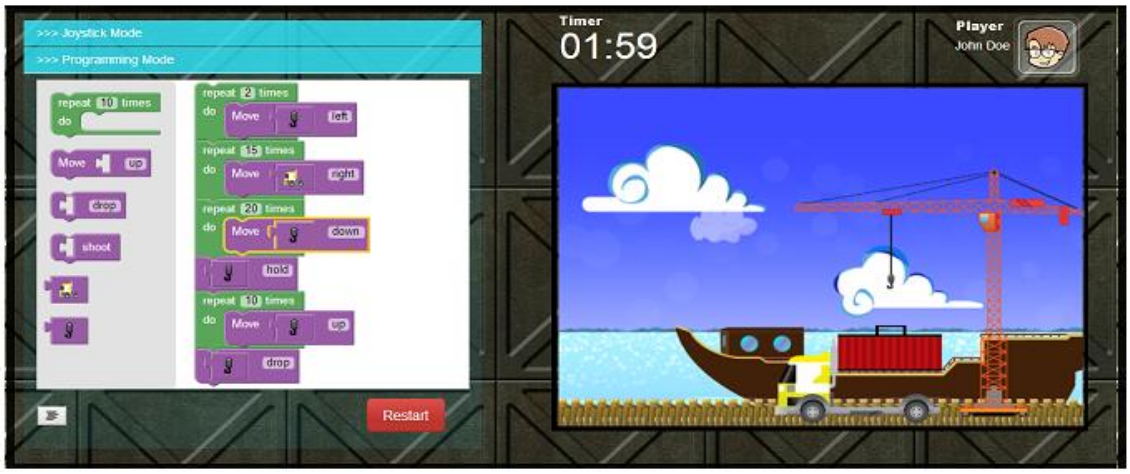

Figura 24: Exemplo de jogo desenvolvido integrado ao Blockly através do framework

5.3 Arquitetura do Jogo

Para começar a programar se faz necessário pensar e planejar como tudo vai funcionar, entender a arquitetura, os componentes, e como tudo se encaixa juntos. Essa sessão explora os principais componentes de código utilizado no framework e como eles se integram.

Todos os subsistemas do jogo foram classificados em uma das três camadas principais: camada de aplicação, camada lógica do jogo e camada de visão do jogo (Figura 25).

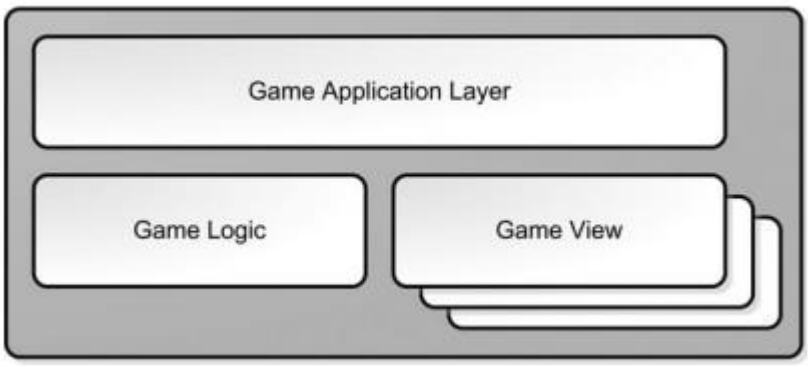

Figura 25: Arquitetura do framework

Fonte: McShaffry e Graham, 2013

A camada de aplicação lida com as chamadas diretas do construct 2. A camada lógica apresenta o estado do jogo e como o ciclo de vida muda ao longo do tempo. Já na camada de visão encontra-se o estado do jogo como, por exemplo, gráficos e sons. A vantagem de separar um jogo em três sistemas independentes é que você terá bastante flexibilidade em possíveis alterações.

A camada de aplicação preocupa-se com a execução do jogo. A camada lógica é o jogo, completamente separado da maquina ou como ele é apresentado ao jogador. Em um mundo perfeito, seria possível simplesmente recompilar todo o código fonte relacionado a camada lógica do jogo, e o mesmo seria executado em qualquer plataforma ou sistema operacional. Nesta área, encontram-se todos os subsistemas que vão administrar o estado do jogo, comunicando as mudanças de estado para outros sistemas e aceitando comandos de entrada a partir de outros sistemas. Ainda na camada lógica, encontram-se sistemas que impõem regras do universo do jogo, como por exemplo, o sistema físico que informa como os objetos do jogo se movem e interage.

O terceiro e ultimo componente da arquitetura é a camada de visão. Este sistema é responsável por apresentar o estado do jogo e traduzir as entradas em comandos do jogo, que serão, posteriormente, enviadas para a camada lógica. Ainda nesta camada, podemos ter várias implementações diferentes, dessa forma é possível ter vários pontos de vista ligados ao jogo e que o computador pode controlar.

5.3.1 Aplicando a arquitetura em um jogo

Para entender como funciona essa arquitetura planejada para o framework, vamos projetar um jogo chamado “Trilha” usando as camadas de aplicação, lógica e visões. Serão projetados dois pontos de vista: o modo joystick, que irá interagir com o jogo a partir de comandos de entrada do teclado; e o modo blockly, que irá interagir com o jogo a partir dos blocos de comandos do ambiente de programação blockly.

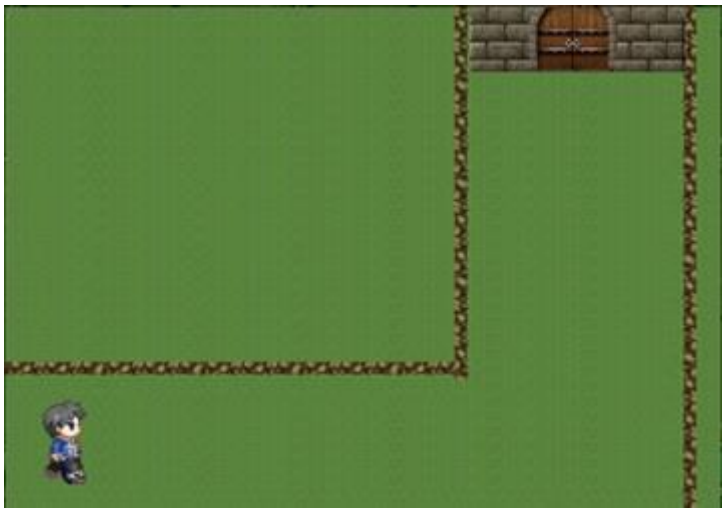

Figura 26: Cenário do jogo Trilha

A lógica do jogo determina que o personagem percorra a trilha até alcançar a porta. Na Figura 26 podemos ver o cenário construído através do construct 2.

Na camada lógica do jogo Trilha terá dados que descrevem o personagem e todas as suas propriedades. O personagem tem várias direções ao longo de toda a rota. Como entrada, a camada lógica do jogo se preocupa apenas com a direção do personagem. Como saída lógica temos as mudanças de estado e eventos do personagem, com cada posição do personagem.

Na Figura 27, observa-se a camada lógica do jogo. Onde foram criadas funções para cada direção do personagem: “moveUp”, “moveLeft”, “moveRight”, “moveDown”. A função “restart” tem como papel colocar o jogo para a posição inicial definida. Como colisão, a lógica estabelecida foi finalizar o jogo quando o personagem alcançar o objeto porta ou muro. Essa colisão irá fazer uma chamada ao sistema para a tela de finalização do jogo, que se encontra na camada de aplicação, modificando a variável won com valor 1 se alcançar a porta, e valor 2 se alcançar o muro.

Figura 27: Camada Lógica

O estado do controlador é interpretado pela camada de visão do jogo e convertidos em comandos, os quais são enviados para a camada lógica do jogo através de um evento, como visto na Figura 28, explicada a seguir.

Figura 28: Camada de visão

Na visão do Keyboard, os comandos gerados através do teclado irão ser passados como evento para a camada lógica, o qual irá fazer o direcionamento indicado do personagem. Por exemplo, o comando “Right arrow” da camada de visão do Keyboard, irá fazer uma chamada a função “moveRight” que na camada lógica irá direcionar o personagem para a direita.

A visão do Blockly é um pouco diferente, pois ele vai receber os comandos gerados pelo ambiente de programação visual Blockly e ser interpretados pelo plugin criado chamado Gamedu. Quando o jogador informar seus blocos de comandos que serão utilizados na execução do jogo, esses blocos serão colocados numa pilha de execução pelo plugin. O plugin Gamedu vai retornar ao sistema a quantidade de blocos existentes na pilha de execução, “return qtd blocksstat {0}”. O sistema vai reservar essa quantidade em uma variável local chamada “QtdeB”, que foi inicializada com o valor zero, “set QtdeB to Game.HasBlocks”.

Enquanto a variavel “QtdeB” for maior que zero, o plugin vai estar retornando o próximo movimento que tem que ser executado, “return game next move form blockly{0}”. A partir disso o sistema vai reservar o próximo movimento em uma variável local chamada “Movent”, que foi inicializada como nula. Enquanto essa variável for nula, o sistema vai fazer a chamada “set Movent to gamedu.movement” que vai realizar um pop na pilha, ou seja, o ultimo elemento da pilha será retornado e executado pelo construct. Por fim, a variável “movente” volta a ser nula.

Para finalizar a arquitetura do o jogo Trilha, na camada de aplicação podem ser encontrados os comandos de inicialização e finalização do jogo. Junto com a inicialização o sistema define as variáveis globais won e timer para zero. A variável timer controla o tempo de duração do jogo. A partir do primeiro comando do jogo a variável vai adicionando 1 ao seu valor. Já a variável won controla a finalização do jogo. Na camada lógica foi visto que a partir da colisão esta variável receberia o valor 1 ou 2, portanto, na camada de aplicação quando won for maior que zero, o sistema irá deixar visível a layer de finalização do jogo. Com a layer visível o sistema verifica o valor da variável, se for 1 aparecerá a mensagem de vencedor, caso contrario aparecerá a mensagem de perdedor. A camada de aplicação pode ser vista na Figura 29.

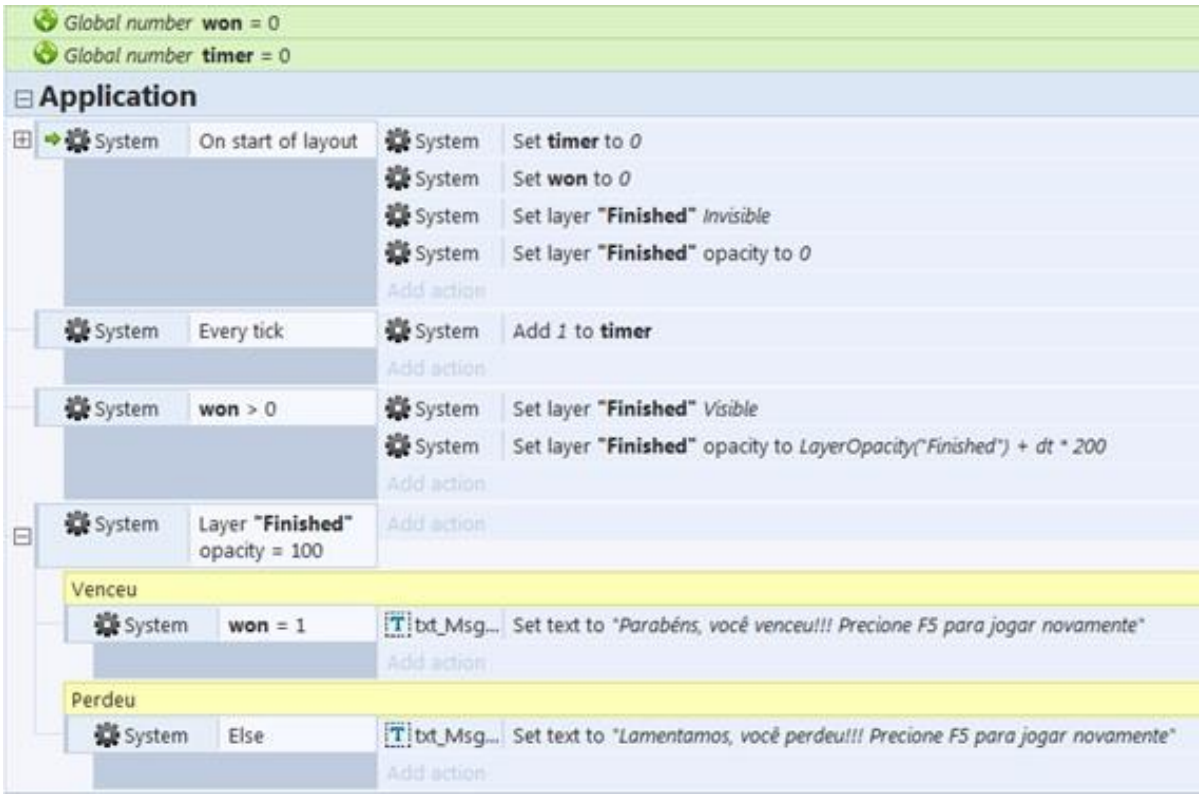

Figura 29: Camada de Aplicação

Finalizada a criação do jogo, ele deve ser exportado para HTML5. O Construct 2 se encarrega de gerar os artefatos de código e mídia necessários para execução do jogo. A Figura 30 apresenta uma visão do jogo em sua versão final, já integrado ao Blockly através do framework.

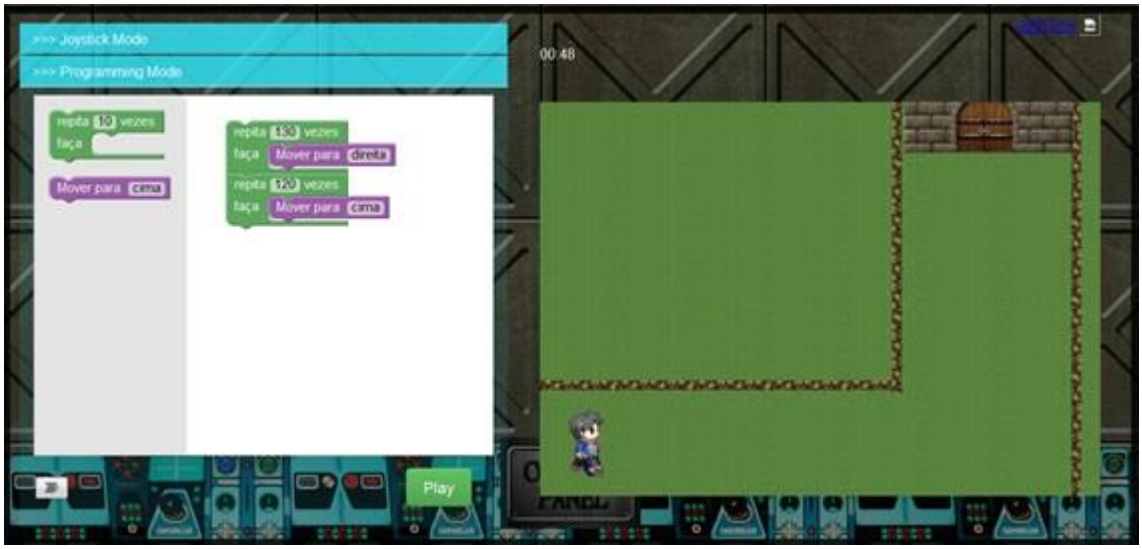

Figura 30: Jogo Trilha

6 Avaliação do framework

Este capítulo apresenta um estudo sobre a viabilidade do framework desenvolvido, visando mostrar sua integração com o Blockly, de modo a criar jogos atrativos para o público alvo e pedagogicamente válidos. O estudo consiste no desenvolvimento de 3 minijogos fazendo uso do framework e de sua avaliação por professores de cursos de computação. Dado que o desenvolvimento dos jogos com a sua integração com o blocky devem ser realizadas por pessoas com conhecimento em computação.

6.1 Definição de objetivos, questões e métricas

Para planejar o estudo, utilizamos a metodologia GQM que é uma abordagem de cima para baixa (top-down) para estabelecer um sistema de medição direcionado a metas/objetivos. A partir desses objetivos, são levantadas questões que abordem os objetivos e métricas que proporcionem respostas à essas questões. GQM define um modelo de medição em três níveis, como ilustrado na Figura 31 (Silva et al., 2014).

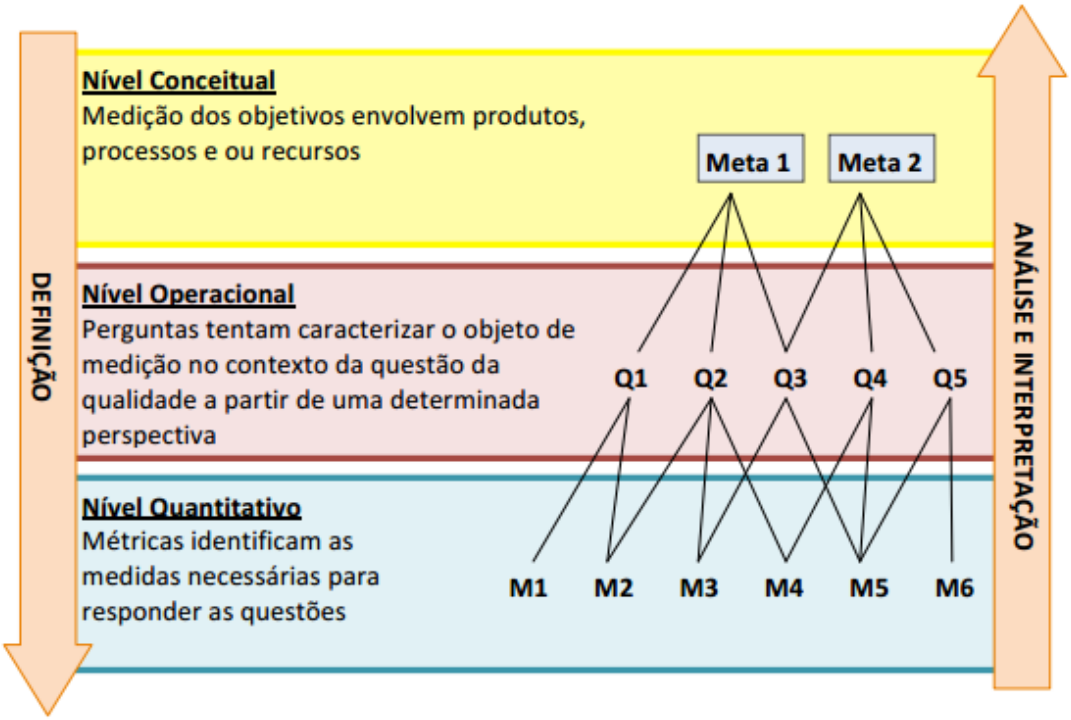

Figura 31: Metodologia GQM

Neste estudo contamos com o seguinte planejamento GQM:

- Metas (Goals)
  - ✓ M1: Avaliar se na opinião dos professores de computação o framework proposto tem potencial para trabalhar lógica de programação em crianças e adolescentes.
  - ✓ M2: Avaliar se, na opinião dos professores de computação, os jogos criados a partir do framework proposto mostram-se apropriados para uso com crianças e adolescentes.
- Questões (Questions)
  - ✓ Q1. Os jogos desenvolvidos com o framework proposto possuem potencial para desenvolver o raciocínio lógico da criança?
  - ✓ Q2. Os jogos desenvolvidos com o framework proposto estão atrativos e apropriados para uso no ensino a crianças, jovens e adolescentes?
- Indicadores e Métricas (Metrics)
  - ✓ I1: Quantidade de professores que concordam ou discordam, e em qual grau de concordância.

6.2 Participantes

Para analisar sobre a qualidade final dos jogos que foram desenvolvidos pelo framework, convidamos dez professores da área de computação, distribuídos pelos estados do Rio Grande do Norte e da Paraíba, para avaliá-los. A seleção dos participantes ocorreu baseado em dois fatores: acessibilidade e disponibilidade dos mesmos participarem do estudo.

O formulário foi encaminhado para quinze professores que lecionam alguma disciplina referente à programação. Obtivemos respostas de dez professores, o perfil dos participantes pode ser visto na Tabela 7.

Tabela 7: Perfil dos participantes.

INSTITUIÇÃO ONDE LECIONA	LECIONA DISCIPLINA DE PROGRAMAÇÃO	SEXO
Faculdade Mauricio de Nassau – UniNassau	Sim	Masculino
IFRN	Sim	Feminino

IFRN	Sim	Masculino
IFRN	Sim	Masculino
IFRN	Sim	Masculino
UFRN	Sim	Masculino
UFRN	Sim	Masculino
UFCG	Sim	Masculino
Faculdade Integrada de Patos - FIP	Sim	Masculino
FACEX	Sim	Masculino

Percebemos que temos nove professores e apenas uma professora envolvida na avaliação do framework. Notamos que três professores são de Universidade particulares e sete são de Universidades Federais (Figura 36).

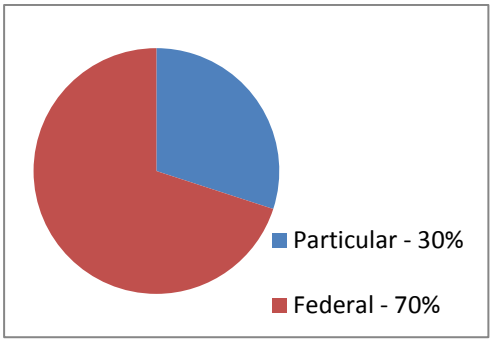

Figura 32: Percentual do tipo de Instituição

6.3 Material utilizado no estudo: jogos desenvolvidos com o framework

Foram desenvolvidos mais dois jogos além do jogo demonstrado no capitulo 5 deste documento. Essas três jogos tem como foco trabalhar com o raciocino lógico de programação baseados em cenários de engenharia.

Uma ilustração do funcionamento de cada jogo pode ser visto nas Figuras 33, 34 e 35.

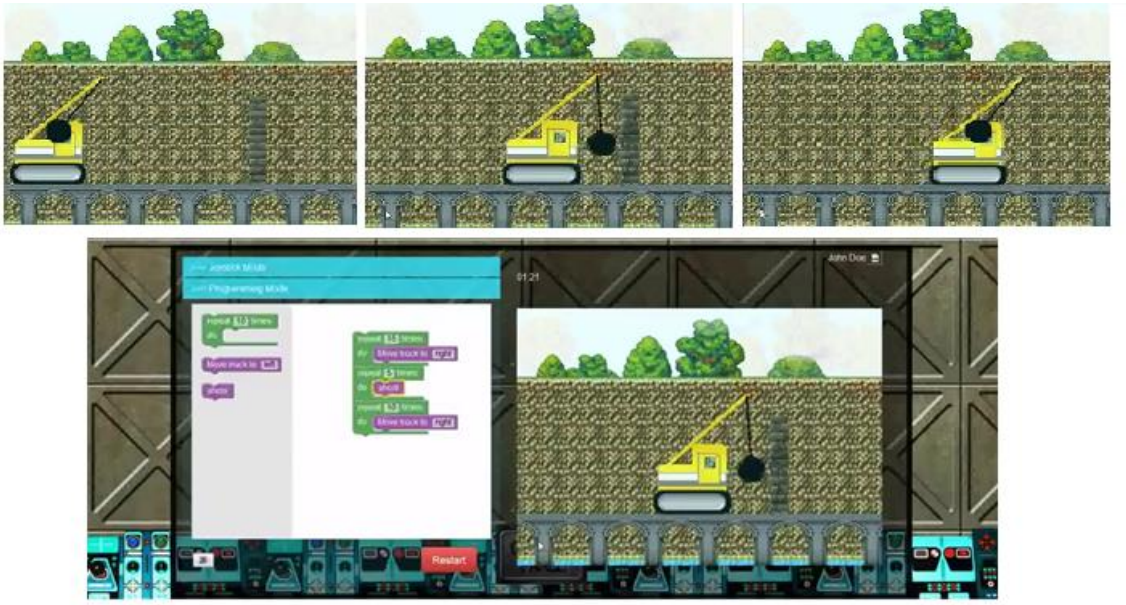

Figura 33: Jogo da bola demolidora

Figura 34: Jogo do guindaste.

Figura 35: Jogo da empilhadeira.

Essas três jogos foram agrupados e avaliados pelos professores participantes. Na Figura 36, podemos ver como os jogos foram apresentados juntamente com seus objetivos, em uma única página HTML.

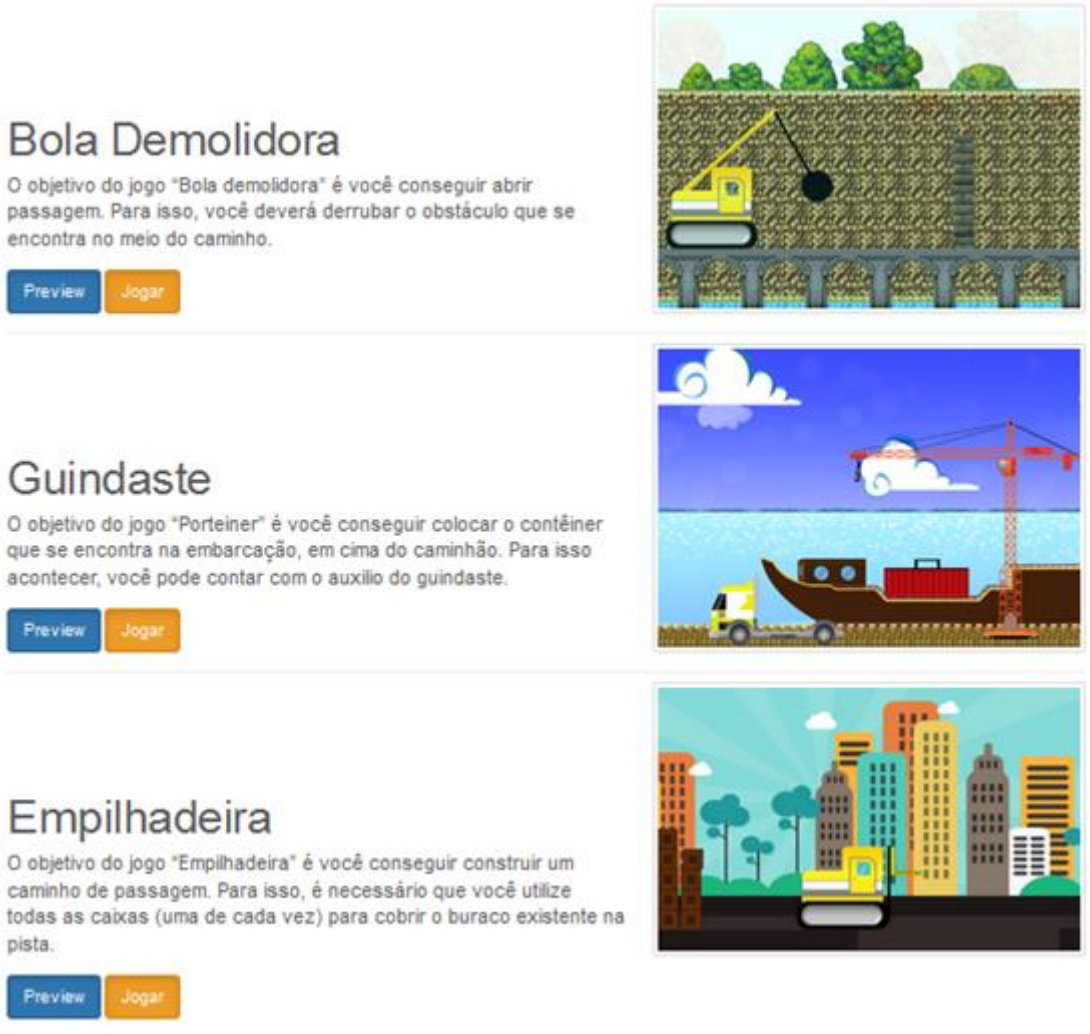

Figura 36: Jogos utilizados no estudo apresentados em conjunto.

O código fonte do framework desenvolvido, contendo os três jogos que foram utilizados nesse estudo de caso, pode ser analisado pelo link http://framework-tainamedeiros.googlecode.com/svn/trunk/.

6.4 Análise dos Resultados

Foi elaborado um conjunto de vídeos que mostravam a execução de cada jogo e um questionário que envolvesse tanto questões de usabilidade como de requisitos pedagógicos. O questionário é geralmente utilizado para encontrar dados de natureza quantitativa. No entanto, também é utilizado em investigação qualitativa, de forma a poder comparar os dados obtidos.

O inquérito por questionário é uma técnica viável de recolha de dados implicando, no entanto, alguns pressupostos considerados básicos, relativamente aos indivíduos questionados (Afonso, 2005).

As perguntas do questionário possuem uma apresentação de forma clara, simples e objetivas e as instruções de preenchimentos são precisas, claras e curtas. O questionário conta como possíveis repostas: concordo (C), concordo parcialmente (CP), discordo parcialmente (DP), discordo (D) e não tenho opinião formada (N).

Os resultados podem ser vistos na tabela 7.

Tabela 7. Avaliação dos professores

AFIRMAÇÕES	C	CP	DP	D	N
Os jogos aplicados ajudam a desenvolver o raciocinio lógico da criança.	5	5	-	-	-
Os jogos possuem potencial para ensinar lógica de programação.	6	4	-	-	-
Os jogos são atrativos, envolvendo e motivando a criança ao desafio de concluir o jogo.	6	4	-	-	-
O nível de concentração exigido está de acordo com o público alvo do jogo. (Fundamental 1)	2	5	2	-	1
O nível de concentração exigido está de acordo com o público alvo do jogo. (Fundamental 2)	4	5	-	-	1
O nível de concentração exigido está de acordo com o público alvo do jogo. (ensino médio)	7	2	-	-	1
O nível de concentração exigido está de acordo com o público alvo do jogo. (ensino superior)	5	2	1	1	1
O tempo de cada jogo é adequado. Não é facil para se tornar chato e também não é dificil para se tornar desestimulador.	5	4	-	-	1
Os jogos podem estimular a vontade da criança jogar novamente.	7	3	-	-	-

Além disso, foi deixado um espaço para possiveis comentários ou sugestões sobre os jogos. Podemos destacar o comentário de um dos entrevistados: “Acredito que os jogos aplicados em conjunto com o ensino básico da lógica de programação podem ajudar e muito a desenvolver o conhecimento da lógica de programação, em qualquer nível de ensino. Acredito também que o diferencial pode ser a complexidade do cenário do jogo aplicado ao nível de ensino (aumentando a complexidade em função do nível)”.

Pode-se verificar que os jogos desenvolvidos pelo framework obteve uma boa avaliação em todos os itens. Apenas dois professores discordaram parcialmente ou

totalmente sobre a aplicação desse tipo de jogo no ensino fundamental 1 ou ensino superior, mas quase todos eles (9 de 10) concordaram que os jogos são interessantes para o ensino de lógica de programação no ensino médio.

7 Trabalhos Relacionados

Primeiramente, temos como trabalhos relacionados algumas ferramentas que foram descritas no capítulo três deste trabalho, como por exemplo, o Scratch e ALICE.

Vimos que Scratch é um aplicativo que possibilita a criação de histórias, animações, jogos e outras produções, bem como o compartilhamento das criações na web. Tudo pode ser feito a partir de comandos que devem ser agrupados de modo lógico. A programação é efetuada através da criação de sequencias de comandos simples, que correspondem a blocos de várias categorias, encaixados e encadeados de forma a produzirem as ações desejadas (Scratch, 2013).

E que o Alice é um ambiente de programação 3D que facilita a criação de animações, jogos interativos, e vídeos. Ele é um software educacional livre que ensina conceitos de programação orientada a objetos em um contexto de criar filmes e vídeo games simples. Em Alice, os objetos 3D (pessoas, animais, veículos) são inseridos no mundo virtual, e é possível criar um programa para animá-los (Alice, 2013).

Porém, essas abordagens incentivam o estudo de programação enquanto o aluno está no desenvolvimento do jogo, ou seja, não necessariamente o jogo possui desafios de lógica de programação durante sua execução. Na verdade, o suporte desses ambientes para construção de jogos com tal objetivo é bastante limitada.

Da forma com que essas ferramentas tratam o ensino de programação, o aluno precisará de um certo nível de conhecimento sobre lógica de programação para começar a fazer os jogos. Isso pode ser comprovado a partir da RSL discutida no capitulo quatro, onde foi observado que nenhum jogo apresentava desafios que ensinem programação. Apenas utilizavam algum mecanismo que ensinam programação ou criando-se jogos ou programando-se robôs.

Outro ponto de discussão como trabalhos relacionados são os jogos descritos no capitulo dois deste trabalho. Vimos que o Light-bot exigia um pequeno conhecimento prévio sobre o que é procedimentos e funções. Já o Code Hunt necessitava que o aluno já soubesse trabalhar com as linguagens Java e C#. No FunSat o aluno precisava entender sobre circuitos e o kLogo-turtle não possui um cenário atrativo para o aluno se motivar.

Por utilizarmos o Blockly no framework, podemos dizer que os jogos gerados manualmente com o Blockly são os que mais se aproximam dos jogos gerados a partir do framework. De fato, o framework tem como diferencial facilitar o desenvolvimento de tais jogos. Com o framework, o jogo pode ser desenvolvido no modo joystick (modo tradicional de jogo) e depois incorporado ao Blockly. De fato, além disso, a arquitetura do framework permite que futuramente o editor de programação (Blockly) seja trocado por outro, seja de programação visual ou por linguagem de programação textual.

8 Conclusões e Trabalhos Futuros

Segundo Larry Corey (code.org), Presidente e Diretor do Centro de pesquisa de câncer Fred Hutchinson, “O conhecimento de programação de computadores é tão importante quanto o conhecimento de anatomia quando se trata de pesquisas médicas ou cuidados clínicos”. De fato, várias iniciativas nacionais e internacionais estão promovendo o ensino de programação ainda no nível básico (final do fundamental e médio completo).

Neste contexto, este trabalho apresenta um framework para desenvolvimento de jogos digitais voltados para o ensino de lógica de programação. Os jogos produzidos precisam ser em HTML5 para sua integração com o Blockly Editor, podendo assim serem rodados na Web ou em plataformas móveis.

A infraestrutura apresentada neste trabalho é um recurso indicado para se desenvolver jogos que ensinam lógica de programação para alunos do ensino médio. O framework facilita o desenvolvimento dos jogos voltados para o ensino de lógica de programação, que podem oferecer diferentes possibilidades de desafios, sendo possível montar os cenários e componentes dos desafios de forma a serem integrados com o editor visual de programação. Dessa forma, o framework permite que os desenvolvedores foquem na construção de jogos atrativos, deixando o trabalho de integração praticamente a cargo do framework.

Através do estudo de caso, realizado por professores da área de computação, foi possível mostrar a viabilidade do framework, bem como fazer uma análise sobre a qualidade final dos jogos produzidos através do framework (integração com o Blockly). O resultado foi considerado promissor pelos professores, bem como a aplicabilidade dos jogos produzidos com esse framework, principalmente no ensino médio. Porém, como trabalho futuro será realizado um estudo de caso avaliando os reais benefícios de se desenvolver jogos utilizando o framework desenvolvido, como também, um estudo de caso realizado com crianças para avaliar a usabilidade dos jogos de lógica de programação que podem ser desenvolvidos pelo framework.

Como trabalho futuro, pretende-se desenvolver jogos com uma maior estrutura e qualidade, onde serão inseridos pequenos desafios durante sua execução, que será feito

de acordo com o framework proposto. Dessa forma o jogo será mais atrativo e a criança ainda poderá aprender lógica de programação com os desafios do framework.

Além disso, espera-se oferecer a utilização desses jogos por meio de oficinas nas escolas, criando vários cenários de ensino-aprendizagem estimulantes. Desenvolver mais jogos educacionais, para estimular outros sentidos do aluno, como por exemplo, jogos de estratégias e aventuras com cenários da atualidade e realidade do aluno.

Podemos elaborar também outros cenários com blocos de programação personalizados para trabalhar com alunos da graduação. Blocos que envolvam comandos mais específicos como, if, for, while e outros, de acordo com um projeto pedagógico a ser criado.

Pretende-se estender o framework para que seja possível criar jogos multiplayer e dessa forma tornar o jogo mais atrativo e competitivo. Também pretendemos programar um ajudante virtual para ajudar e guiar a criança durante o processo do jogo, tornando assim sua experiência mais completa e independente.

Sabemos que os professores precisam oferecer aos alunos alguma forma de experiência prática, contribuindo assim com a diminuição da distância existente entre a educação e as competências demandadas sobre determinado assunto. A utilização de jogos como sendo uma nova estratégia de ensino-aprendizagem nas escolas faz com que melhore o desempenho do aluno como também ajuda ao professor a conhecer as dificuldades do aluno, avaliando seus resultados.

A utilização de jogos digitais como ferramenta de auxílio à transmissão de conhecimento permite estender o conhecimento mais rapidamente. Portanto, vários jogos podem ser desenvolvidos para o uso do professor e aluno como método de buscar enriquecer as aulas. No processo de aprendizagem a utilização de jogos torna-se uma ótima ferramenta para o professor, pois estimula o interesse do aluno, possibilitando a construção do conhecimento a partir das suas interações com os jogos educativos, fazendo com que o jogo seja um aliado no processo de desenvolvimento da criança/aluno.

Por fim, compreendemos que os jogos educativos são excelentes ferramentas que o docente pode utilizar no processo de ensino aprendizagem, visto que eles contribuem e enriquecem o desenvolvimento intelectual e social do educando. No entanto, compreendemos que os mesmos não podem ser utilizados como únicas

estratégias didáticas, pois não garantem a apropriação de todos os conhecimentos esperados (Leal et al., 2005).

8.1 Artigos Publicados

Em paralelo ao desenvolvimento de cada etapa do trabalho foi possível elaborar alguns artigos. Foram eles:

- Infraestrutura para criação de jogos voltados para o ensino de programação no ensino médio, aceito e apresentado no III Workshop de Teses e Dissertações do CBSoft (WTDSoft 2013). Um evento permanente associado ao Congresso Brasileiro de Software: Teoria e Prática (CBSoft), dedicado à apresentação e à discussão de trabalhos de mestrado e doutorado em andamento nas áreas de Engenharia de Software, Linguagens de Programação e Métodos Formais.

- Ensino de programação utilizando jogos digitais - uma revisão sistemática da literatura, aceito e apresentado na Revista Novas Tecnologias na Educação que tem por objetivo publicar trabalhos desenvolvidos na área da Informática na Educação.

- Um framework para criação de jogos voltados para o ensino de lógica de programação, aceito e apresentado no 25º Simpósio Brasileiro de Informática na Educação (SBIE) que tem como objetivos divulgar a produção científica nacional nesta área e proporcionar um ambiente para a troca de experiências e ideias com profissionais, professores, estudantes e pesquisadores nacionais e estrangeiros. O SBIE é dirigido à comunidade de Informática na Educação e pesquisadores de áreas afins

8.2 Colaboradores do projeto

O projeto de desenvolvimento desse framework contou com a colaboração de alguns alunos que fizeram parte do laboratório de pesquisa da GamEdu[1], localizado na Universidade Federal do Rio Grande do Norte (UFRN).

[1] www.gamedu.com

A aluna de mestrado do Departamento de Informática e Matemática Aplicada (DIMAp) da UFRN, Pedrina Brasil, ajudou com o desenvolvimento da API utilizada no framework. E alguns alunos do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte (IFRN), ajudou na construção dos cenários dos jogos desenvolvidos pela ferramenta Construct 2.

REFERÊNCIAS

AFONSO, N (2005) Investigação Naturalista em Educação, Porto, Edições ASA

ALICE. [Online]. Available: http://www.alice.org/index.php

BALASUBRAMANIAN, Nathan; WILSON, Brent G. Games and Simulations. In: SOCIETY FOR INFORMATION TECHNOLOGY AND TEACHER EDUCATION INTERNATIONAL CONFERENCE, 2006. Proceedings… v.1. 2006. Disponível em: <http://site.aace.org/pubs/foresite/GamesAndSimulations1.pdf>. Acesso em: 07 Jan. 2013.

BARROS, Edson de A. R., ZAMBONI, Lincoln C., PAMBOUKIAN, Sergio V. D., Ensino de Computação para estudantes de Engenharia, no COBENGE 2004. Disponível em: < http://meusite.mackenzie.com.br/edsonbarros/publicacoes/BARROS_Edson_Ensino_de_Computacao.pdf > Acesso em: 04 jun. 2012.

BECTA. Computer Games in Education Project. Coventry: BECTA, 2001. Disponível em:<http://partners.becta.org.uk/index.php?section=rh&rid=13595>.Acesso em 27 jan. 2013.

BITTENCOURT, J. R., "Mini-curso - Promovendo a lucidicadade através de jogos livres". XVI Simpósio Brasileiro de Informática na Educação, 2005.

CEDER, V. and YERLER, N. Teaching Programming with Python and PyGame. In: PyCon, 2003.

CODE HUNT. http://olhardigital.uol.com.br/noticia/42031/42031

DADRIX. Por que usar o Constrcut 2. 2012. Disponível em: http://dadrix.com.br/por-que-usar-o-construct-2/>. Acesso em: 08 fev. 2013.

FABRICATORE, C. Learning and videogames: An unexploited synergy. In: INTERNATIONAL CONFERENCE OF THE ASSOCIATION FOR EDUCATIONAL COMMUNICATIONS AND Flatschart, F. Desenvolvimento Web: HTML5, CSS3 e JavaScript. 2011. Disponível em: <http://www.slideshare.net/fabioflat/html5-code>. Acesso em: 29 jan. 2013

FORTUNA, T. R. Sala de aula é lugar de brincar? In: XAVIER, M.L.F. e DALLA

ZEN, M.I.H. Planejamento: análises menos convencionais. Porto Alegre: Mediação, 2000 (Cadernos de Educação Básica, 6) p. 147- 164.

FUNDAÇÃO PENSAMENTO DIGITAL. Em discussão o ensino de programação de computador nas escolas. 2010. Disponível em:<http://www.pensamentodigital.org.br/>. Acesso em: 25 set. 2013

FUNSAT. http://www.inovacaotecnologica.com.br/noticias/noticia.php?artigo=jogo-online-explora-habilidades-jogador-montar-chips#.U4p3CPldU9w

GEE, J. P. What Video Games Have to Teach Us About Learning and Literacy. Second Edition: Revised and Updated Edition. 2nd . Palgrave Macmillan, 2007. ISBN 1403984530.

GROS, Begoña. The impact of digital games in education. First Monday, v. 8, n. 7, jul. 2003. Disponível em: <http://www.firstmonday.org/issues/issue8_7/xyzgros/index.html>. Acesso em: 22 jan. 2013.

HINE, C. Virtual ethnography. Thousand Oaks, CA: SAGE Publications, Inc. 2000.

HSIAO, Hui-Chun. A Brief Review of Digital Games and Learning. DIGITEL 2007, The First IEEE International Workshop on Digital Game and Intelligent Toy Enhanced Learning. Los Alamitos, CA, USA.

HUANG, S.; DISTANTE, D. On Practice-Oriented Software Engineering Education. In: CONFERENCE ON SOFTWARE ENGINEERING EDUCATION & TRAINING WORKSHOPS – CSEETW '06, 2006, Turtle Bay. PROCEEDINGS … Washington: IEEE Computer Society, 2006.

JOHNSON, S. Everything Bad is Good for You. Riverhead Trade, 2006.

KIETCHENHAM, B. A. Guidelines for performing Systematic Literature Reviews in Software Engineering. 2007.

KIRRIEMUIR, John; MCFARLANE, Angela. Literature Review in Games and Learning. Bristol: Futurelab, 2004. 39 p. Disponível em: <http://www.futurelab.org.uk/resources/publications_reports_articles/literature_reviews/Literature_Review378>. Acesso em 20 jan. 2013.

KLOGO-TURTLE. http://styx.nied.unicamp.br/ucanaunicamp/producao-materiais/classmate/aplicativos-do-classmate-tutoriais/klogo-turtle/view

Leal, T.; Albuquerque, E.; Leite, T. (2005) “Jogos: alternativas didáticas para brincar alfabetizando (ou alfabetizar brincando?)”. In: MORAIS, A. G.; ALBUQUERQUE, E. B. C e LEAL, T. F. Alfabetização: apropriação do sistema de escrita alfabética. Recife, PE: Autêntica, 2005.

LIGHT-BOT.
https://play.google.com/store/apps/details?id=com.lightbot.lightbot&hl=pt-br

MCFARLANE, Angela; SPARROWHAWK, Anne; HEALD Ysanne. Report on the educational use of games: An exploration by TEEM of the contribution which games can make to the education process. 2002. Disponível em: <http://www.teem.org.uk/publications/teem_gamesined_full.pdf>. Acesso em: 29 jan. 2013.

MCSHAFFRY, M., GRAHAM, D. Game Coding Complete, 4ºEdição. 2013

MEDEIROS, V. Jogos Educacionais = Chatice, Certo?. Disponível em: <http://consoleacademico.wordpress.com/2010/04/19/jogos-educacionais-chatice-certo-at-ento-era/>. Acesso em: 02 mai. 2014.

MENDES, T. G. Games e Educação: Diretrizes de projeto para jogos digitais voltados à aprendizagem. Porto Alegre, 2012. Disponível em: https://www.lume.ufrgs.br/bitstream/handle/10183/61009/000860539.pdf?sequence=1. Acesso em: 02 mai. 2014.

MENEZES, C. S. (Org.). Informática Educativa II - Linguagens para Representação do Conhecimento. Vitória: UFES, 2003 Fascículo usado em cursos de graduação do NEAD/CREAD/UFES.

MIT Media Lab, 2013. Disponível em: <http://scratch.mit.edu/>. Acesso em: 06 fev. 2013

MITCHELL, Alice; SAVILL-SMITH, Carol. The use of computer and video games for learning: A review of the literature. Londres: Learning and Skills Development Agency (LSDA), 2004. Disponível em: <htttp://www.lsda.org.uk/files/PDF/1529.pdf>. Acesso em 20 jan. 2013.

MORATI, P. B. (2003) “Por que utilizar jogos educativos no processo de ensino aprendizagem?” Rio de Janeiro: NCE/UFRJ. Disponível

em:<http://www.nce.ufrj.br/ginape/publicacoes/trabalhos/PatrickMaterial/TrabfinalPatrick2003.pdf.> Acesso em: mar. 2013.

MULLER, N. Framework, o que é e para que serve?, 2008. Disponível em: http://www.oficinadanet.com.br/artigo/1294/framework_o_que_e_e_para_que_serve. Acesso em: 18 jul 2014.

PAPERT, S. Teaching children thinking (AI Memo No.247 andLogo Memo No. 2).Cambridge, MA: MIT ArtificialIntelligence Laboratory. 1970.

PETRÓ, G. Game "Conflitos globais" tenta quebrar paradigma e colocar jogos na escola. Disponível em:< http://g1.globo.com/Noticias/Games/0,,MUL1534669-9666,00-GAME+CONFLITOS+GLOBAIS+TENTA+QUEBRAR+PARADIGMA+E+COLOCAR+JOGOS+NA+ESCOLA.html> . Acesso em: 02 mai. 2014.

PRENSKY, M. Digital Game-Based Learning. 2 ed. Paragon House Publishers, 2007.

PRENSKY, M. Don't Bother Me Mom--I'm Learning! Paragon House Publishers, 2006.

PRIETO, Lilian Medianeira et al. Uso das Tecnologias Digitais em Atividades Didáticas nas Séries Iniciais. Renote: revista novas tecnologias na educação, Porto Alegre, v. 3, n. 1, p.1-11, maio 2005. Disponível em: <http://www.cinted.ufrgs.br/renote/maio2005/artigos/a6_seriesiniciais_revisado.pdf>. Acesso em: 26 jan 2013.

RAABE, A. L. A.; SILVA, J. M. C. Um Ambiente para Atendimento as Dificuldades de Aprendizagem de Algoritmos. In: XXV Congresso da Sociedade Brasileira de Computação. São Leopoldo/RS. 2005.

RECUERO, R. Jogos e Práticas sociais no facebook: um estudo de caso da máfia wras. Disponivel em: http://www.raquelrecuero.com/arquivos/livrofalcaofinal.pdf

REWALD, F. (2010) Escolas e estudantes desenvolvem games educativos. Folha de São Paulo, 11 jan. 2010. Disponível em: http://www1.folha.uol.com.br/folha/informatica/ult124u677394.shtml. Acesso em: 27 jan. 2010.

RITCHIE, D.; DODGE, B. Integrating Technology Usage across the Curriculum through Educational Adventure Games. In: Anais... . p.10. Houston. Restaurado de <http://eric.ed.gov/ERICWebPortal/custom/portlets/recordDetails/detailmini.jsp?_nfpb=

true&_&ERICExtSearch_SearchValue_0=ED349955&ERICExtSearch_SearchType_0=no&accno=ED349955, 1992>.

ROLLINGS, A., MORRIS, D. "Game Architecture and Design", Arizona Coriolis, 2000

ROSA, S. M. de S. A importância da prática do jogo na aprendizagem das quatro operações fundamentais com números naturais. Lagarto –SE, 2011. Disponível em: http://pt.slideshare.net/MItalo/artigo-a-importncia-dos-jogos-na-aprendizagem-matemtica-sueli-maria. Acesso em: 27 jan. 2010

SÁ, E. J. V; TEIXEIRA, J. S. F; FERNANDES, C. T. Design de atividades de aprendizagem que usam Jogos como princípio para Cooperação. In: XVIII Simpósio Brasileiro de Informática na Educação – SBIE, São Paulo - SP, Brasil. 2007.

SCIRRA, 2013. Disponível em: http://www.scirra.com/construct2. Acesso em: 08 fev. 2013.

SCRATCH. [Online]. Available: http://scratch.mit.edu/

SENAC, 2014. Games na educação: a batalha está começando. Disponível em: <http://www.ead.sp.senac.br/newsletter/setembro04/entrevista/entrevista.htm>. Acesso em: 02 mai. 2014.

SHELLY, Gary B., CASHMAN, Thomas J., HEBERT, Charles W., Alice 2.0: Introdutory Concepts and Techniques. SHELLY CASHMAN SERIES®, Course Technology, Cengage Learning, Boston Massachusetts, USA, 2007.

SILVA, C. V. P. da., CAMPOS, D. de C., MOURA, D. C. de., NERY, P. GQM Goal – Question – Metric, 2009. Disponível em: < http://www.cin.ufpe.br/~scbs/metricas/seminarios/GQM_texto.pdf >. Acesso em: 02 mai. 2014.

SILVEIRA, Sidnei R.; BARONE, Dante A. C. Jogos educativos computadorizados utilizando a abordagem de algoritmos genéticos. Curso de pós graduação em Ciência da Computação-UFRGS. Disponível http://www.c5.cl/ieinvestiga/actas/ribie98/151.html. Acesso em: 10 fev. 2013.

TAROUCO, L. M. R.; ROLAND, L. C.; FABRE, M. C. J. M.; KONRATH, M. L. P. Jogos educacionais. In: Novas Tecnologias na Educação - RENOTE, v.2, n.1, 2004.

THE CODE.ORG. Every student in every school should have theopportunity to learn computer science. Feb. 2014. Availableat: http://code.org.

VANDEVENTER, Stephanie S.;WHITE, James A. Expert Behavior in Children's Video Game Play. Simulation Gaming, v. 33, n. 1, p. 28-48, 2002. Disponível em: <http://sag.sagepub.com/cgi/content/abstract/33/1/28>. Acesso em: 20 jan. 2013.

YOYOGAMES. Disponivel em: <http://www.yoyogames.com/>