

**INSTITUTO FEDERAL DE MINAS GERAIS
CAMPUS SÃO JOÃO EVANGELISTA**

DAVI HAGAP EMANUEL DA SILVA; WANDREY MATHEUS MOREIRA DA COSTA

**ESTUDO E DESENVOLVIMENTO DE UM JOGO EDUCACIONAL PARA O
ENSINO DE PROGRAMAÇÃO BÁSICA**

SÃO JOÃO EVANGELISTA

2018

DAVI HAGAP EMANUEL DA SILVA; WANDREY MATHEUS MOREIRA DA COSTA

**ESTUDO E DESENVOLVIMENTO DE UM JOGO EDUCACIONAL PARA O
ENSINO DE PROGRAMAÇÃO BÁSICA**

Trabalho de conclusão de curso apresentado ao Instituto Federal de Minas Gerais - *Campus* São João Evangelista como exigência parcial para obtenção do título de Bacharel em Sistemas de Informação.

Orientador: Prof. Me. Bruno de Souza Toledo
Coorientadora: Prof.^a Ma. Karina Dutra de Carvalho Lemos

SÃO JOÃO EVANGELISTA

2018

FICHA CATALOGRÁFICA

S581e
2018

Silva, Davi Hagap Emanuel da; Costa, Wandrey Matheus Moreira da.

Estudo e desenvolvimento de um jogo educacional para o ensino de programação básica. / Davi Hagap Emanuel da Silva; Wandrey Matheus Moreira da Costa. – 2018.

60f.; il

Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação) – Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais – Campus São João Evangelista, 2018.

Orientador: Prof. Me. Bruno de Souza Toledo.

Orientadora: Prof.^a Ma. Karina Dutra de Carvalho Lemo.

1. Jogo educacional. 2. Programação básica. 3. Software educacional. 4. Ferramentas didáticas. I. Silva, Davi Hagap Emanuel da. II. Costa, Wandrey Matheus Moreira da. III. Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais – Campus São João Evangelista. IV. Título.

CDD 794.81536

Elaborada pela Biblioteca Professor Pedro Valério

Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais
Campus São João Evangelista

Bibliotecária Responsável: Rejane Valéria Santos – CRB-6/2907

DAVI HAGAP EMANUEL DA SILVA; WANDREY MATHEUS MOREIRA DA COSTA

**ESTUDO E DESENVOLVIMENTO DE UM JOGO EDUCACIONAL PARA O
ENSINO DE PROGRAMAÇÃO BÁSICA**

Trabalho de conclusão de curso apresentado ao Instituto Federal de Minas Gerais - *Campus* São João Evangelista como exigência parcial para obtenção do título de Bacharel em Sistemas de Informação.

Aprovado em: 29/11/2018

BANCA EXAMINADORA



Orientador: Prof. Me. Bruno de Souza Toledo
Instituto Federal de Minas Gerais - *Campus* São João Evangelista



Coorientadora: Prof.^a Ma. Karina Dutra de Carvalho Lemos
Instituto Federal de Minas Gerais - *Campus* São João Evangelista



Convidado: Prof. Esp. Fernando Henriques Mafra
Instituto Federal de Minas Gerais - *Campus* São João Evangelista

SÃO JOÃO EVANGELISTA

2018

AGRADECIMENTOS

Agradecemos aos nossos familiares e amigos, especialmente aos mais presentes nessa jornada, por todo apoio e incentivo. Aos alunos que fizeram parte dessa pesquisa, aos professores Bruno Toledo, Karina Dutra e Fernando Henrique, que nos acompanharam no desenvolvimento desse trabalho, bem como os demais que fizeram parte da nossa formação e nos incentivaram a prosseguir até aqui.

*A vida é apenas uma visão momentânea das
maravilhas deste assombroso universo, e é
triste que tantos se desgastem sonhando com
fantasias espirituais.*

(Carl Sagan)

RESUMO

Introdução a Programação é uma disciplina tradicionalmente difícil para estudantes iniciantes em cursos com cunho voltado a computação. Isso se dá por fatores como falta de interesse dos alunos, modelo de ensino tradicional, dificuldade na abstração dos conceitos abordados, ou até mesmo pela dificuldade intrínseca da matéria. Os diversos obstáculos encontrados pelos alunos nas disciplinas de programação iniciais dos cursos da área de computação acabam acarretando em números altos de reprovações e desistências nessas disciplinas, que por sua vez pode levar até mesmo ao abandono do curso. Para amenizar esses problemas enfrentados no ensino de programação básica o presente trabalho teve como objetivo o desenvolvimento de um jogo educacional para o ensino de programação básica. O resultado foi aplicado aos alunos do Instituto Federal de Minas Gerais - *campus* São João Evangelista matriculados na disciplina de Algoritmos e Estrutura de Dados I do curso Bacharelado em Sistemas de Informação. A metodologia utilizada na pesquisa é descritiva qualitativa, pois os resultados descrevem os dados obtidos por meio de questionário, que visaram analisar o impacto do jogo na aprendizagem de programação, bem como a experiência pessoal dos alunos na utilização do mesmo. Os resultados obtidos demonstraram que o jogo foi capaz de estimular os alunos, e os mesmos relataram um aumento na capacidade de lembrar e compreender os conceitos básicos de programação.

Palavras-chaves: Jogo educacional. Programação básica. *Software* educacional. Ferramentas didáticas.

ABSTRACT

Programming introduction is a traditionally difficult discipline to beginner students in courses focused on computing. This is due to factors such as student's lack of interest, traditional teaching model, difficulty in the abstraction of the approached concepts, or even because of the intrinsic difficulty of the subject. The several obstacles found by students in the introductory courses of Programming from the Computer Science field end up leading to high number of disapprovals and withdrawals in these courses, which in turn can lead to even dropping out of the course. To mitigate these problems faced in basic programming teaching, the present work had the objective of developing an educational game for the teaching of basic programming. The result was applied to the students of the Instituto Federal de Minas Gerais - Campus São João Evangelista enrolled in the discipline of Algorithms and Data Structure I of the Bachelor's degree in Information Systems. The research was qualitative descriptive, because the results describe the data obtained through a questionnaire, which aimed to analyze the impact of the game in programming learning, as well as the personal experience of students in the use of it. The results showed that the game was able to stimulate students, and they reported an increase in the ability to remember and understand the basic concepts of programming.

Keywords: Educational game. Basic programming. Educational software. Didactic tools.

LISTA DE FIGURAS

| | |
|---|----|
| Figura 1 - Tela inicial | 29 |
| Figura 2 - Menu Principal..... | 30 |
| Figura 3 - Tela Carregar | 31 |
| Figura 4 - Tela de créditos | 32 |
| Figura 5 - Aviso de perda de progresso do botão “Novo Jogo” | 33 |
| Figura 6 - Fase | 34 |
| Figura 7 - Compilador com alguns comandos..... | 35 |
| Figura 8 - Janela de ajuda..... | 36 |
| Figura 9 - Janela de aviso de erro | 37 |
| Figura 10 - Tela de pontuação | 38 |
| Figura 11 - Pergaminho de história | 39 |

LISTA DE TABELAS

| | |
|---|----|
| Tabela 1 - Questionário Pré-Aplicação | 41 |
| Tabela 2 - Questionário Pós-Aplicação | 41 |

LISTA DE GRÁFICOS

| | |
|---|----|
| Gráfico 1 - Análise qualitativa de aprendizado (pré-aplicação)..... | 43 |
| Gráfico 2 - Análise qualitativa de motivação | 44 |
| Gráfico 3 - Análise qualitativa de experiência do usuário..... | 44 |
| Gráfico 4 - Análise qualitativa de aprendizado (pós-aplicação)..... | 45 |

LISTA DE SIGLAS

2D - Bidimensional

3D - Tridimensional

IDE - *Integrated Development Environment*

IFMG-SJE - Instituto Federal de Minas Gerais - *campus* São João Evangelista

JPEG - *Joint Photographics Experts Group*

PIBITI - Programa Institucional de Bolsas de Iniciação em Desenvolvimento Tecnológico e Inovação

PNG - *Portable Network Graphics*

MIT - *Massachussets Institute of Tecnology*

SUMÁRIO

| | | |
|----------|--|-----------|
| 1 | INTRODUÇÃO | 12 |
| | 1.1 ESTRUTURA DA MONOGRAFIA..... | 13 |
| 2 | REFERENCIAL TEÓRICO | 14 |
| | 2.1 ALGORITMOS E PROGRAMAS | 14 |
| | 2.2 DIFICULDADES NO ENSINO DE PROGRAMAÇÃO | 15 |
| | 2.3 FERRAMENTAS DIDÁTICAS NO ENSINO DE PROGRAMAÇÃO | 17 |
| | 2.4 JOGOS EDUCACIONAIS NO ENSINO DE PROGRAMAÇÃO..... | 18 |
| | 2.5 ETAPAS DO DESENVOLVIMENTO DE JOGOS | 19 |
| | 2.5.1 Game design document | 19 |
| | 2.5.2 Level design | 20 |
| | 2.5.3 Criação/animação | 21 |
| | 2.5.4 Programação | 21 |
| | 2.6 FERRAMENAS UTILIZADAS | 22 |
| | 2.6.1 Adobe Photoshop CC | 22 |
| | 2.6.2 Blender | 22 |
| | 2.6.3 Unity | 23 |
| | 2.6.4 C sharp | 23 |
| | 2.6.5 MonoDevelop | 23 |
| | 2.7 TRABALHOS CORRELATOS | 24 |
| 3 | METODOLOGIA | 26 |
| | 3.1 NATUREZA DA PESQUISA..... | 26 |
| | 3.2 POPULAÇÃO E AMOSTRA | 26 |
| | 3.3 INSTRUMENTOS UTILIZADOS | 27 |
| | 3.4 MÉTODOS E PROCEDIMENTOS..... | 28 |
| | 3.5 TRATAMENTO DOS DADOS..... | 28 |
| | 3.6 PROTOTIPAÇÃO..... | 29 |
| 4 | RESULTADOS E DISCUSSÃO | 40 |
| | 4.1 ANÁLISE QUALITATIVA DO USO DO SOFTWARE EDUCACIONAL | 40 |
| | 4.2 RESULTADOS DOS DADOS DO QUESTIONÁRIO PÓS APLICAÇÃO | 42 |
| 5 | CONSIDERAÇÕES FINAIS | 46 |
| | REFERÊNCIAS | 47 |
| | APÊNDICES | 52 |
| | APÊNDICE A - Game Design Document | 52 |

1 INTRODUÇÃO

Introdução a Programação é uma disciplina tradicionalmente difícil para estudantes iniciantes, sendo uma área de pesquisa bastante explorada, embora ainda sem soluções conclusivas (PEARS *et al.*, 2007).

Nos cursos com cunho voltado ao desenvolvimento de *software*, uma das peculiaridades negativas é a elevada taxa de evasão dos alunos, principalmente após conhecidas as barreiras nas disciplinas de programação, tais como a falta de contato com a elaboração de algoritmos e a má interpretação dos problemas propostos (RAPKIEWICZ *et al.*, 2006).

De acordo com Gomes *et al.* (2008), isto acontece devido à dificuldade em realizar uma análise do problema e a criação de sua solução, normalmente ocasionado pela falta de compreensão do problema mencionado. Logo, o aluno leigo, repentinamente se depara com um enunciado inicialmente complexo, exigindo a formulação de um algoritmo para a resolução de determinado problema, o que resulta em uma experiência não muito factível de interesse, ou que, no mínimo poderia ser otimizada de forma a fomentar a interação dos alunos com a disciplina.

Identificada essa dificuldade dos alunos em matérias de programação, torna-se necessário o desenvolvimento de ferramentas didáticas que sejam capazes de apoiar o professor, ensinando e principalmente incentivando os alunos ao estudo da programação. Mas como apresentar de maneira atrativa e eficiente a lógica de programação a alunos não familiarizados com o desenvolvimento de algoritmos?

Dentre as ferramentas didáticas desenvolvidas para suprir essa demanda estão os jogos educacionais. O uso de jogos no meio educacional é amplamente estudado e diversas pesquisas foram realizadas ressaltando a importância do mesmo na facilitação do aprendizado assim como seus benefícios no aspecto cognitivo. Tarouco *et al.* (2004), define o jogo como uma ginástica mental, aumentando a flexibilidade cognitiva, aumentando as redes de conexões neurais e alterando o fluxo sanguíneo no cérebro em estados de concentração. Ainda segundo os autores, “a utilização de jogos computadorizados na educação proporciona ao aluno motivação, desenvolvendo também hábitos de persistência no desenvolvimento de desafios e tarefas”.

Para Rapkiewicz *et al.* (2006, p. 4), a simulação trazida pelos jogos também auxilia na visualização do problema por parte do aluno: “O jogo permite tornar real o que seria apenas

imaginário no caso de enunciados de problemas apresentados no quadro ou em listas de exercício em papel”.

Diante do supracitado, este projeto propôs o desenvolvimento de um jogo educacional para estudo de programação básica que, através do aspecto lúdico foi capaz de incentivar e motivar os alunos ao estudo e construção de algoritmos, bem como seu aprendizado na disciplina de Introdução à Programação, disponibilizando ao professor mais um recurso didático.

Assim, o objetivo geral deste trabalho consiste em desenvolver um jogo interativo e dinâmico com cunho educacional capaz de entreter e atrair os alunos no ensino de lógica de programação básica. Para alcançar o objetivo geral foram levantados os seguintes objetivos específicos: a) familiarizar os alunos à lógica de programação e incentivá-los ao estudo e aprofundamento da mesma; b) auxiliar o professor no sentido de atrair a atenção dos alunos para a disciplina; c) minimizar as dificuldades encontradas por alguns alunos no estudo de programação; d) tornar as aulas da disciplina de Introdução a Programação mais dinâmicas, interativas e, conseqüentemente, mais interessantes para os alunos e; e) diminuir a incidência de reprovações e desistências na disciplina.

1.1 ESTRUTURA DA MONOGRAFIA

Este trabalho está organizado em cinco capítulos. O primeiro apresenta uma introdução e os objetivos da pesquisa. O segundo contempla o referencial teórico e apresenta os principais conceitos e abordagens sobre o tema. O terceiro capítulo descreve a metodologia aplicada na realização da pesquisa. O quarto contém os resultados e as discussões do trabalho. E por fim, o quinto e último capítulo destaca as considerações finais.

2 REFERENCIAL TEÓRICO

Lakatos e Marconi (2010) definem a fundamentação teórica como o levantamento de conceitos teóricos que tem como objetivo a elaboração do estudo e seu embasamento, realizando uma revisão literária do tema. Esta sessão abordará informações que fundamentam o trabalho e as discussões na literatura que abrangem os algoritmos e programas de computadores, as dificuldades encontradas nas disciplinas de programação, as ferramentas didáticas desenvolvidas e utilizadas nesse âmbito, o uso de jogos educacionais para o ensino de programação e lógica de algoritmo bem como as etapas do desenvolvimento de um jogo educacional.

2.1 ALGORITMOS E PROGRAMAS

A palavra algoritmo é muitas vezes associada diretamente a computação, entretanto, Medina e Fertig (2005) afirmam que existem várias definições possíveis para a mesma. Os autores designam a origem da palavra ao nome do matemático iraniano Abu Abdullah Mohammad Ibn Musa al-Khawarizmi, e exemplifica seu uso em outras áreas como em receitas culinárias e nas ações para se fazer um balancete, na administração. Os autores utilizam duas definições: Algoritmo é “um procedimento passo a passo para a solução de um problema” e “uma sequência detalhada de ações a serem executadas para executar alguma tarefa”.

Diante das diversas definições de algoritmos, a formalizada na ciência da computação foi a dada por Alan Turing e Alonzo Church em 1936 que define que “um algoritmo é um conjunto não ambíguo e ordenado de passos executáveis que definem um processo finito” (SOUZA *et al.*, 2014). Independentemente da definição utilizada, os algoritmos estão presentes na vida de qualquer pessoa como parte intrínseca de um planejamento para uma atividade qualquer, complexa ou não. Os autores corroboram com a ideia de que os algoritmos são, instintivamente, usados em atividades do nosso dia a dia como lavar um carro, preparar um bolo e montar um guarda roupa, mas os mesmos advertem que o algoritmo não pode ser confundido com a atividade, pois são os passos necessários para realização da mesma.

Os algoritmos são partes essenciais das disciplinas de introdução a programação. Junto com os programas de computador, eles são a porta de entrada para o ambiente de programação que os alunos vivenciam mais à frente no curso.

Com relação aos programas de computadores, Souza *et al.* (2014) define-os como a transcrição dos algoritmos em uma linguagem compreendida por um computador. Geralmente é através dos programas de computador que os estudantes resolvem os problemas propostos pelo professor, principalmente em um estágio mais avançado da disciplina de introdução a programação.

Um bom aproveitamento no aprendizado de algoritmos e programas de computador em disciplinas básicas de programação é enriquecedor para o aluno, pois tais conceitos irão fundamentar boa parte das disciplinas seguintes em cursos de computação. Entretanto, diversas dificuldades são encontradas nessas disciplinas iniciais.

2.2 DIFICULDADES NO ENSINO DE PROGRAMAÇÃO

As disciplinas de programação estão presentes não somente em cursos voltados a computação, mas também compõem a grade de outras áreas do conhecimento, geralmente em um patamar mais básico. Independentemente do curso ou área de conhecimento em que é aplicada, essas disciplinas acabam tendo altos índices de reprovação e/ou evasão.

Confirmando o supracitado, Bennedsen e Caspersen (2007), em um levantamento mundial à cerca dos índices de reprovação e evasão no ensino de programação básica, tiveram como resultado uma taxa de reprovação nas universidades de 33%. A amostra foi de estudantes de universidades e colégios (em menor grupo) em diversos países do mundo.

Na literatura existem diversos estudos sobre as dificuldades nas disciplinas de programação básica em cursos de computação. Alguns autores relacionam os altos índices de reprovação com a má preparação dos alunos. Robins (2010), salienta que capacidade cognitiva, estilos cognitivos, motivação e atitude dos alunos tem sido fatores contribuintes para os altos índices de reprovação e evasão nessas disciplinas. O estudo de Almeida *et al.* (2002) fortalece a ideia de que a falta de interesse dos alunos é um dos fatores responsáveis pelo seu mal desempenho nas disciplinas de programação.

Por outro lado, autores como Borges (2000), se atentam a outros aspectos do processo de ensino-aprendizagem, defendendo a ideia de que o modelo tradicional de ensino é um fator

determinante nos resultados ruins em disciplinas de programação. Segundo o autor, o que mais pesa negativamente no modelo de ensino tradicional é a dificuldade em motivar os alunos ao estudo de algoritmos.

No ensino tradicional o aluno é submetido a constantes repetições do conteúdo ministrado, sendo memorizadas posteriormente por constantes atividades. Segundo Reboul (1982) *apud* Pimenta, Anastasiou (2005) os alunos registram as fórmulas ou palavras ministradas sem compreendê-las de fato.

Existem também análises que englobam mais fatores, como métodos de estudo, métodos de ensino e até mesmo a motivação dos alunos. Esses fatores acabam relacionando uma variedade maior de causas aos resultados ruins. Júnior e Rapkiewicz (2004) confirmam essa ideia afirmando que os desafios do processo de ensino-aprendizagem de fundamentos da programação são de origem questionável, pois engloba alunos, professores, as metodologias aplicadas e até mesmo a dificuldade intrínseca da área.

Corroborando com o supracitado, Borges (2007), em uma análise mais aprofundada sobre o assunto, aponta alguns fatores que contribuem para os altos índices de reprovação nas disciplinas de programação básica como: a) métodos de ensino não personalizados e que estão mais voltados ao ensino da linguagem de programação utilizada do que a capacidade de resolução de problemas, reforçando o estudo de Borges (2000); b) métodos de estudo enviesados que privilegiam memorização ao invés do verdadeiro entendimento do que está sendo passado; c) atitudes e habilidades dos estudantes que não estão acostumados com a abstração necessária para solucionar um problema matemático, bem como sua falta de conhecimento em matemática e lógica; d) natureza da programação, sendo essa de dificuldade intrínseca; e) desmotivação e psicológico dos estudantes conturbado no momento de ingresso em um curso superior, visto que é quando as disciplinas básicas de programação geralmente são ministradas, acaba sendo, por vezes, turbulento.

Diante dos diversos estudos e análises realizadas sobre o processo de ensino-aprendizagem nas disciplinas de programação básica, que descrevem detalhadamente as dificuldades enfrentadas, vem sendo desenvolvidos trabalhos com propostas de auxiliar esse processo por meio de ferramentas didáticas.

2.3 FERRAMENTAS DIDÁTICAS NO ENSINO DE PROGRAMAÇÃO

Por causa da demanda global por profissionais na área da computação, ferramentas didáticas foram desenvolvidas para auxiliar o professor no ensino de programação básica, funcionando como facilitador do processo de ensino-aprendizagem.

Outras áreas do conhecimento já estão presentes nos esforços relacionados ao ensino de programação, como no estudo de Ribeiro *et al.* (2011), que analisa o impacto do uso de robótica no ensino de introdução a programação a alunos do 1º e 2º período dos cursos nos quais essa disciplina compõe a grade. O uso da robótica não só motivou os alunos a encontrar soluções para problemas difíceis como incentivou a colaboração entre os mesmos.

Existem também ambientes computacionais desenvolvidos com o intuito de apoiar o estudante no aprendizado de programação. Essas ferramentas podem ser voltadas a simplificação de uma linguagem específica ou à introdução de conceitos de programação. O VisuAlg é um exemplo de ambiente voltado ao aprendizado da lógica de programação, permitindo a construção e depuração detalhada do pseudocódigo desenvolvido (DE SOUZA, 2009). O autor, em seu estudo sobre a ferramenta VisuAlg, afirma que o uso da mesma pode promover a experimentação e aproximar os alunos ao ambiente de programação mais cedo.

No mesmo contexto do VisuAlg e com o objetivo de simplificar a linguagem Java, a ferramenta *JavaTool*, desenvolvida por Mota *et al.* (2008), oferece uma interface simples e uma animação do que foi codificado, permitindo ao aluno a visualização do que foi feito e, assim, uma adaptação facilitada à linguagem Java.

O trabalho de Cardoso e Antonello (2015) reuniu algumas ferramentas didáticas promissoras no ensino de lógica de algoritmos e obteve resultados iniciais positivos. O trabalho contou com dinâmicas de grupo, a ferramenta VisuAlg, Kits da Placa Arduino e a ferramenta *Scratch*, que será abordada na próxima sessão. A abordagem envolveu as disciplinas de Algoritmos e Oficina da Criatividade, promovendo a interdisciplinaridade e obtendo resultados positivos nas mesmas.

Além dos ambientes computacionais e estratégias didáticas apresentadas, um outro grupo de ferramentas se destaca na literatura no que diz respeito ao ensino de programação básica, os jogos educacionais, que possuem resultados promissores e veem sendo bastante utilizados nesse contexto.

2.4 JOGOS EDUCACIONAIS NO ENSINO DE PROGRAMAÇÃO

Um dos recursos didáticos mais explorados no que tange o ensino de programação básica é o jogo didático. Na literatura, diversos autores relatam sucesso no uso de jogos para, principalmente, motivar os alunos e incentivar o estudo de programação, bem como desenvolver raciocínio lógico e introduzir conceitos básicos de algoritmos.

Segundo Rapkiewicz *et al.* (2006), os jogos computacionais voltados a educação são *softwares* que possuem uma abordagem pedagógica que utiliza a exploração livre e o lúdico, além de possuir conteúdos com objetivos educacionais baseados no lazer e diversão.

Se bem aplicadas, os jogos educacionais podem ser ferramentas eficientes no ensino conforme afirmam Tarouco *et al.* (2004). Os autores também ressaltam que os jogos podem divertir e motivar o estudante enquanto facilitam o processo de aprendizagem.

Um jogo educacional é composto de características específicas que o tornam interessante e motivador para os jogadores. Uma delas é a presença de elementos multimídias que enriqueçam a experiência e imersão e, por consequência, o aprendizado. Silva e Morais II (2011) ressaltam que o uso de diversas linguagens e formas geométricas é capaz de motivar e incentivar a criatividade dos jogadores.

Visando a importância de recurso multimídia nos jogos educacionais, Rapkiewicz *et al.* (2006), afirmam que quando o jogo possui esses recursos, o aluno se sente mais atraído do que pelas formas tradicionais de ensino como questões em papel ou quadro. Os autores salientam que, além da multimídia, o jogo deve oferecer boas simulações, assim o estudante consegue visualizar os passos necessários para atingir um objetivo. Sendo assim, os jogos educacionais são capazes de atrair e motivar os estudantes, por apresentar de forma mais interessante o conteúdo a ser ministrado. Segundo Silva e Morais II (2011), os jogos computacionais na educação auxiliam os alunos a desenvolverem persistência e motivação para vencerem os desafios apresentados nos jogos.

De acordo com Rapkiewicz *et al.* (2006), no ensino de programação, os jogos agem de modo eficiente, ao permitir ao aluno pensar na solução do problema de forma mais descontraída. Quando se joga, o estudante passa a pensar no jogo de forma mais parecida com a codificação de algoritmos. O algoritmo está presente nos jogos e na forma como os estudantes pensam no mesmo e através dele são desenvolvidas as habilidades de raciocínio lógico para a resolução dos problemas. Johnson (2005) corrobora com a ideia de Rapkiewicz

et al. (2006) ao assemelhar a forma de pensar dos jogadores com a dos programadores, sendo ambos uma série de instruções lógicas e sequenciais.

Para a elaboração de um jogo educacional, com recursos capazes de prender a atenção do aluno e fornecer conteúdos didáticos, os desenvolvedores do mesmo devem seguir algumas etapas a fim de não se perderem no desenvolvimento e alcançar os objetivos propostos (ALMEIDA; MEDEIROS, 2008).

2.5 ETAPAS DO DESENVOLVIMENTO DE JOGOS

No desenvolvimento dos jogos, como em qualquer criação de *software*, se faz necessário o uso de um processo de desenvolvimento. Para Almeida e Medeiros (2008) os responsáveis pela execução deste processo se dividem em três grandes áreas: *game design*, arte e programação. Segundo Breyer *et al.* (2006), o *game design* é responsável por toda a parte conceitual e aspectos projetuais do jogo, a equipe de arte é responsável pela apresentação visual e a de programação pela implementação funcional.

Segundo Almeida e Medeiros (2008), para a elaboração de um jogo, o processo de desenvolvimento segue etapas a fim de facilitar o desenvolvimento do mesmo, sendo elas: *game design document*, *level design*, Criação/animação e Programação.

2.5.1 Game design document

A primeira etapa no desenvolvimento de jogos envolve o planejamento e a documentação da ideia e especificações das funcionalidades e jogabilidade do mesmo. Segundo Netto, Machado e Moraes (2006), este passo objetiva não perder o foco principal da ideia, assim como documentar como pretende-se desenvolver esta ideia, quais as necessidades específicas do jogo e o seu público alvo.

Perucia *et al.* (2005) recomendam o uso de duas ações antes de iniciar o desenvolvimento de um jogo: pensar e planejar, sendo que quanto mais dedicado for a fase de planejamento, mais fácil será ter uma visão completa do jogo e dos possíveis problemas que poderão surgir.

Segundo Almeida e Medeiros (2008) e Clua e Bittencourt (2004), nesta etapa são documentados e definidos todos os aspectos que se planeja incluir no jogo, tais como o roteiro, que deverá necessariamente ser interativo, para que permita a interferência do jogador na história, elementos artísticos, tais como detalhes do cenário, esboço dos personagens e objetos, descrições das fases, a descrição da jogabilidade, a interface e regras que regem o jogo.

Para Netto, Machado e Moraes (2006) a síntese da documentação de todos os passos mencionados anteriormente é conhecido como *game design document* ou *design bible*. Rollings e Morris (2004) definem este documento como uma espécie de manual de instruções para os desenvolvedores do jogo. O *game design* mostra ser importante no desenvolvimento de jogos, para guiar os desenvolvedores e evitar possíveis problemas de implementação e ideias de criação.

Após a criação dessa documentação, a próxima etapa é a da concepção das fases, ou *level design*, que seguirá as diretrizes que foram propostas no mesmo.

2.5.2 Level design

Almeida e Medeiros (2008) afirmam que esta etapa do desenvolvimento de jogos é fundamental, principalmente em projetos complexos, com muitas fases e missões, na qual se faz necessário um planejamento das mesmas. Alguns autores como Clua (2007) argumentam que o *level design* ou *game design*, como também é conhecido, devem ser inseridos no *game design document*, devido sua complexidade. Clua e Bittencourt (2004) ressaltam que a documentação dessa etapa deve ser elaborada por um artista.

Segundo Almeida e Medeiros (2008), o *level designer* é quem idealiza cada desafio que o jogador deve superar para passar de uma fase a outra. Clua e Bittencourt (2004) complementam que o *game design* é responsável por toda a conceituação artística do jogo, tais como as características do cenário, esboços dos personagens, descrição das texturas, mapas e descrições das fases. Depois da conceituação artística do jogo, começa o momento da criação dos elementos, sendo esta realizada em outra etapa do projeto conhecida como criação/animação.

2.5.3 Criação/animação

Almeida e Medeiros (2008) definem esta etapa sendo de responsabilidade dos artistas de conceito e animadores, sendo que ambos trabalham juntos, um auxiliando o outro. Nesta parte do desenvolvimento de jogos, é realizada a construção dos elementos tridimensionais, elementos bidimensionais, o desenvolvimento das animações e dos efeitos sonoros.

Para Clua e Bittencourt (2004) a modelagem 3D pode-se dividir em dois tipos, sendo eles a modelagem estrutural, com uma de suas principais características a falta de animação tais como terrenos e objetos estáticos do cenário e a modelagem dinâmica, em que possuem animações como em personagens e objetos animados.

Ainda segundo os autores, os elementos bidimensionais, são utilizados na texturização dos elementos tridimensionais e na integração da interface do jogo, tais como em botões, janelas, barras de energia e outros elementos gráficos. Os autores definem que as texturas são elementos gráficos que representam os materiais que compõem os modelos tridimensionais, tais como madeira, metal, plástico.

Para que as animações e modelos criados nessa etapa possam interagir com o jogador, é necessário um trabalho de programação, que será o responsável por dar vida ao jogo.

2.5.4 Programação

No desenvolvimento de jogos, a programação é essencial, unificando todos os elementos presentes no projeto em um produto. Almeida e Medeiros (2008) complementam que a equipe de programação é responsável pela unificação do projeto a partir da construção mecânica do jogo, com a programação dos sons, a programação das missões e a programação da interface. Para que todas essas etapas de desenvolvimento sejam realizadas com sucesso, é necessário o uso de algumas ferramentas específicas. Essas são abordadas na próxima sessão.

2.6 FERRAMENAS UTILIZADAS

As ferramentas são utilizadas para a criação da arte, da programação e na união de todos os recursos para a finalização e compilação do jogo. Dentre alguns recursos presentes no desenvolvimento de jogos podemos citar o *Adobe Photoshop CC*, *Blender*, *Unity*, *C Sharp* e *MonoDevelop*.

2.6.1 Adobe Photoshop CC

O *Adobe Photoshop CC* é um *software* de edição de imagens com suporte para o *Windows* e *Mac OS*, desenvolvido pela *Adobe Systems*. Segundo o Adobe (2018) este *software* permite a criação de banner, estampas, embalagens, logomarcas, ilustrações bidimensionais e tridimensionais e outros projetos ligados a arte com a manipulação de vários formatos de imagens, tais como o *.png* e *.jpeg*.

2.6.2 Blender

O *Blender 3D* é um *software* gratuito de código aberto multiplataforma desenvolvido pela *Blender Foundation*. O *software* permite a criação de vastos conteúdos 3D, oferecendo funcionalidades completas do *pipeline 3D* (etapas do desenvolvimento de conteúdos tridimensionais), tais como modelagem (construção de um objeto tridimensional), *rigging* (conjunto de controladores que simulam articulações para deformação da malha em uma futura animação do objeto 3D), renderização (processo que gera um produto final de um processamento digital), animação (processo de criação de imagens sequenciais capazes de criar um efeito de movimento), simulação (processo de simulação do ambiente real no ambiente digital), composição (elementos secundários como luz, sombra e objetos que iram a compor a cena) e pós-produção (correções de luz, cor, saturação realizadas no final da produção com o intuito de melhorar as imagens), além de oferecer recursos para edição de vídeos e criação de jogos (BLENDER, 2018).

2.6.3 Unity

O motor de jogos é essencial para o desenvolvimento do mesmo, sendo este o responsável pela criação da física dentro do jogo e a união e interação de todos os elementos presentes na cena. Segundo Brito (2011) o motor do jogo é a alma da animação interativa, sendo este responsável por controlar e permitir as interações com os objetos.

De acordo com o *Unity* (2018), o *software* é um motor de jogos 2D e 3D multiplataforma capaz de gerar produtos finais para *Windows*, *Linux*, *web*, entre outras plataformas. O *software* permite a criação de jogos 2D ou 3D, oferece recursos para a criação de conteúdos de realidade virtual. O *Unity* permite escolher entre duas linguagens disponíveis para a programação dos seus projetos: *Java* ou *C sharp*. A linguagem escolhida para o desenvolvimento desse trabalho foi o *C sharp*, essa escolha deu-se devido à familiaridade prévia dos desenvolvedores com a mesma.

2.6.4 C sharp

Segundo a *Microsoft* (2015), o *C sharp* (C#) é uma linguagem de programação orientada a objetos e foi desenvolvida pela *Microsoft* como parte da plataforma *.NET*. O mesmo foi baseado nas linguagens de programação *C*, *C++* e *Java*, adaptando os melhores recursos presentes em cada linguagem e acrescentando novos recursos importantes para o desenvolvimento de programas, tais como orientação a objetos, elementos gráficos, componentes de interface, multimídia, processamento de banco de dados dentre outros.

2.6.5 MonoDevelop

Segundo o site do *MonoDevelop* (2018), este é um Ambiente de Desenvolvimento Integrado (IDE) multiplataforma, que permite aos desenvolvedores escrever rapidamente aplicativos de desktop e da web no *Linux*, *Windows* e *Mac OS*. A IDE possui suporte para várias linguagens de programação como *C#*, *F#*, *Visual Basic* e *.NET*. Ela também é capaz de criar aplicativos para várias plataformas diferentes.

2.7 TRABALHOS CORRELATOS

A iniciativa de se usar jogos no ensino de programação básica não é nova. Alguns autores já relataram o uso desses recursos e notaram significativo interesse e dedicação por parte dos alunos ao utilizarem essas ferramentas.

Uma das ferramentas mais mencionadas na literatura para o ensino de programação é a ferramenta *Scratch*, desenvolvida pelo *Massachusetts Institute of Technology* (MIT), Instituto de Tecnologia de Massachussets, e disponibilizada gratuitamente na internet. Ela une um ambiente de codificação simples e intuitivo com uma posterior representação visual do que foi codificado. Zanetti *et al.* (2017) aplicaram a ferramenta *Scratch* para crianças de 10 a 13 anos de idade do nível fundamental com o objetivo de desenvolver conceitos de lógica de programação. O ambiente utilizado mescla uma codificação simplificada com uma abordagem visual lúdica e motivadora. Aplicando o *Scratch* a alunos do ensino superior, Oliveira *et al.* (2016), notaram um grande interesse dos alunos em aprender novos comandos, bem como uma assimilação considerável dos conceitos de programação aplicados, como variáveis e estruturas de repetição.

Outra ferramenta bastante similar no que se refere ao modo de funcionamento ao *Scratch* é o *Robocode*, que foi desenvolvido pela *AlphaWorks*. Essa ferramenta também funciona mesclando estes dois ambientes, mas essa utiliza a linguagem Java na codificação e não uma própria como é o caso do *Scratch*. No *Robocode*, programa-se o comportamento de robôs virtuais de batalha e depois acompanha-se seu desempenho contra outros robôs também programados por outras pessoas (AMARAL *et al.*, 2015).

Uma das aplicações do *Robocode* foi a realizada por Amaral *et al.* (2015), que aplicaram a ferramenta para estudantes do ensino médio e perceberam essa abordagem como promissora, em que, o uso do jogo não só motivou os alunos como incentivou a busca pelo conhecimento e a competição.

As ferramentas abordadas anteriormente são apenas as mais mencionadas, existem outros diversos jogos parecidos que seguem a mesma linha de funcionamento com diferenças apenas na quantidade de recursos e/ou simplicidade. A abordagem é quase sempre a mesma: uma aula ou oficina na qual a ferramenta é introduzida à amostra e a partir de então a utiliza.

De acordo com os autores dos trabalhos acima mencionados, os jogos mostraram-se competentes em cumprir os objetivos propostos, prendendo a atenção dos jogadores, divertindo e motivando-os a resolverem os desafios apresentados e fazendo assim com que

eles assimilassem conceitos e noções de algoritmos e programação. O que não se tem relato é se ao longo da disciplina em questão ou no decorrer do curso, os efeitos dessas abordagens se mantiveram, ou seja, se não foi uma abordagem paliativa e momentânea que, a longo prazo, não tivera um impacto significativo. O desenvolvimento de um jogo mais complexo que após aplicado em uma eventual aula ou oficina, manteve o jogador curioso e motivado em desvendar e resolver seus desafios por mais horas de jogatina até então não foi observado. Percebe-se então que, quanto mais exposto a um jogo educacional um aluno for, maior a sua absorção do conteúdo e possivelmente, maior o seu interesse e motivação em estudar e desenvolver algoritmos, cumprindo o jogo seu objetivo educacional de uma maneira ainda mais interessante.

3 METODOLOGIA

De acordo com a linha da metodologia científica definida por Lakatos e Marconi (2010), este capítulo aborda a natureza da pesquisa, bem como a identificação do seu caráter, a população da amostra, os instrumentos utilizados, ferramentas utilizadas, métodos e procedimentos e o tratamento dos dados.

3.1 NATUREZA DA PESQUISA

A pesquisa adotada possui caráter qualitativo, pois seu relatório conterà a definição do uso do jogo educacional na área do objeto de estudo citado anteriormente, bem como a dos alunos. Dalfovo (2008) descreve o método qualitativo sendo trabalhado predominantemente com informações coletadas pelo pesquisador, senda esta não expressa em números, ou quando existentes, esses números e as conclusões neles baseadas representam menor interferência na análise.

A metodologia aplicada neste trabalho foi de caráter descritivo, segundo Gil (2010) o método descritivo tem como objetivo estabelecer relações entre variáveis ou descrever determinadas características de certa população ou fenômeno.

Segundo Gerhardt e Silveira (2009), os pesquisadores que optam por utilizar o método qualitativo, buscam os porquês de um fenômeno ocorrer, transformando esses dados adquiridos em informações aprofundadas e ilustrativas.

3.2 POPULAÇÃO E AMOSTRA

Como população desse estudo, foram utilizados os alunos do curso de bacharelado em Sistemas de Informação do Instituto Federal de Minas Gerais - *Campus* São João Evangelista (IFMG-SJE) e a amostragem, se restringirá aos alunos da disciplina de Algoritmos e Estruturas de Dados I (um), ministrada no segundo semestre do primeiro ano de curso, que participarão de uma capacitação sobre o uso do jogo e futuramente averiguarão a qualidade do

mesmo, bem como o impacto do seu uso no seu desempenho na disciplina em questão e no estudo de programação.

3.3 INSTRUMENTOS UTILIZADOS

As pesquisas realizadas para a concepção desse trabalho foram direcionadas de forma a verificar a viabilidade do desenvolvimento do mesmo através de uma revisão da literatura, que constitui a fundamentação teórica do trabalho e teve como base artigos científicos que justificam e descrevem as potencialidades do uso do jogo educacional no ensino da programação.

Os pesquisadores usaram como base, o projeto de pesquisa interno no IFMG-SJE, cujo objetivo foi desenvolver um jogo educacional para o estudo de programação básica e disponibilizá-lo para o professor da disciplina de Introdução a Programação e aos alunos do curso bacharelado em Sistemas de Informação. Esse projeto foi financiado pelo IFMG-SJE, por meio do Programa Institucional de Bolsas de Iniciação em Desenvolvimento Tecnológico e Inovação (PIBITI) que tem como objetivo estimular estudantes do ensino superior a práticas de desenvolvimento tecnológico e processos de inovação (CNPQ, 2018). Os dados coletados para o desenvolvimento do jogo foram através da análise e levantamento de requisitos feitos com o professor da disciplina.

O presente trabalho utilizou a aplicação de um questionário aos alunos para avaliar sua satisfação com o jogo e seu funcionamento, bem como o impacto do uso do mesmo no domínio do aluno à cerca do objeto de estudo citado anteriormente.

O questionário aplicado aos alunos foi composto por questões abertas e fechadas, que foram os meios básicos para coleta de dados. Essa estruturação dá-se através dos critérios de avaliação de um jogo educacional propostos por De Oliveira Medeiros e Schimiguel (2012): (1) qualidade do conteúdo, relacionado à veracidade e apresentação equilibrada das ideias, (2) alinhamento do objetivo da aprendizagem, (3) motivação, que é a capacidade de motivar os alunos, (4) imersão, a capacidade de envolver o jogador profundamente, (5) objetivos claros, (6) *feedback* e adaptação, elaboração do *feedback* positivo e negativo do jogador e capacidade de se adaptar ao nível de habilidade do mesmo, (7) apresentação, relacionado ao aspecto visual, (8) interação social, disposição de meios de interação com outros jogadores e, (9)

reusabilidade, capacidade de reutilizar o jogo com diferentes alunos que tenham diferentes interesses e/ou idades.

3.4 MÉTODOS E PROCEDIMENTOS

A elaboração e construção do projeto seguiram as etapas do desenvolvimento de jogos, que consistem no *game design document*, *level design*, criação/ animação e programação.

Na etapa do *game design document*, foram realizadas reuniões com o professor da disciplina envolvida no estudo, a fim de entender as necessidades e dificuldades dos alunos. Apresentadas as dificuldades, começou o planejamento do jogo, a interface que foi adotada, a jogabilidade e o roteiro, conforme (APÊNDICE A - Game Design Document).

Posteriormente ao planejamento do jogo, deu-se início a etapa de *level design*, na qual foram elaborados os detalhes de cada cena e os desafios que o jogador deverá enfrentar para conseguir avançar entre os níveis. Nesta etapa utilizando o *software Unity*, foi realizada a construção de cada cena e o posicionamento dos elementos presentes no jogo, tais como os efeitos sonoros, objetos bidimensionais e tridimensionais.

A criação e animação dos elementos necessários para a composição dos cenários, foram realizados pelos *softwares Blender e Photoshop*, sendo o primeiro utilizado para a modelagens de personagens e objetos tridimensionais, e o segundo, *Photoshop*, utilizado na manipulação de imagens que compõe os objetos tridimensionais e na concepção de elementos da interface.

A programação do jogo foi realizada aplicando a linguagem de programação *C sharp*, utilizando a IDE *MonoDevelop* presente no *Unity*. Nesta etapa foram desenvolvidas as interações entre o personagem e os objetos apresentados na cena, os critérios para passar os níveis, as interações entre objetos e efeitos sonoros.

3.5 TRATAMENTO DOS DADOS

Após a realização dos procedimentos, houve o tratamento dos dados coletados na pesquisa através de testes de funcionamento, usabilidade e impacto na disciplina, teste este

que foi realizado pela amostragem definida anteriormente. Os resultados dos testes foram utilizados para gerar um relatório para análise qualitativa, e por fim, depois do projeto pronto e testado foi realizada a publicação dos resultados e disponibilização do *software* desenvolvido no site do instituto.

3.6 PROTOTIPAÇÃO

A prototipação tem por objetivo facilitar a interação do desenvolvedor, o *software* e os possíveis usuários e proporcionar um melhor entendimento do sistema desenvolvido.

A Figura 1 apresenta a tela inicial que apresenta o nome do jogo e a instrução “Pressione qualquer tecla...”.

Figura 1 - Tela inicial

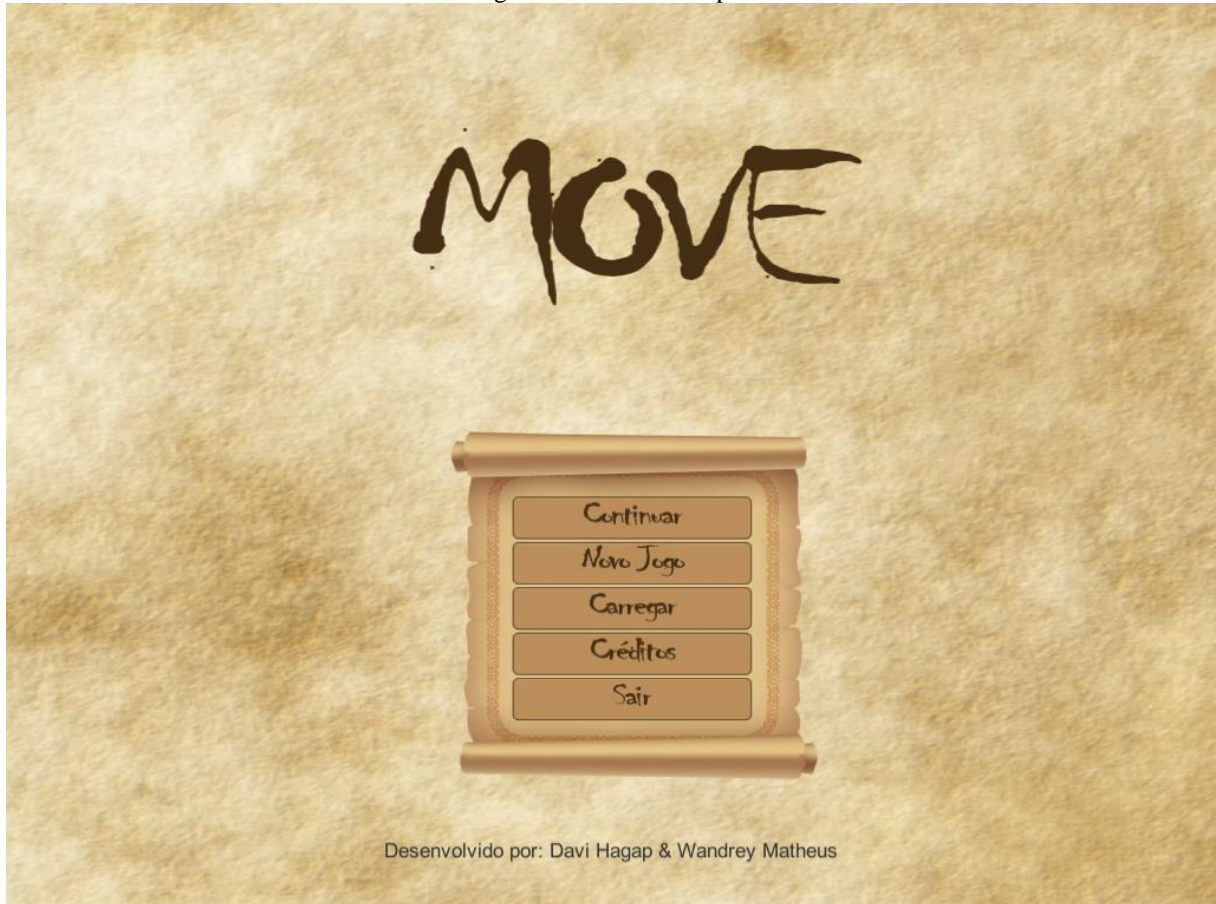


Fonte: Próprio dos autores.

Ao pressionar uma tecla ou realizar um clique, o jogo direciona para o menu inicial. Na Figura 2 tem-se o menu principal apresenta cinco botões, sendo eles: “Continuar”, “Novo

Jogo”, “Carregar”, “Créditos” e “Sair”. A opção “Continuar” só se faz presente após se completar a primeira fase. O botão “Sair” encerra o jogo.

Figura 2 - Menu Principal



Fonte: Próprio dos autores.

Ao selecionar a opção “Carregar”, é iniciada a tela em que o jogador visualiza a sua pontuação nas fases já completadas, bem como jogá-las novamente ao clicar no símbolo de *play* localizado à frente da pontuação. Também é possível retornar ao menu principal por meio do botão “Voltar”, como mostra a Figura 3.

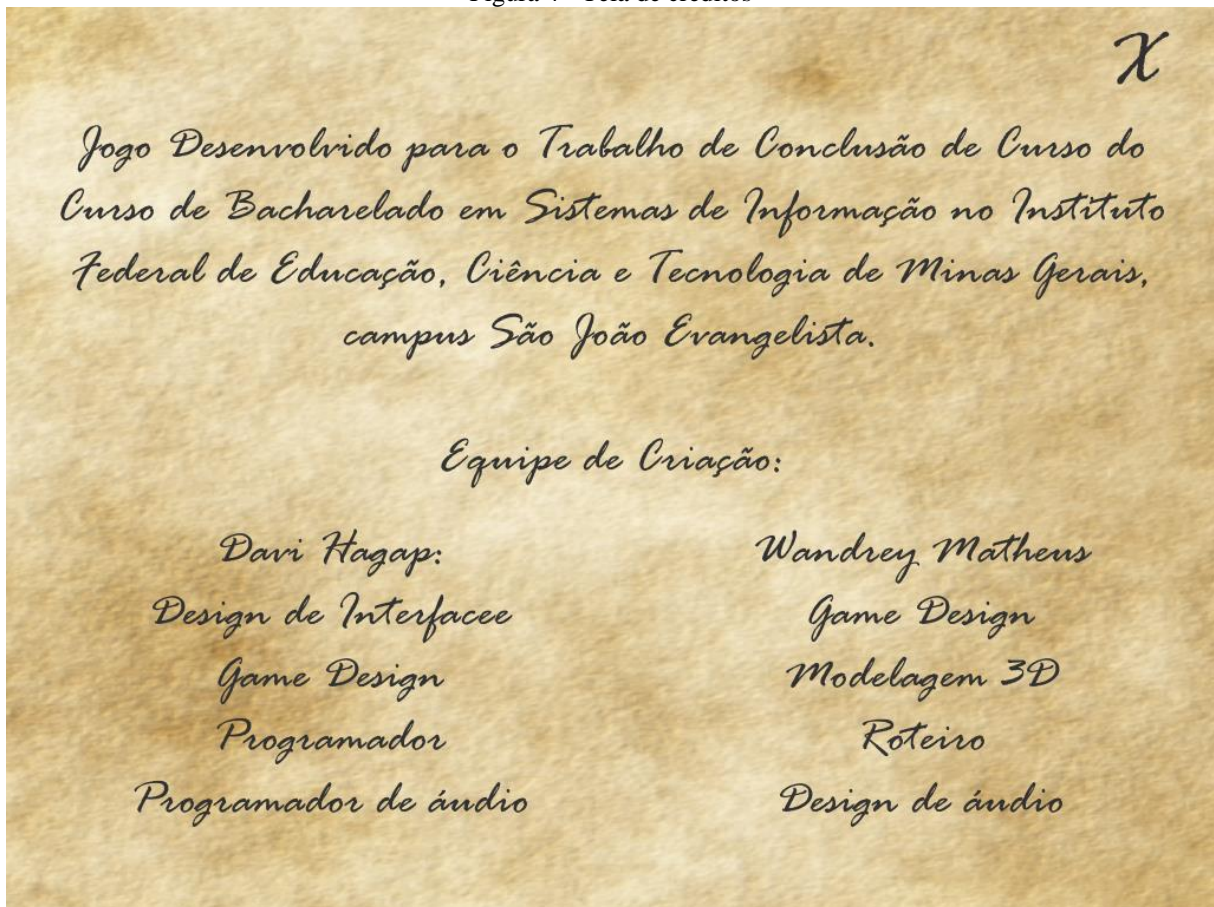
Figura 3 - Tela Carregar



Fonte: Próprio dos autores.

Ainda no menu principal, ao clicar no botão “Créditos” o jogador é direcionado a tela mostrada na Figura 4, que apresenta informações acerca dos desenvolvedores. O botão “X” no lado superior direito leva de volta ao menu principal.

Figura 4 - Tela de créditos



Fonte: Próprio dos autores.

No menu principal, a opção “Continuar” o leva a última fase não completada. Já o botão “Novo Jogo” avisa o jogador que se prosseguir, perderá o progresso adquirido, de acordo com a Figura 5. Se o jogador quiser prosseguir, a primeira fase é carregada. O aviso não é mostrado caso não haja progresso a ser perdido.

Figura 5 - Aviso de perda de progresso do botão “Novo Jogo”



Fonte: Próprio dos autores.

Todas as fases mantêm a interface conforme a Figura 6, sendo a única diferença são os comandos disponíveis, que são desbloqueados conforme se avança no jogo. A interface é composta pelo “Número da fase”, mostrado no lado superior esquerdo, o “Compilador”, pelos “Comandos Disponíveis”, e os botões “Menu”, que leva novamente ao menu principal, “Reiniciar”, que reinicia a fase atual, e o botão de ajuda, representado pelo símbolo “?”.

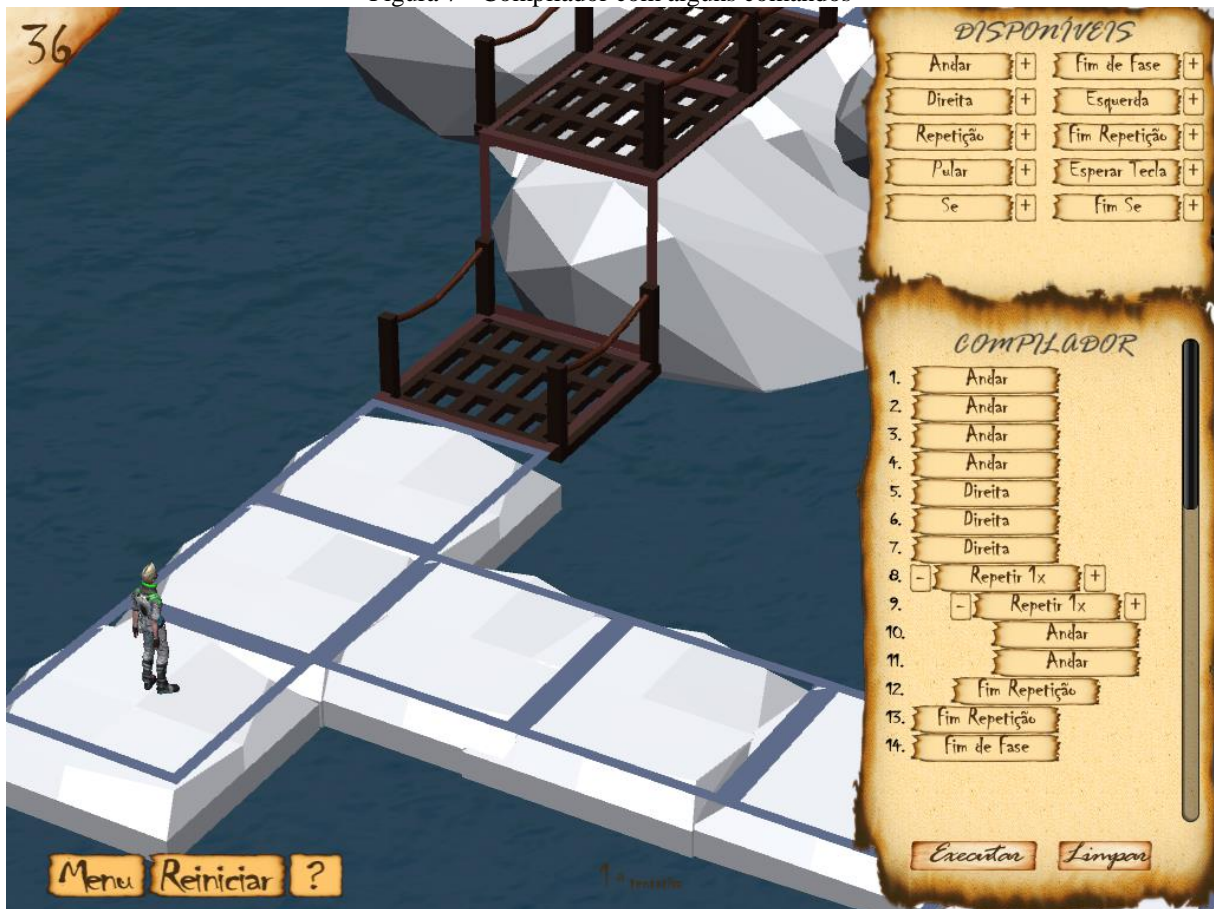
Figura 6 - Fase



Fonte: Próprio dos autores.

O “Compilador” é preenchido clicando-se nos botões “+” do lado de cada um dos “comandos” ou arrastando-os para dentro do mesmo. A Figura 7 exemplifica um “Compilador” com alguns comandos. O “Compilador” ainda contém dois botões, “Executar” e “Limpar”. O botão “Limpar” descarta os comandos adicionados no compilador. Já ao clicar em “Executar”, os comandos adicionados são executados pelo personagem.

Figura 7 - Compilador com alguns comandos



Fonte: Próprio dos autores.

O botão “?” abre uma janela de ajuda com as explicações dos comandos que o jogador pode usar, e também com os controles do jogo, de acordo com a Figura 8. O “X” no lado superior direito fecha a janela de ajuda e volta à fase.

Figura 8 - Janela de ajuda



Fonte: Próprio dos autores.

O jogo apresenta, por meio de janelas de aviso, retornos de erros do jogador. A Figura 9 exemplifica uma dessas janelas, que contém a explicação do erro do jogador e um botão “Ok” que reinicia a fase para que o mesmo tente novamente.

Figura 9 - Janela de aviso de erro



Fonte: Próprio dos autores.

Ao se completar uma fase com sucesso, a tela de pontuação é aberta com a “estrela” recebida naquela fase. De acordo com a Figura 10, o jogador pode continuar para a próxima fase através da opção “Avançar”, jogar a fase novamente ao clicar em “Recomeçar”, ou voltar ao menu principal com a opção “Menu”.

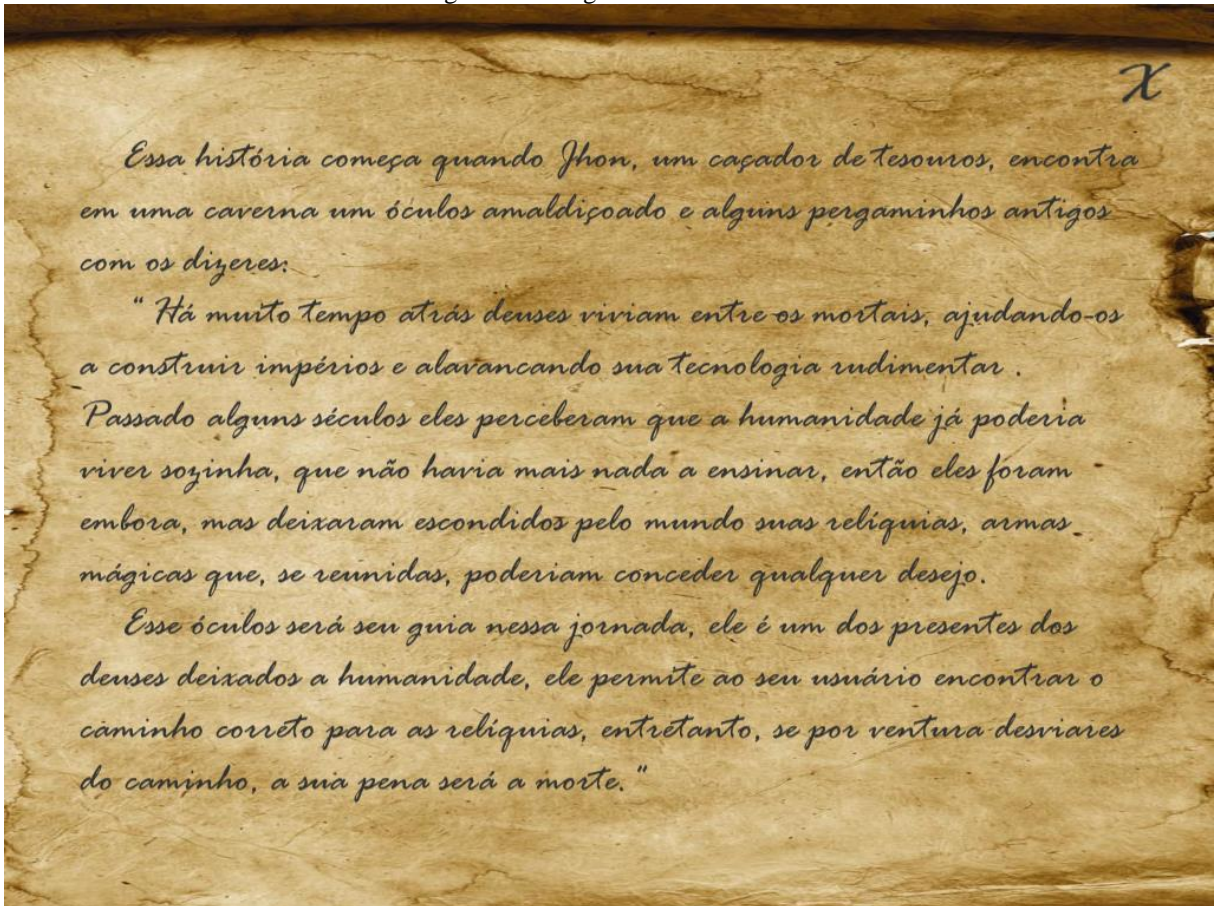
Figura 10 - Tela de pontuação



Fonte: Próprio dos autores.

Em determinadas fases o jogador se depara com “pergaminhos de história”, que tem como objetivo cativar o jogador por meio de elementos de enredo, conforme a Figura 11. O jogador deve clicar no “X” no lado superior direito para prosseguir para a fase.

Figura 11 - Pergaminho de história



Fonte: Próprio dos autores.

Os pergaminhos de história aparecem em momentos de transição de cenário, fornecendo ao jogador uma justificativa da fase em que ele está e fortalecendo a noção de continuidade do jogo.

4 RESULTADOS E DISCUSSÃO

Para apresentação do resultado oriundo dos questionários, foram elaborados gráficos e tabelas, seguidos da análise dos dados obtidos. A análise utilizada na pesquisa qualitativa aplicada aos alunos foi feita de modo descritivo, porque se pretendia analisar a experiência dos mesmos em relação ao uso do jogo educacional.

4.1 ANÁLISE QUALITATIVA DO USO DO SOFTWARE EDUCACIONAL

O instrumento utilizado para a coleta de dados foi composto por escalas de resposta do tipo *Likert* de 5 pontos (5 - Concordo plenamente, 4 - Concordo parcialmente, 3 - Indiferente, 2 - Discordo parcialmente, 1 - Discordo totalmente). Segundo Alexandre *et al.* (2003) a escala *Likert* é dividida geralmente em 4 (quatro) ou 5 (cinco) categorias ordinais. Utilizou-se essa escala para facilitar o entendimento dos respondentes diante a mensuração de sua experiência em relação ao funcionamento do jogo, além de facilitar o tratamento das informações obtidas nos questionários aplicados aos mesmos.

O emprego do jogo educacional como ferramenta didática no ensino de programação básica se deu de forma prática, por meio do uso de suas funcionalidades pelos alunos ao decorrer de uma aula da disciplina de Algoritmos e Estruturas de Dados I, previamente autorizada pela professora ministrante. Ao final do questionário houve a possibilidade de uma sugestão ou crítica à respeito do jogo, a qual colaborou para o aprimoramento do mesmo.

A Tabela 1 apresenta o questionário aplicado aos alunos no momento anterior à aplicação, que foi composto por quatro perguntas fechadas. Esse questionário foi um instrumento de comparação da maneira como os alunos estavam em relação aos assuntos e temas abordados na disciplina e, posteriormente, de como os mesmos estariam depois de submetidos ao jogo, sendo vinte e seis respondentes.

A Tabela 2 exhibe os atributos relativos à utilização do jogo pelos alunos. As variáveis que identificam esses indicadores foram estabelecidas com as iniciais “Q”, sendo que a numeração que as acompanham indica a localização das sentenças no questionário em um total de vinte e uma questões. O questionário da Tabela 2 teve dezoito respondentes, que analisaram a usabilidade e a utilização durante o tempo que foram submetidos ao mesmo.

Tabela 1 - Questionário Pré-Aplicação

| |
|---|
| Q1 O quão bem você consegue se lembrar de informações acerca da lógica de programação |
| Q2 O quão bem você consegue compreender conceitos de lógica de programação |
| Q3 O quão bem você consegue aplicar conceitos de lógica de programação |
| Q4 O quão motivado você se sente em relação ao estudo de programação básica |

Fonte: Elaborado pelos autores.

As questões presentes na Tabela 1 buscaram investigar como os alunos se sentiam em relação ao domínio de conceitos e temas relacionados à programação básica. Todas as questões foram elaboradas como uma adaptação do modelo proposto por Savi *et al.* (2010), mais especificamente no aspecto conhecimento que, segundo o autor, diz respeito ao impacto no aprendizado.

Tabela 2 - Questionário Pós-Aplicação

| |
|---|
| Q1 Você completou o jogo |
| Q2 A interface gráfica é adequada e apresenta boa qualidade |
| Q3 O uso de ilustrações/animações desperta, mantém e reforça a atenção e a motivação |
| Q4 O uso de cor e recursos sonoros desperta, mantém e reforça a atenção e a motivação |
| Q5 O conteúdo do jogo está relacionado com conceitos já adquiridos |
| Q6 O jogo foi mais difícil de entender do que o esperado |
| Q7 O jogo torna o aprendizado mais dinâmico e interessante |
| Q8 Gostou do jogo e não se sentiu ansioso ou entediado |
| Q9 As habilidades melhoraram gradualmente com a superação dos desafios |
| Q10 O jogo é adequadamente desafiador, as tarefas não são muito fáceis nem muito difíceis |
| Q11 Gostaria que o jogo tivesse mais fases |
| Q12 Jogaria novamente |
| Q13 Houve esforço para ter bons resultados no jogo |
| Q14 Sente-se estimulado a aprender com o jogo |
| Q15 Gostaria de aprender mais sobre o assunto abordado pelo jogo. |
| Q16 O quão bem você consegue se lembrar de informações acerca da lógica de programação |
| Q17 O quão bem você consegue compreender conceitos de lógica de programação |
| Q18 O quão bem você consegue aplicar conceitos de lógica de programação |
| Q19 O quão motivado você se sente em relação ao estudo de programação básica |

| |
|--|
| Q20 O quão bem você consegue se lembrar de informações acerca da lógica de programação |
|--|

| |
|------------------------|
| Q21 Sugestões/Críticas |
|------------------------|

Fonte: Adaptado pelos autores.

As questões presentes na Tabela 2 foram feitas com base no modelo proposto por Savi *et al.* (2010), sendo que as questões (Q2 a Q7) foram elaboradas como adaptações ao aspecto motivação proposto pelo autor, e tem como objetivo entender como o jogo afetou o modo como os jogadores se sentiram ao usá-lo. Já as questões (Q8 a Q15) dizem respeito ao aspecto experiência do usuário, e buscam averiguar características e capacidades do próprio jogo em manter a imersão e o divertimento, ter desafios adequados, e apoiar habilidades e competências do jogador (SAVI *et al.*, 2010).

Para que pudesse ser feito um paralelo com os dados obtidos através do primeiro questionário aplicado, as questões (Q16 a Q20) se repetiram no questionário de pós-aplicação.

A primeira questão (Q1) existe somente para controlar a quantidade de pessoas que completaram o jogo, já a questão (Q21) foi o espaço dado aos respondentes para que os mesmos sugerissem critérios de melhorias e ou adaptações no jogo.

4.2 RESULTADOS DOS DADOS DO QUESTIONÁRIO PÓS APLICAÇÃO

Antes da aplicação do jogo aos alunos, foi aplicado o questionário da Tabela 1, os resultados estão expostos no Gráfico 1. Após o questionário, o jogo foi apresentado aos alunos em sua ideia básica e no seu funcionamento, para então ser aplicado durante o restante da aula. Ressaltando que a todo momento se fez necessário, dúvidas acerca da jogabilidade foram esclarecidas.

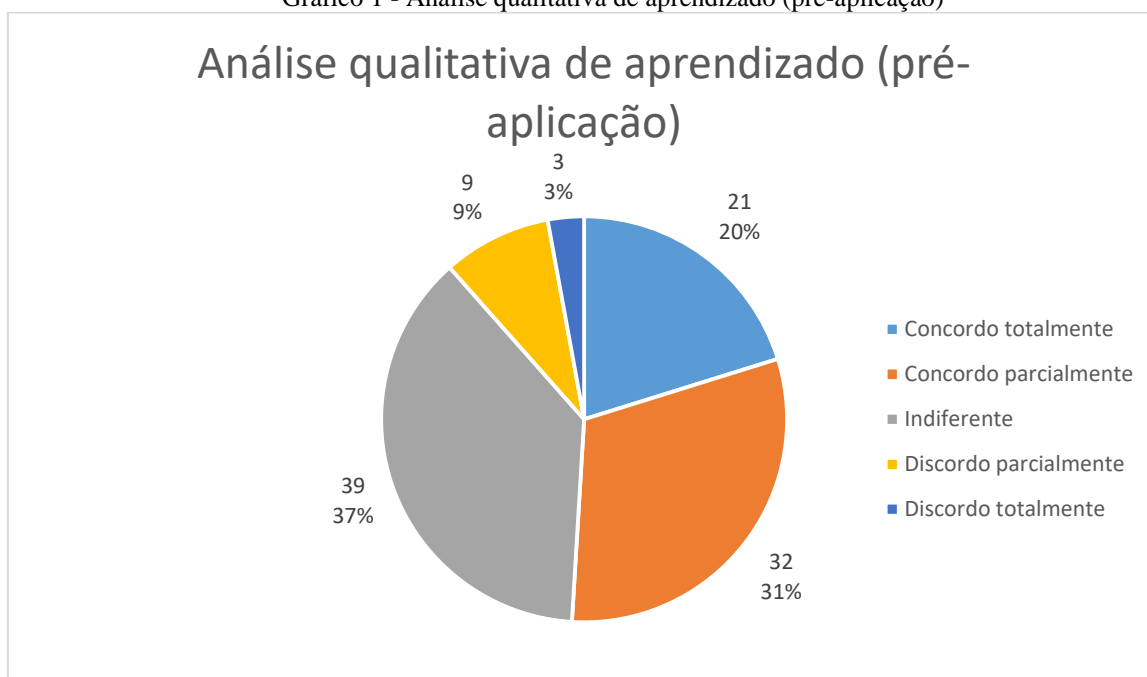
Foi proposto como forma de incentivo uma premiação aos alunos que completassem todas as fases do jogo, em que os mesmos dispuseram do prazo de duas semanas para completá-lo. Foi disponibilizado um link onde os alunos puderam obter o jogo através de *download*, para que assim, eles jogassem em casa.

Findado o prazo estipulado para completar o jogo, foi aplicado aos alunos o questionário da Tabela 2. Os dados coletados indicam que o jogo possui a capacidade de manter o jogador cativado, motivado e entretido, atendendo os requisitos de jogo educacional descritos por Almeida e Medeiros (2008), conforme pode ser visualizado no Gráfico 2.

No que se refere a experiência do usuário, a maioria dos alunos afirmaram que gostaram do jogo, se esforçaram para obter bons resultados e jogariam novamente. Relataram também que sentiram as habilidades melhorarem gradualmente com a superação dos desafios propostos e estimulados a aprender com o jogo e sobre o assunto tratado, e que o mesmo é adequadamente desafiador e que gostariam que houvessem mais fases. Esses dados são demonstrados no Gráfico 3.

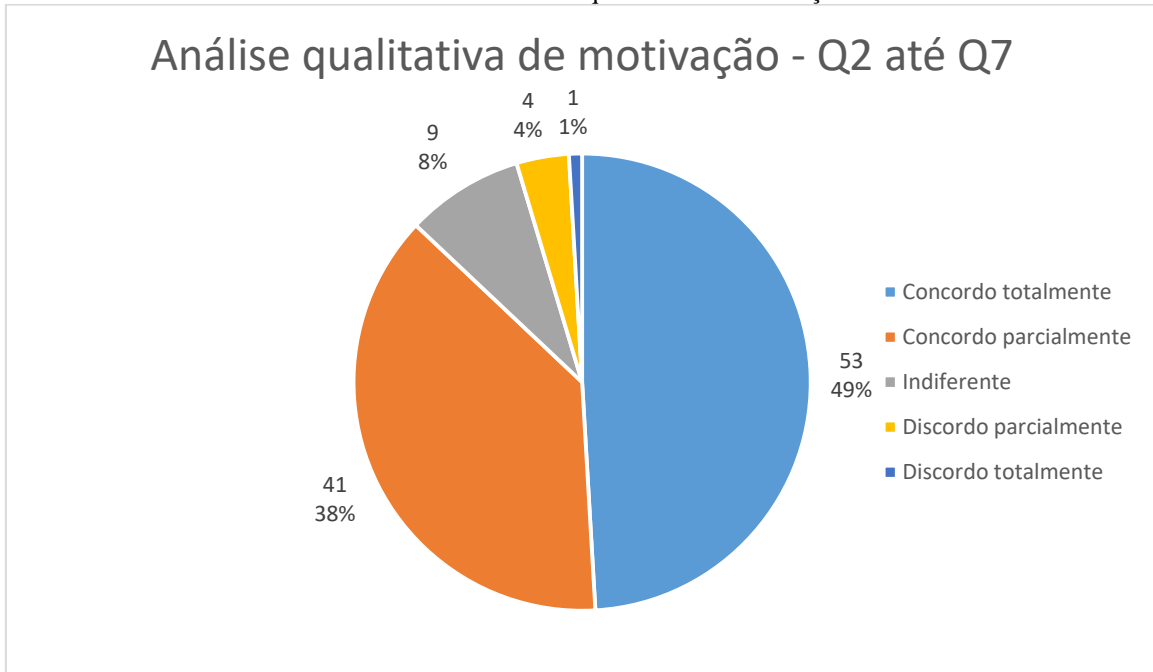
Apesar dos resultados positivos, a taxa de conclusão do jogo foi de apenas 44,4 %, devido ao tempo na qual os alunos tiveram contado com o mesmo.

Gráfico 1 - Análise qualitativa de aprendizado (pré-aplicação)



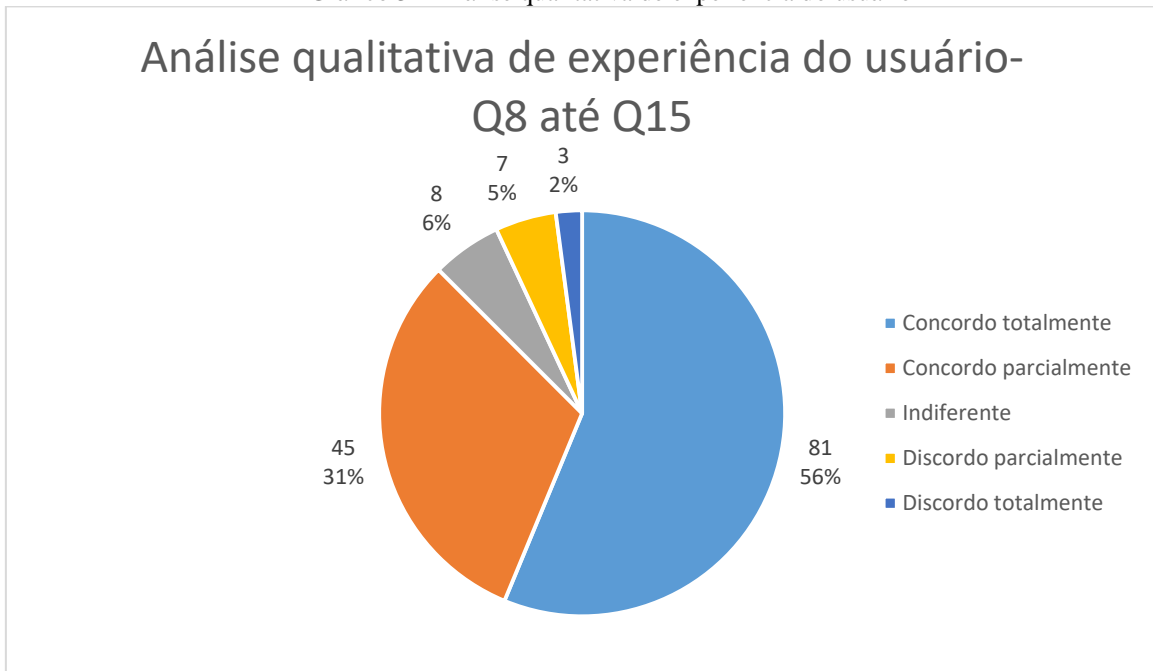
Fonte: Elaborado pelos autores.

Gráfico 2 - Análise qualitativa de motivação



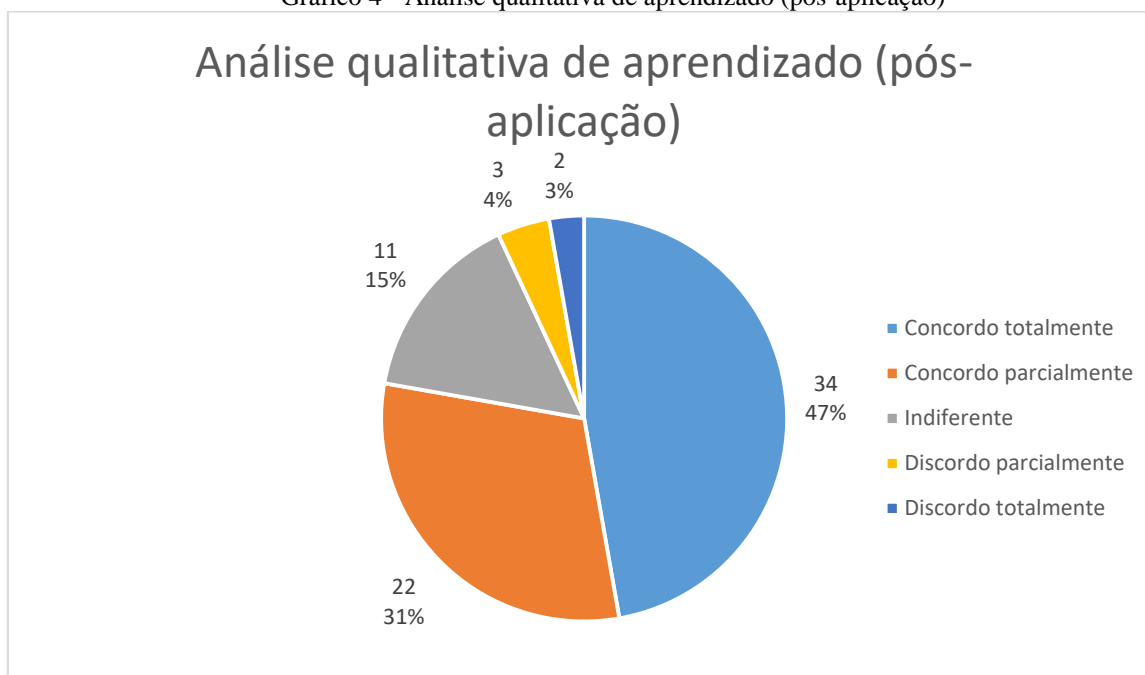
Fonte: Elaborado pelos autores.

Gráfico 3 - Análise qualitativa de experiência do usuário



Fonte: Elaborado pelos autores.

Gráfico 4 - Análise qualitativa de aprendizado (pós-aplicação)



Fonte: Elaborado pelos autores.

Ainda é possível observar que o jogo teve um impacto positivo no aprendizado dos alunos. Comparando-se os dados obtidos antes e depois da aplicação do jogo (Tabela 1 e Tabela 2, respectivamente), houve um aumento de 27 % no número de alunos que afirmaram se lembrar, compreender e aplicar conceitos de lógica de programação básica, bem como se sentiram motivados em relação ao estudo da mesma. Esses dados podem ser verificados comparando-se o Gráfico 1 com o Gráfico 4 e corroboram com Rapkiewicz *et al.* (2006), Tarouco *et al.* (2004) e Silva e Morais II (2011) que afirmam que os jogos educacionais tem a capacidade de motivar e atrair os alunos enquanto facilitam o processo de aprendizagem.

5 CONSIDERAÇÕES FINAIS

O jogo Move está em funcionamento e poderá ser obtido através do site do *campus* <https://www.sje.ifmg.edu.br>. Todos os objetivos e etapas propostas nessa pesquisa foram realizadas, todavia existem melhorias a serem feitas.

A área de pesquisa foi enriquecida, visto que através dos dados obtidos, pode-se observar o impacto do Move no ensino de programação básica.

Foi constatada grande aceitação do jogo por parte dos alunos e o impacto à longo prazo mostrou que o mesmo contribuiu bastante para o aprendizado. Entretanto, não houve uma taxa tão grande de conclusão do jogo, dado o pouco tempo de exposição e/ou o momento em que o mesmo foi aplicado aos alunos, o que pode ter contribuído para este aspecto negativo.

Como proposta de trabalho futuro, pode-se realizar a aplicação do Move à alunos da disciplina Introdução à Programação, na qual os alunos mais inexperientes poderiam ser mais beneficiados com o uso do mesmo, visto que os conceitos abordados nele são de caráter básico e introdutório. Outra melhoria seria o desenvolvimento do jogo para a plataforma móvel, tais como *tablets* e celulares, no qual se mostram bastante presente na vida dos estudantes, facilitando o acesso ao jogo e assim aproximá-los ao tema abordado.

REFERÊNCIAS

- ADOBE. **Reimagine a realidade**. 2018. Disponível em: <<https://www.adobe.com/br/products/photoshop.html>>. Acesso em: 10 maio 2018.
- ALEXANDRE, J. W. C., ANDRADE, D. D., VASCONCELOS, A. D., ARAÚJO, A. D. e BATISTA, M. J. Análise do número de categorias da escala de Likert aplicada à gestão pela qualidade total através da teoria da resposta ao item. In: **Encontro Nacional De Engenharia De Produção**, v. 23, p. 1-20, 2003.
- ALMEIDA, E. S., COSTA, E. B., BRAGA, J. D. H., SILVA, K. S., PAES, R. B. e ALMEIDA, A. A. M. AMBAP: Um Ambiente de Apoio ao Aprendizado de Programação. In: X Workshop sobre Educação em Computação (WEI 2002), 2002, Florianópolis- SC. **Anais...** Florianópolis- SC: WEI, 2002; SBC, 2002.
- ALMEIDA, Janiel Henrique Pinheiro; MEDEIROS, Wander Klaysson Aparecido. Mobile Games: Etapas de desenvolvimento e especificidades. In: **IV Seminário Jogos eletrônicos, educação e comunicação: construindo novas trilhas**. 2008.
- AMARAL, Laurence *et al.* Plataforma robocode como ferramenta lúdica de ensino de programação de computadores-pesquisa e extensão universitária em escolas públicas de minas gerais. In: **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)**. 2015. p. 200.
- BENNEDSEN, Jens; CASPERSEN, Michael. (2007). Failure rates in introductory programming. **SIGCSE Bulletin**. 39. 32-36. 10.1145/1272848.1272879.
- BLENDER. **Creative freedom starts here**. 2018. Disponível em: <www.blender.org>. Acesso em: 10 maio 2018.
- BORGES, M. A. **Avaliação de uma metodologia alternativa para a aprendizagem de programação**. 2000. Disponível em: <<http://www.niee.ufrgs.br/eventos/SBC/2000/pdf/wei/relatos/selecionados/wei006.pdf>>. Acesso em: 12 maio 2018.
- BREYER, F.B; MOURA, D; PADOVANI, S; NEVES, A. **Definição de Métodos de acompanhamento de qualidade para Game Design**, 2006. Disponível em: <www.cin.ufpe.br/~sbgames/proceedings/aprovados/23634.pdf>. Acesso em: 10 abr. 2018.

BRITO, Allan. **Blender 3D: Jogos e animações interativas**. São Paulo: Novatec, 2011.

CARDOSO, Rogério; ANTONELLO, Sérgio. Interdisciplinaridade, programação visual e robótica educacional: relato de experiência sobre o ensino inicial de programação. In: Workshops do IV Congresso Brasileiro de Informática na Educação (CBIE 2015), 2015. **Anais...** CBIE, 2015, p. 1255.

CLUA, E. W. G. Desenvolvimento de jogos 3D. In: SILVA, E. M.; MOITA, F. M. S.; SOUSA, R. P. (Orgs.). **Jogos eletrônicos: construindo novas trilhas**. Campina Grande: EDUEP, 2007.

CLUA, E. W. G; BITTENCOURT, J. R. Desenvolvimento de Jogos 3d: Concepção, Design e Programação. In: XVI Simpósio Brasileiro de Informática na Educação (SBIE 2005), 2005, São Leopoldo- RS. **Anais...** São Leopoldo- RS: SBC, 2005, pp. 1313-1357.

CLUA, E. W. G; BITTENCOURT, J. R. Uma nova concepção para criação de jogos educativos. In: XV Simpósio Brasileiro de Informática na Educação (SBIE 2004), 2004, Manaus- AM. **Anais...** Manaus- AM: SBC, 2004, pp. 1-34.

CNPQ. **PIBITI - Programa Institucional de Bolsas de Iniciação em Desenvolvimento Tecnológico e Inovação**. 2018. Disponível em: < <http://cnpq.br/apresentacao>>. Acesso em: 14 maio 2018.

DE OLIVEIRA MEDEIROS, Maxwell; SCHIMIGUEL, Juliano. Uma Abordagem para avaliação de jogos educativos: ênfase no ensino fundamental. In: XXIII Simpósio Brasileiro de Informática na Educação (SBIE 2012), 2012, Rio de Janeiro- RJ. **Anais...** Rio de Janeiro- RJ: SBC, 2012.

DE SOUZA, Cláudio Morgado. VisuAlg-Ferramenta de apoio ao ensino de programação. **Revista Eletrônica TECCEN**, v. 2, n. 2, p. 01-09, 2009.

DALFOVO, Michael Samir; LANA, Rogério Adilson; SILVEIRA, Amélia. Métodos quantitativos e qualitativos: um resgate teórico. **Revista Interdisciplinar Científica Aplicada**, Blumenau, v.2, n.4, p.01-13, Sem II. 2008.

GERHARDT, T. E.; SILVEIRA, D. T. **Métodos de pesquisa**. Rio Grande do Sul: UFRGS Editora, 2009

GIL, Antônio Carlos. **Como elaborar projetos de pesquisa**. 5. ed. São Paulo: Atlas, 2010.

GOMES, A.; HENRIQUES, J.; MENDES, A. J. Uma proposta para ajudar alunos com dificuldades na aprendizagem inicial de programação de computadores. **Educação, Formação & Tecnologias**, v. 1 n. 1, p. 93-103, 2008. Disponível em: <<https://eft.educom.pt/index.php/eft/article/view/23>>. Acesso em: 20 mar. 2018.

JOHNSON, S. **Surpreendente**: a televisão e o videogame nos tornam mais inteligentes. Rio de Janeiro: Elsevier. 2005.

JÚNIOR, J. C. R. P.; RAPKIEWICZ, Clevi Elena. O processo de ensino-aprendizagem de fundamentos de Programação: uma visão crítica da pesquisa no Brasil. In: **Anais do XII Workshop sobre Educação em Computação (SBC)**. 2004.

LAKATOS, Eva Maria; MARCONI, Marina de Andrade. **Metodologia científica**. São Paulo: Atlas, 2000. Metodologia do trabalho científico, v. 6, 2010.

MEDINA, Marco; FERTING, Cristina. **Algoritmos e programação**: teoria e prática. São Paulo: Novatec Editora, 2005.

MICROSOFT. **Introdução a linguagem C# e ao .NET framework**. 2015. Disponível em: <<https://docs.microsoft.com/pt-br/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the-net-framework>>. Acesso em: 10 maio 2018.

MOTA, Marcelle Pereira; PEREIRA, Lis W. Kanashiro; FAVERO, Eloi Luiz. Javatool: Uma ferramenta para o ensino de programação. In: **Congresso da Sociedade Brasileira de Computação. Belém**. XXVIII Congresso da Sociedade Brasileira de Computação. 2008. p. 127-136.

NETTO, J. C. M; MACHADO, L. S; MORAES, R. M. **Um estudo comparativo de ferramentas para a criação de jogos educacionais baseados em realidade virtual**. 2006. Disponível em: <https://www.researchgate.net/profile/Liliane_Machado/publication/228848910_Um_estudo_comparativo_de_ferramentas_para_a_criacao_de_jogos_educacionais_baseados_em_realidade_e_virtual/links/00463524ef597bc905000000/Um-estudo-comparativo-de-ferramentas-para-a-criacao-de-jogos-educacionais-baseados-em-realidade-virtual.pdf>. Acesso em: 14 abr. 2018.

OLIVEIRA, Manassés Vitorino; RODRIGUES, Luciene Cavalcanti; QUEIROGA, Ana. Material didático lúdico: uso da ferramenta Scratch para auxílio no aprendizado de lógica da

programação. In: XXII Workshop de Informática na Escola (WIE 2016), 2016, Uberlândia-MG. **Anais...** Uberlândia- MG: WIE, 2016, p. 359.

PEARS, A. *et al.* A survey of literature on the teaching of introductory programming. **SIGCSE Bulletin**, v. 39, n. 4, p. 204-223, 2007. Disponível em: <<http://dl.acm.org/citation.cfm?id=1345441>>. Acesso em: 16 mar. 2018.

PERUCIA, A. S., BERTHÊM, A.C., BERTSCHINGER, G.L., MENEZES, R.R.C. **Desenvolvimento de jogos eletrônicos - Teoria e Prática**. São Paulo: Novatec, 2005. p. 23.

PIMENTA, Selma Garrido; ANASTASIOU, Léa das Graças Camargos. **Docência no Ensino Superior**. 2. ed. São Paulo: Cortez, 2005.

RAPKIEWICZ, Cleli Elena *et al.* Estratégias pedagógicas no ensino de algoritmos e programação associadas ao uso de jogos educacionais. **RENOTE**, v. 4, n. 2, 2006.

REBOUL, Olivier. **O que é Aprender**. Coimbra, Portugal: Livraria Almedina, 1982.

RIBEIRO, Paula Ceccon; MARTINS, Carlos Bazílio; BERNARDINI, Flávia Cristina. A Robótica como Ferramenta de Apoio ao Ensino de Disciplinas de Programação em Cursos de Computação e Engenharia. In: XVII Workshop de Informática na Escola (WIE 2011), 2011, Aracaju- SE. **Anais...** Aracaju- SE: WIE, 2011, p. 1108-1117.

ROBINS, A. Learning edge momentum: A new account of outcomes in CS1. **Computer Science Education**, v. 20, n. 1, p. 37-71, 2010.

ROLLINGS, Andrew; MORRIS, Dave. **Game Architecture and Design: A New Edition**. New Riders Publishers, 2004.

SAVI, Rafael; VON WANGENHEIM, Christiane Gresse; ULBRICHT, Vania; VANZIN, Tarcisio. Proposta de um modelo de avaliação de jogos educacionais. **Renote**, v. 8, n. 3, 2010.

SILVA, I. K de O.; MORAIS II, Marçal José de Oliveira. Desenvolvimento de jogos educacionais no apoio do processo de ensino-aprendizagem no ensino fundamental. **HOLOS**, v. 5, 2011.

SOUZA, Bruno; JÚNIOR, José; FORMIGA, Andrei. **Introdução a Programação**. João Pessoa: Editora da UFPB, 2014.

TAROUCO, L. M. R., *et al.* Jogos educacionais. 2. ed. **RENOTE**. Novas Tecnologias na Educação, 2004.

UNITY. **Unity-Products**. 2018. Disponível em: <www.unity3d.com>. Acesso em: 10 maio 2018.

ZANETTI, Humberto Augusto Piovesana *et al.* Proposta de ensino de programação para crianças com Scratch e Pensamento Computacional. **Tecnologias, Sociedade e Conhecimento**, v. 4, n. 1, p. 43-58, 2017.

APÊNDICES

APÊNDICE A - Game Design Document

Move

Game Design Document

Versão: 2.1

Autores:

Davi Hagap Emanuel da Silva

Wandrey Matheus Moreira da Costa

São João Evangelista MG,
Julho de 2018

Índice

| | |
|----------------------------------|-----------|
| 1. HISTÓRIA..... | 54 |
| 2. GAMEPLAY..... | 56 |
| 3. PERSONAGENS..... | 57 |
| 4. CONTROLES..... | 57 |
| 5. CÂMERA..... | 57 |
| 6. UNIVERSO DO JOGO | 58 |
| 7. INTERFACE | 59 |

1. HISTÓRIA

Rock é um caçador de tesouros que está sempre em buscas de relíquias, em sua última aventura ele encontrou uns pergaminhos que relatam cinco relíquias sagradas que estão espalhados pelo mundo. Segundo a lenda, a junção destas relíquias podem conceder um desejo, destes coisas simples até mesmo o tornar em um deus.

1º PERGAMINHO

A nossa história começa quando Rock um caçador de tesouro encontra em uma caverna um óculos e alguns pergaminhos antigos com os dizeres:

“A muito tempo atrás deuses viviam entre os mortais, ajudando-os a construir impérios e alavancando sua tecnologia rudimentar. Passados alguns séculos eles perceberam que a humanidade já poderia viver sozinha, que não existiam mais nada a ensinar, então eles foram embora e deixaram escondidos pelo mundo suas relíquias, armas mágicas que se reunidas poderiam conceder qualquer desejo.”

Óculos Amaldiçoados

Esses óculos é um dos presentes dos deuses deixados há humanidade, ele permite ao seu usuário encontrar o caminho correto para as relíquias, entretanto se desviares do caminho a sua pena será a morte.

Bastão de Keebt

Keebt considerada a primeira Xamã existente, sendo a deusa da magia antiga e das florestas selvagens, deixou escondido em sua floresta um bastão capaz de manipular e dobrar a sua vontade todos os aspectos da natureza selvagem.

CONTINUAÇÃO

2º PERGAMINHO

A próxima relíquia será encontrada no templo do deus Asta, este por sua vez gostava de pregar algumas peças e algumas são mortais, seu templo é conhecido por ter algumas armadilhas. Neste templo ele guarda a Espada de Ansata, esta relíquia permite a manipulação mental.

3º PERGAMINHO

Ogum o deus da guerra e senhor do submundo acredita que somente os fortes conseguem sair de um de seus domínios, ele sabe recompensar os corajosos que não temem o perigo ou fogem de uma batalha. Ogum deixou aos mortais as Maças do Terror, duas armas que representam a sua própria brutalidade, força e terror nos corações de seus adversários.

4º PERGAMINHO

Ni'tras a deusa dos sonhos, da imaginação e da sabedoria escolheu as montanhas como lar, sendo segundo ela o lugar perfeito para meditação e busca constante do auto entendimento e do entendimento de tudo ao seu redor. Ni'tras deixou ao mundo carnal o Machado do Julgamento arma tão poderosa que permite cortar a realidade, entretanto usando uma vez se auto destrói junto com o usuário, associando assim ao poder das palavras que se lançadas podem criar ou destruir.

5º PERGAMINHO

Skadi a deusa do inverno era a mais fria entre os deuses, sendo esta a que menos se preocupava com os mortais, mesmo assim ela deixou para eles a Espada Ymir, arma capaz de criar poderosas rajadas de vento congelantes.

2. GAMEPLAY

Move será um jogo educacional voltado para o ensino de programação básica, a mecânica adotada para a jogabilidade é a de cliques na tela, em outras palavras o jogador irá escolher os comandos que deseja que o personagem realiza, tais como andar, virar para a esquerda ou direita, pular, também estarão disponíveis comandos de ações para ilustrar melhor a programação básica, como exemplo repetição, se, esperar tecla.

Os comandos que o jogador quiser que o personagem realize irão aparecer em um compilador, o jogador irá programar as ações do personagem e logo em seguida irá clicar em executar os comandos, dessa forma o personagem começará a executar as animações correspondentes aos comandos escolhidos pelo jogador.

Para passar de uma fase a outra o personagem deve chegar ao final do percurso, em algumas fases o personagem tem que pegar algumas relíquias que aparecerão pelo caminho, somente coletando os objetivos e chegando ao final do percurso e parando a execução no final que se passara de uma fase a outra.

Move contará com um sistema de estrelas baseados na quantidade de linhas de códigos que o jogador usou para chegar ao final da fase, terminando a fase irá aparecer a tela de estrela, quando menos linhas de códigos utilizadas para completar a fase melhor será as estrelas, caso o usuário gaste linhas de código em excesso, vai ter que repetir a fase para conseguir no mínimo uma estrela e avançar de fase.

3. PERSONAGENS

O personagem utilizado no jogo, é o Ethan que é disponibilizado pelo motor de jogos Unity 3D, o personagem se chamará Rock e será um caçador de recompensas.

4. CONTROLES

O personagem será controlado por comandos disponíveis na tela, ao selecionar os comandos desejados e clicar em executar a animação do personagem será iniciada.

Para isso o usuário usará o teclado e mouse para realizar as ações dentro do jogo

Mouse

| | |
|----------------|---------------------------------|
| Botão Esquerdo | Seleciona os comandos desejados |
| Botão Direito | Exclui comando no compilador |
| Scroll | Zoom |

Teclado

| | |
|------------|--|
| W, A, S, D | Mover a câmera |
| C | Mudar a câmera |
| V | Mudar a velocidade de execução da animação |

5. CÂMERA

As câmeras utilizadas no jogo serão de rolagem, na qual o usuário poderá mover e dar zoom para visualizar elementos que em primeiro momento não aparecem na cena, as câmeras estarão posicionadas no alto apontando para baixa, sendo assim possível enxergar grande parte do cenário e do percurso no qual o personagem deve percorrer.

6. UNIVERSO DO JOGO

O jogo será composto por vários blocos de cenários, cada bloco será de cenário único e temas, no total serão cinco tipos de cenários, assim o jogo não fica cansativa e monótono para os jogadores

Bloco 1 - Floresta

Este cenário contará com elementos de florestas, tais como árvores, árvores mortas, pedras, cogumelos, troncos, deverá possuir som de animais, ventos e tudo que remete as florestas.

Bloco 2 - Templo

Este cenário contará com elementos presentes em tempos, tais como pedras, estruturas antigas, colunas, destroços, armadilhas, o som deverá remeter mistério e sons de água presente nos cenários.

Bloco 3 - Lava

Este cenário contará com elementos presentes em vulcões, tais como pedras e lava, emitindo sons das lavas borbulhantes e mistério.

Bloco4 - Montanhas

Este cenário contará com elementos presentes montanhas tais como pedras, montanhas, pontes, abismos, um cenário que ocorre em uma região bastante alta, devido a isso devera possuir sons misteriosos e de grande quantidade de ventos.

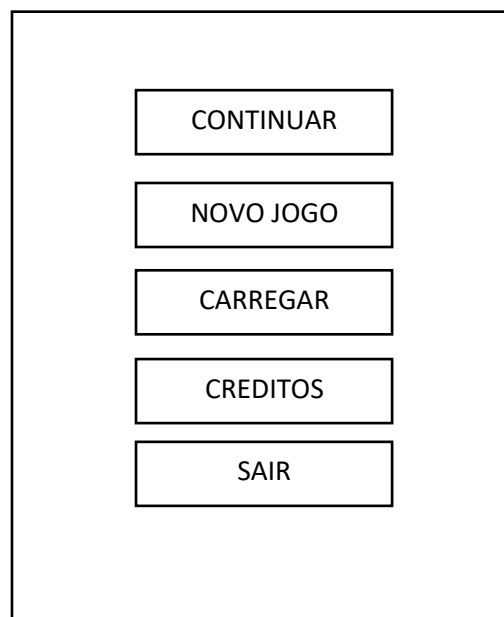
Bloco5 - Gelo

Este cenário contará com elementos presentes locais gelados, como pedras de gelos flutuantes em um mar gelado, montanhas de gelo, pontes pra atravessar, neve e sons que remetem tempestades de neve.

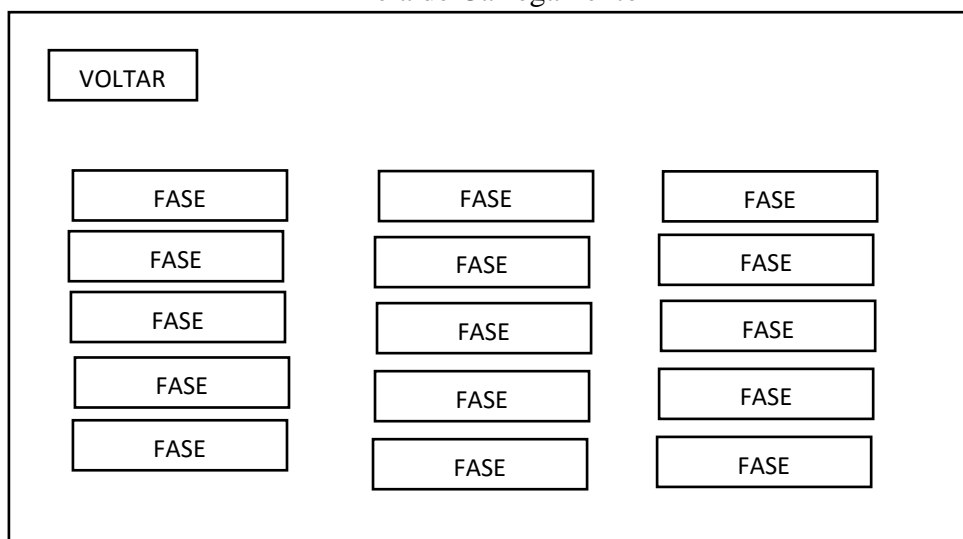
7. INTERFACE

A interface para a jogo deverá ser simples e ao mesmo tempo atrativa e de fácil uso para os jogadores, deverá contar com imagens e botões de fácil cesso e entendimento.

Tela de menu

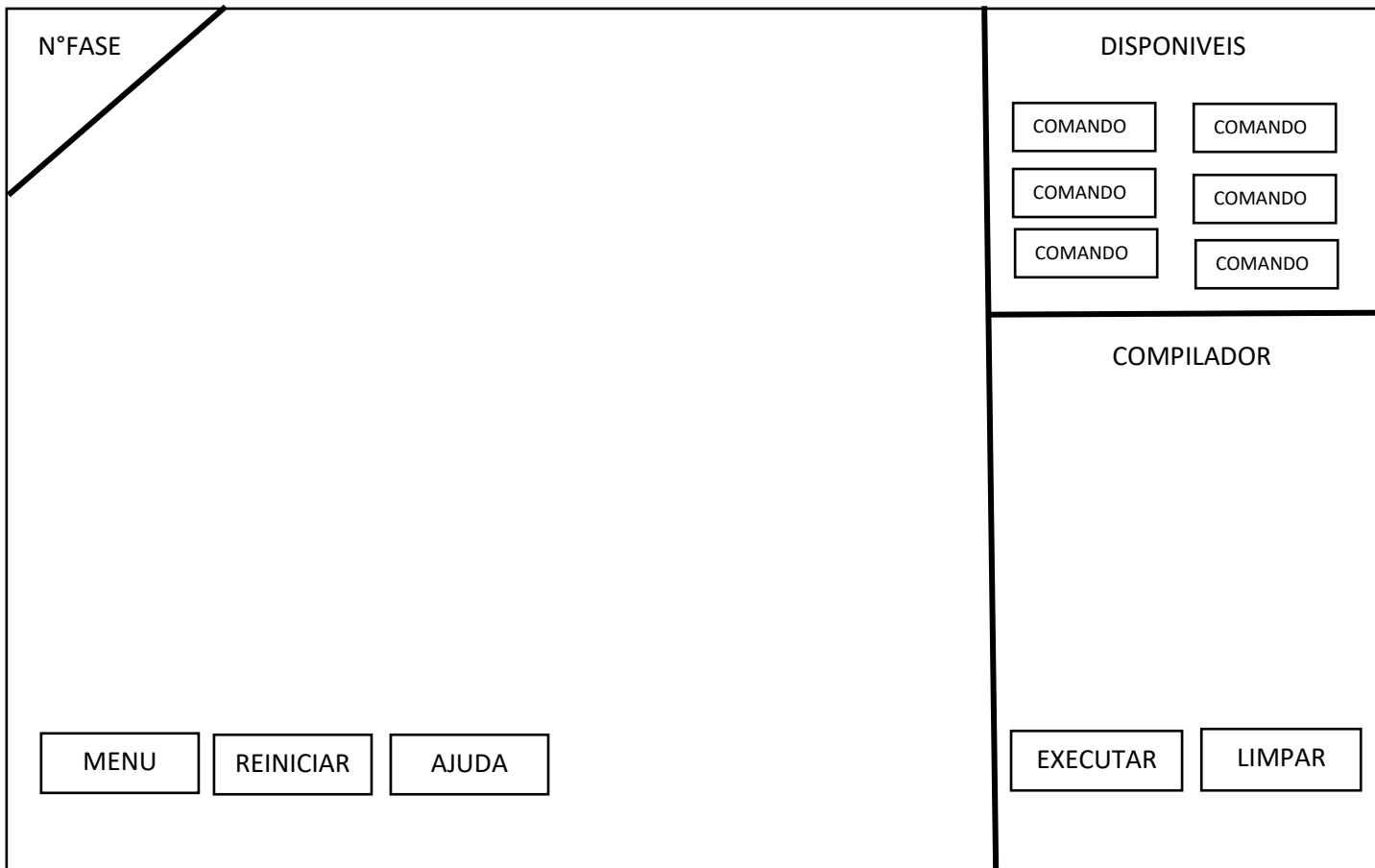


Tela de Carregamento



CONTINUAÇÃO

Tela do Jogo



Tela de Ajuda

