



FACULDADE DE TECNOLOGIA DE AMERICANA
Curso Superior de Tecnologia em Jogos Digitais

Antônio Kleber Cremonese
Bruno Bueno Nicoleti
Gabriel Yeiso Tamashiro

CODE NIGHTMARE

Americana, SP

2017

FACULDADE DE TECNOLOGIA DE AMERICANA
Curso Superior de Tecnologia em Jogos Digitais

Antônio Kleber Cremonese

Bruno Bueno Nicoleti

Gabriel Yeiso Tamashiro

CODE NIGHTMARE

Relatório técnico desenvolvido em cumprimento à exigência curricular do Curso Superior de Tecnologia em Jogos Digitais sob a orientação do Prof. Dr. Kleber de Oliveira Andrade.

Americana, SP

2017

FICHA CATALOGRÁFICA – Biblioteca Fatec Americana - CEETEPS
Dados Internacionais de Catalogação-na-fonte

C937c CREMONESE, Antonio Kleber

Code Nightmare / Antonio Kleber Cremonese ; Bruno Bueno Nicoletti ; Gabriel Yeiso Santos Tamashiro. – Americana, 2017.

28f.

Monografia (Curso de Tecnologia em Jogos Digitais) - - Faculdade de Tecnologia de Americana – Centro Estadual de Educação Tecnológica Paula Souza

Orientador: Prof. Dr. Kleber de Oliveira Andrade

1 Jogos eletrônicos I. NICOLETI, Bruno Bueno II. TAMASHIRO, Gabriel Yeiso Santos III. ANDRADE, Kleber de Oliveira IV. Centro Estadual de Educação Tecnológica Paula Souza – Faculdade de Tecnologia de Americana

CDU: 681.6

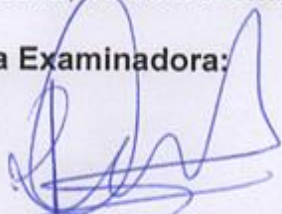
Antônio Kleber Cremonese
Bruno Bueno Nicoleti
Gabriel Yeiso Tamashiro

CODE NIGHTMARE

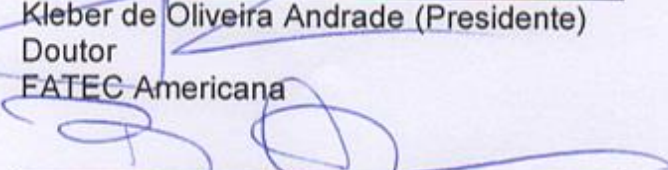
Relatório técnico apresentado como exigência parcial para obtenção do título de Tecnólogo em Jogos Digitais pelo CEETEPS/Faculdade de Tecnologia – FATEC/ Americana.

Americana, 15 de Dezembro de 2017.

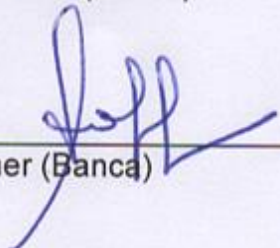
Banca Examinadora:



Kleber de Oliveira Andrade (Presidente)
Doutor
FATEC Americana



Benedito Aparecido Cruz (Banca)
Mestre
FATEC Americana



Renato Kraide Soffner (Banca)
Doutor
FATEC Americana

RESUMO

Este jogo foi planejado inicialmente com o ideal de servir como um complemento e/ou como um primeiro contato para o aprendizado de programação, para jovens e crianças. A proposta se iniciou após uma breve pesquisa e a verificação da escassez de jogos e materiais nesse modelo. O projeto foi idealizado como um jogo para dispositivos com o sistema operacional *android*, como *Smatphones* e *tablets*, utilizando uma mecânica para com a qual o jogador possa desenvolver os códigos solicitados é necessário que o mesmo arranje blocos, cada qual com uma instrução, em determinada ordem para que assim estes formem uma instrução de código. O gênero do jogo é uma mescla de *Escape the room* (estilo de jogo onde o jogador se encontra preso em algum ambiente e precisa interagir com elementos para escapar deste local) e *puzzle* (onde enigmas são propostos para serem resolvidos e assim avançar no jogo). Tanto o cenário quanto o personagem são modelos em 3D, no estilo *Low Poly*, que é o estilo artístico definido pela pouca quantidade de polígonos em sua composição, e a visão que se tem é uma visão isométrica, onde a câmera é fixada numa angulação de 45°. A *engine* utilizada para o desenvolvimento do jogo foi a *Unity*, por ser uma plataforma completa, que possui diversos recursos que agilizam o processo de criação e dentro desta, todos os códigos utilizados foram programados na linguagem *C#*. O jogo foi disponibilizado na *Play Store* com o nome de *Code Nightmare* e pode ser baixado gratuitamente.

Palavras Chave: jogo; código; desenvolvimento; *Unity*.

ABSTRACT

This game was firstly planned to serve as a complement and/or a first contact for the code development learning, for kids and teens. The idea started after a brief research and the confirmation of how little is the amount of materials on this model. The project was idealized as a game for the Android system gear, such as Tablets and smartphones, using a mechanic which for the player to develop the asked codes, he must use blocs, each one with a code instruction, putting them in order to form a real code line. The game genre is a mix of Escape the room (A type of game where the player is locked in a room and must use and find items to escape) and Puzzle games. The whole scenario and the player character are 3D models with the Low Poly style, which is an art style defined by the use of few polygons in its composition; for the game vision was used an isometric view, that is achieved by using a camera in a fixed 45° angle. The game engine chosen to be used in this project was the Unity, for being such a complete platform, who has a lot of resources that make the whole process of creation and development a lot faster. Inside the Unity, all the code scripts used were written in the C# language. The game is available on the Play Store by the name of Code Nightmare and it's free.

Keywords: *game; code; development; Unity.*

SUMÁRIO

1 INTRODUÇÃO.....	07
2 METODOLOGIA.....	09
2.1 Brainstorming.....	09
2.2 Programação.....	09
2.3 Arte.....	10
2.4.1 Arte Icone e menu.....	11
2.4 Som.....	12
2.5 Idioma.....	13
2.6 Estratégia e Level Design.....	13
3 IMPLEMENTAÇÃO.....	14
3.1 Telas e menu.....	14
3.2 Funcionamento da fase.....	15
3.3 Cenário.....	15
3.3 Personagem.....	17
4 RESULTADOS.....	19
5 CONSIDERAÇÕES FINAIS.....	26
CRONOGRAMA.....	27
REFERÊNCIAS.....	28

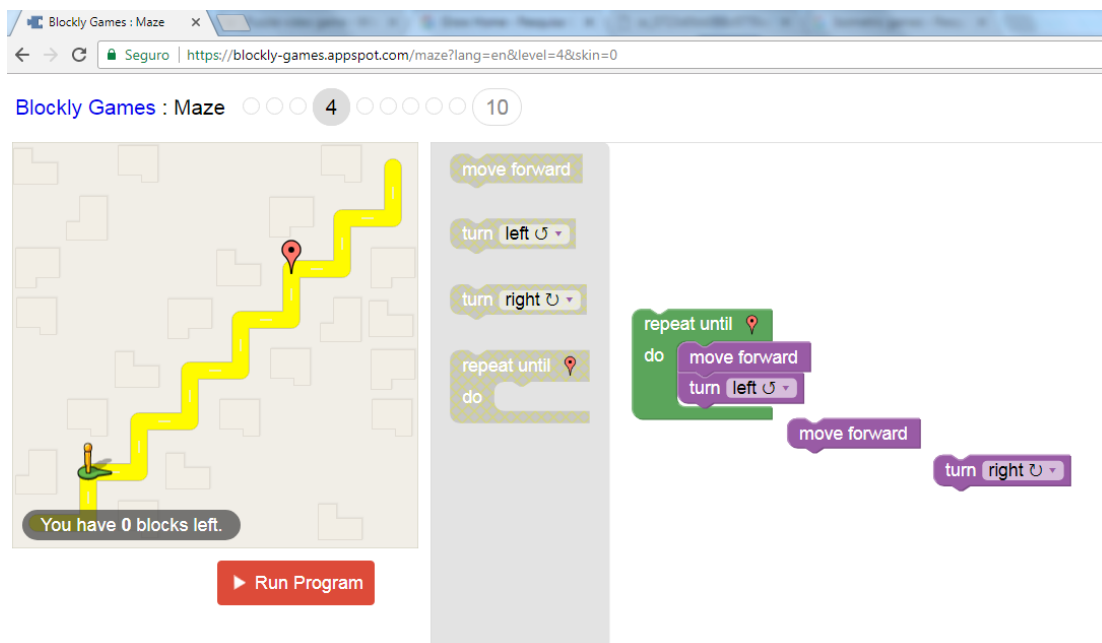
1 INTRODUÇÃO

A princípio, a ideia base foi a criação de um jogo que possuísse elementos que auxiliassem no aprendizado de conceitos básicos de programação.

A partir desta ideia foi então definido o estilo e gênero do jogo, juntamente com uma breve história. O gênero do jogo é uma mistura de dois gêneros, o *Escape The Room*, sendo este um estilo de jogo definido pela mecânica e história simples, onde o jogador se encontra preso em um ambiente e deve interagir com itens e elementos do cenário para resolver problemas e enigmas, escapando assim do local onde está; o outro gênero é o *puzzle*, que pode ser caracterizado pela ênfase no uso de raciocínio e lógica para a solução dos problemas e desafios propostos.

Para a implementação dos elementos de *puzzle* em conjunto com os conceitos de programação a serem passados foi utilizado como inspiração a mecânica do conjunto de jogos do Google chamados de Google Blockly Games, que usam a ideia de blocos para a criação de instruções de códigos a serem executados, como demonstrado na Figura 1.

Figura 1: Imagem do jogo Blockly Games



Fonte: Google – Blocly Games

No jogo desenvolvido, cada um dos blocos possui uma instrução anexada e com a organização dos mesmos em determinada ordem, o jogador destrava portas, podendo assim avançar no jogo.

O estilo artístico utilizado no jogo, foi o *Low Poly* com todos os elementos em 3D. O estilo *Low Poly* pode ser definido como um estilo onde a textura dos objetos é construída com uma baixa quantidade de polígonos, fazendo deste um estilo artístico simples e de fácil reconhecimento.

Buscamos neste jogo atingir um público com o interesse no aprendizado ou curiosidade referente a programação, o que de acordo com pesquisas é um mercado que tem crescido rapidamente nos últimos anos, principalmente entre os jovens e crianças.

2 METODOLOGIA

Neste capítulo discorreremos sobre os métodos que foram utilizados na construção e desenvolvimento do jogo. Explicando os processos de *Brainstorming* (que se refere ao conceito do debate e discussão de ideias), passando pelas partes de arte, interface, *level design* (definido como a parte dentro do desenvolvimento de um jogo onde são criados os níveis e desafios), som e programação.

2.1 BRAINSTORMING

Como dito no capítulo anterior a ideia principal foi a criação de um jogo que possuísse elementos que auxiliassem no aprendizado de conceitos básicos de programação. Tendo isto em mente, iniciamos então a criação de uma história para servir como base, criando assim um início e um final para o jogo. Chegamos então na proposta de um jogo com a principal característica de um *Escape the room*, onde o jogador deve interagir com elementos chave do ambiente para assim escapar do prédio que nos serve de cenário, dentro dessas interações inserimos os elementos de *puzzle* que por sua vez contem a ideia principal do jogo, que consiste nos elementos de programação. Foi nesta etapa também onde definimos o cenário onde o jogo se passa, que é a representação do interior de uma empresa no ramo de TI, o mesmo é composto por um prédio com quatro andares, onde cada andar conta com diferentes salas e diferentes layouts, exceto pelo térreo que possui quatro salas, todos os andares possuem cinco salas onde é possível a exploração.

2.2 PROGRAMAÇÃO

O jogo todo foi desenvolvido dentro da Unity, iniciamos o projeto na versão da Unity 5.6.2 e posteriormente fizemos o *upgrade* para a versão 2017.1.0. O motivo de escolha da *Unity* em relação as outras *engines* se deve ao fato de estarmos familiarizados com o ambiente da mesma, tendo desenvolvido alguns projetos dentro do ambiente acadêmico. Uma das ferramentas disponibilizadas pela *Unity* que facilita todo o processo de desenvolvimento e foi utilizada constantemente pelo grupo foi o *Unity Colab*, que permite com que várias pessoas trabalhem no mesmo projeto e possam enviar as mudanças que realizaram, uns para os outros, possibilitando assim

com que todos trabalhem na mesma versão do projeto, evitando conflitos e agilizando muito o processo.

Os dois membros do grupo, Antônio Kleber e o Bruno Bueno, possuem bastante experiência com o trabalho de programação e desenvolvimento, portanto foram os membros responsáveis por esta sessão do projeto e por opção dos mesmos todo o código de programação foi escrito na linguagem orientada a objeto C#.

2.3 ARTE

No processo de *brainstorming* ficou definido que o estilo de arte a ser utilizado é o 3D com o elemento Low Poly, que está presente em diversos jogos atuais, principalmente nos meios *indies* e *mobiles*, por serem elementos que exigem uma quantidade menor de recursos como processamento e memória em comparação a elementos com uma textura muito mais definida. Um exemplo de Low Poly é o jogo Grow Home (Figura 2) lançado em 2015 pela Ubisoft, que também foi feito na Unity e possui todo seu cenário e personagens com o efeito *Low Poly*.

Figura 2: Imagem do jog *Grow Home* (2015)



Fonte: Ubisoft / Ubisoft Reflections – *Grow Home*

Outra característica artística é a utilização de uma câmera isométrica, que pode ser definido pela fixação da câmera em um ângulo de 45°. Com este tipo de posicionamento da câmera, é possível uma visão mais ampla do cenário e uma

navegação mais ágil. Esta característica é utilizada a muito tempo em diversos jogos em 2D a muito tempo, e recentemente em jogos com objetos em 3D também, exemplos de jogos de grande sucesso que utilizaram essa técnica são jogos das series Pokémon e Age of Empires (Figura 3).

Figura 3: Imagem do jogo Age of Empires II: The Age of Kings – Visão Isométrica.



Fonte: Microsoft / Age of Empires II

2.3.1 ARTE ICONES E MENU

Nas telas inicial, de opções e de créditos, foi utilizada uma técnica muito comum recentemente que é a disposição dos botões e textos em um fundo que exiba parte do cenário real do jogo. Como referencia foi utilizado o jogo *Cafe Racer* (Figura 4), jogo para *android* que também conta com o visual *Low Poly*. Todas estas telas foram feitas completamente dentro da Unity, com ressalva a alguns textos que foram criados com o *software Photoshop*. O membro do grupo Gabriel Yeiso foi o responsável pela criação destas cenas.

Seguindo o estilo padrão dos jogos *Escape The Room*, a *HUD (Heads up Display)* Expressão em inglês referente às informações mostradas ao jogador na tela de jogo, foi realizada de forma minimalista, exibindo apenas um mini mapa no canto superior direito, um pequeno botão para interação no canto inferior direito e o controle de movimentação do personagem que aparece apenas quando acionado no canto esquerdo da tela.

Figura 4 – Menu principal do jogo *Cafe Racer*.



Fonte: PiguinSoft

2.4 SOM

Para a melhoria da experiência durante o jogo e uma maior imersão, foi adicionado ao jogo uma trilha sonora e diversos efeitos sonoros. Toda a trilha foi adquirida de forma gratuita, com seus direitos autorais livres, permitindo o uso e edição dos mesmos. Esta foi a solução encontrada para a falta de equipamentos e experiência com a produção de trilhas sonoras.

Tendo em vista os jogos no estilo *Escape The Room*, que possuem trilhas sonoras mínimas, e a ideia de introduzir uma atmosfera de isolamento e tensão, a música de fundo durante o período de jogo possui um tom mais sombrio e uma semelhança com trilhas sonoras de ficção científica. Os botões do menu e as interações com computadores possuem sons específicos para cada uma.

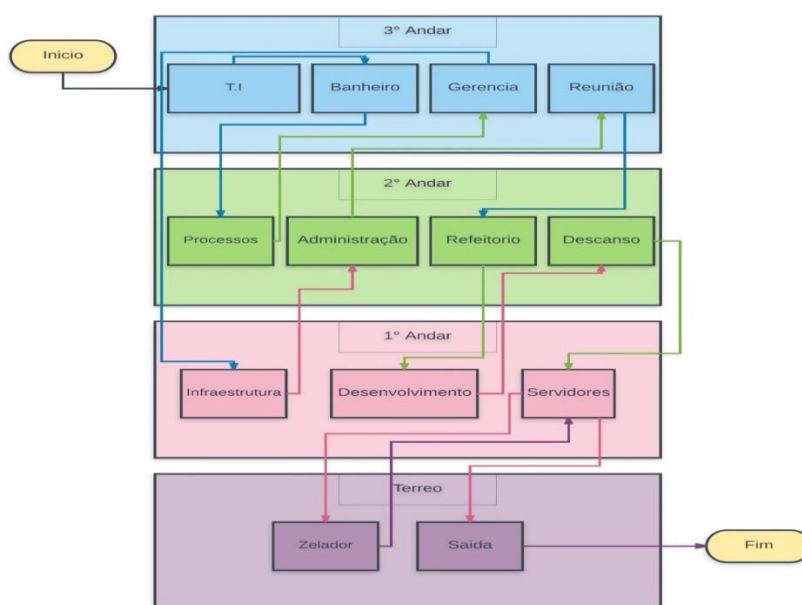
2.5 IDIOMA

Tendo em vista o público nacional todo o jogo foi desenvolvido voltado para o idioma local, o português, uma vez em que a leitura e a absorção do conteúdo se tornam mais rápidos e eficazes se comparados com a leitura de textos escritos em um idioma estrangeiro.

2.6 ESTRATEGIA E LEVEL DESIGN

O objetivo principal do jogo é conseguir sair do prédio, e para que isso aconteça o jogador deve destravar a porta da saída principal na sala onde se encontram os servidores, no 1º andar. Pensando nisso tudo foi criada uma sequência não linear para o destravamento das portas, conforme demonstrada na figura 5 o fluxo entre as salas e andares, elevando assim a dificuldade do jogo e fazendo com que o jogador percorra todo o terreno disponível de uma forma mais divertida, estimulando a curiosidade do jogador.

Figura 5 – Fluxo de relação entre salas e andares.



Fonte: Autoria própria

Para que o jogador possa abrir as portas que se encontram travadas é necessário resolver um desafio proposto de desenvolver alguma linha de código em uma linguagem baseada em C. Esta linguagem foi escolhida por ser a primeira língua que muitos dos estudantes entram em contato.

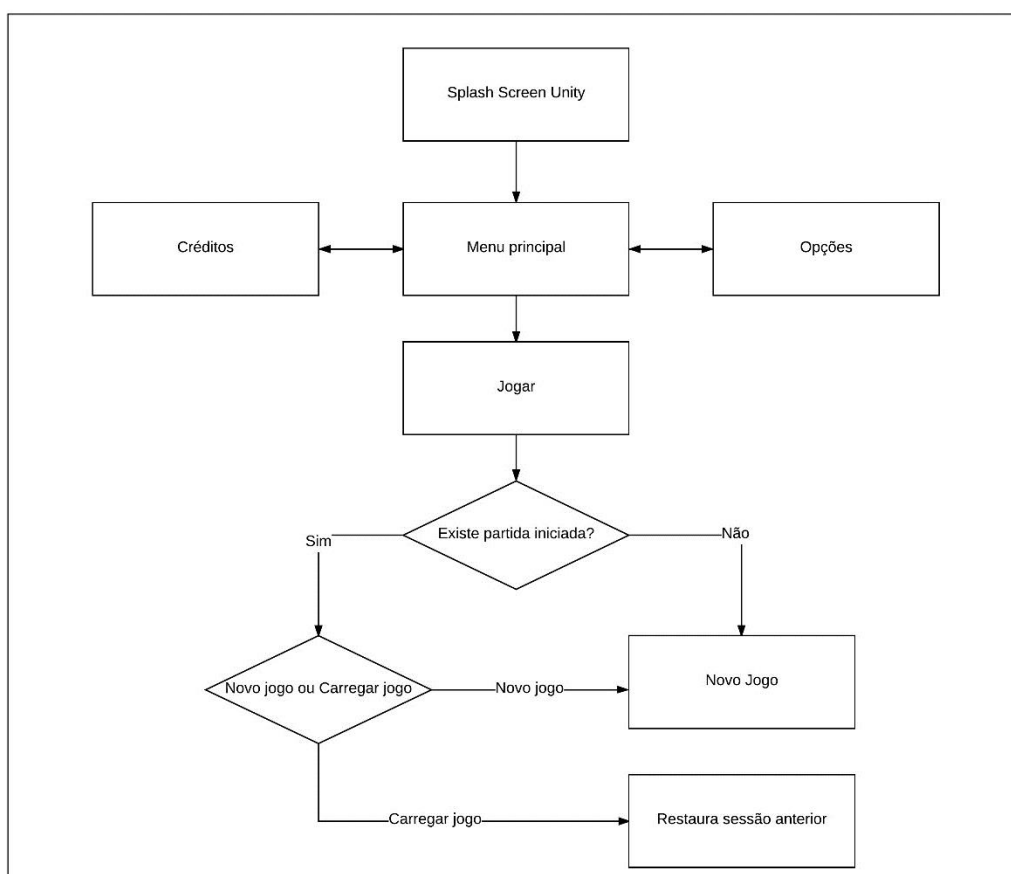
3 IMPLEMENTAÇÃO

Nos próximos sub-tópicos será descrito o funcionamento do jogo, abrangendo suas telas, menus, logica responsável pela verificação da solução dos desafios.

3.1 TELAS E MENU

Após o carregamento do jogo, a tela inicial é apresentada ao jogador, nela constam 3 opções, são elas: Jogar, Opções e Créditos, cada uma com seu respectivo botão. Ao selecionar a opção jogar, caso o jogador já tenha iniciado uma partida, é perguntado se ele pretende retomar de onde parou ou iniciar um novo jogo. Um diagrama para representar e entender esta tela pode ser verificado na figura 6.

Figura 6 – Diagrama menu principal.



Fonte: Autoria própria

Na tela de Opções, o jogador pode escolher desabilitar ou habilitar o som, alterar a velocidade de movimento do personagem e também o estilo de controle, que

pode ser escolhido entre *Click and Point*, estilo onde o jogador clica em um ponto no chão definindo o destino do personagem, ou *Joystick*, que é o modelo padrão de controle para a maioria dos jogos, onde o jogador movimenta o personagem através de um controlador analógico exibido na tela.

3.2 FUNCIONAMENTO DA FASE

Como citado anteriormente, o objetivo principal do jogador é encontrar uma forma de sair do prédio e isso só pode ser alcançado destrancando todas as portas e finalmente abrindo a porta principal. Então para implementação desta mecânica, cada porta foi ligada virtualmente a um dos computadores onde é possível a interação do jogador, fazendo assim a verificação de cada uma destas portas, pois assim que o desafio proposto em respectivo computador é resolvido, a porta vinculada a este é aberta. Os desafios seguem um aumento crescente em sua complexidade, devido ao aumento do número de blocos ou ao uso de diferentes comandos na mesma sentença de código.

3.3 CENARIO.

Seguindo a ideia principal de fazer o jogo todo se passar no interior de uma empresa, após serem definidos a quantidade de andares e as salas que cada andar comportaria, distribuimos a criação dos mesmos entre os membros, como inspiração e modelo de referencia utilizamos um cenário do jogo *Persona 2: Eternal Punishment* (Figura 7), da produtora japonesa Atlus. O cenário é composto por diversos itens de escritório, um elevador e também possui a visão isométrica que foi utilizada em nosso projeto. Para a redução de tempo e conseguirmos uma melhor qualidade, optamos por não modelarmos os objetos do cenário por conta própria, devido a inexperiência do grupo com a modelagem em 3D e aplicação de texturas em alta qualidade de definição, assim sendo, foi realizada a compra de um *Asset* com os itens necessários para a construção do nosso cenário como um todo.

Figura 7 – Imagem do jogo *Persona 2: Eternal Punishment*. (2000)



Fonte: Atlus/Persona 2: Eternal Punishment

Assets são um conjunto de itens, como modelos em 3D, texturas, *scripts* e afins, que são vendidos ou disponibilizados de forma gratuita e podem ser incorporados no seu projeto.

Ao fim, o cenário do jogo é composto pelos seguintes andares e salas:

- Térreo: Recepção, banheiro masculino, banheiro feminino e sala do zelador.
- 1º Andar: Sala de infraestrutura, sala de desenvolvimento, sala dos servidores e banheiro.
- 2º Andar: Setor de processos, administração, refeitório, sala de descanso e banheiro.
- 3º Andar: Setor de T.I., sala de reunião, sala da gerencia e banheiro.

Todas as salas são exploráveis, o acesso à maioria delas é realizado através de uma porta de entrada que se encontra trancada e como foi detalhado anteriormente, esta vinculada a um dos computadores e é liberada após a solução com sucesso de seu respectivo desafio. Para a identificação dos elementos

interativos, utilizamos um efeito que adiciona uma coloração diferenciada sobre os mesmos, facilitando assim a jogabilidade.

3.4 PERSONAGEM.

Como detalhado anteriormente o jogo se passa no período de pesadelo do personagem, portanto definimos que o personagem não se encontraria em seu estado normal. Ele então é representado por um zumbi (Figura 8), que veste terno e gravata e possui uma maleta. Uma alusão ao trabalho excessivo exercido por este.

Figura 8 – Personagem Zumbi



Fonte: Autoria própria.

Ao fim do jogo o personagem é exibido em sua forma normal, em uma das cenas. Conforme a Figura 9.

Ambos os personagens foram adquiridos na *Unity Asset Store*, devido a falta de experiência do grupo com modelagem 3D. Algumas das animações foram realizadas com utilização do sistema online Mixamo.

Figura 9 – Personagem em sua forma normal.



Fonte: Autoria própria.

4 RESULTADOS

De acordo com as informações apresentadas nos capítulos anteriores, neste capítulo serão exibidos os resultados atingidos com o desenvolvimento do jogo.

A tela inicial do jogo é o menu principal (Figura 10), neste encontramos o título do jogo na parte superior o cenário de uma das salas ao fundo e três botões, que são: Jogar, Créditos e Opções.

Figura 10 – Tela Inicial

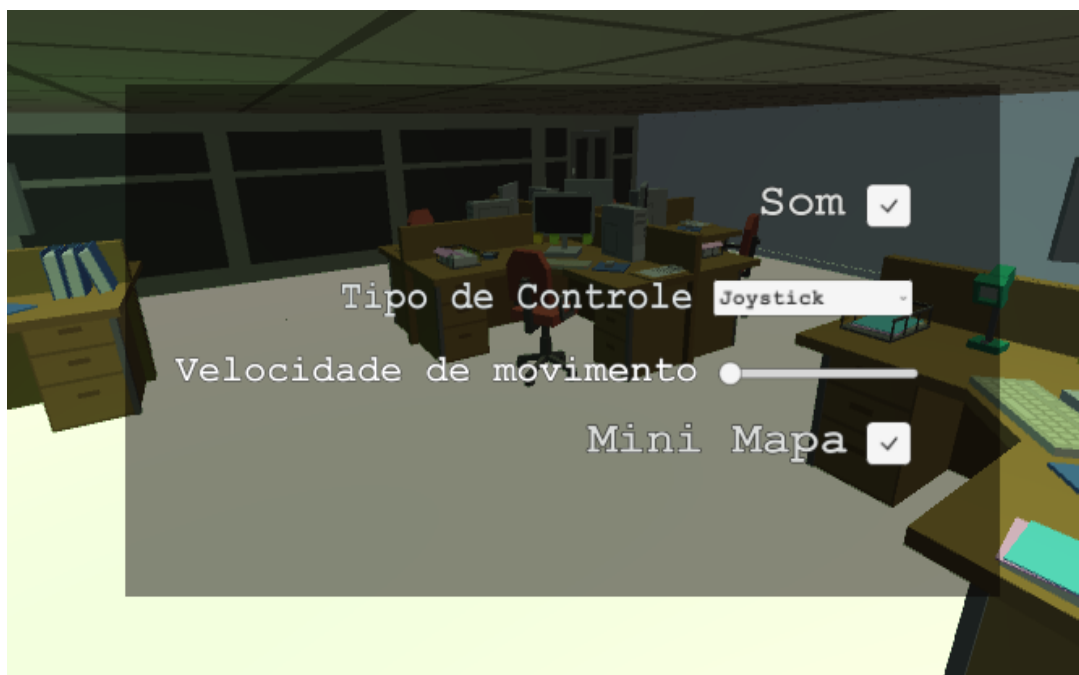


Fonte: Autoria própria.

Ao selecionar Opções é exibido ao jogador a tela respectiva (Figura 11), onde é possível realizar as configurações de jogabilidade, podendo habilitar ou desabilitar o som e o mini mapa, alterar a velocidade do personagem e o estilo do controle. Ao fundo é exibido o cenário de uma das salas do jogo.

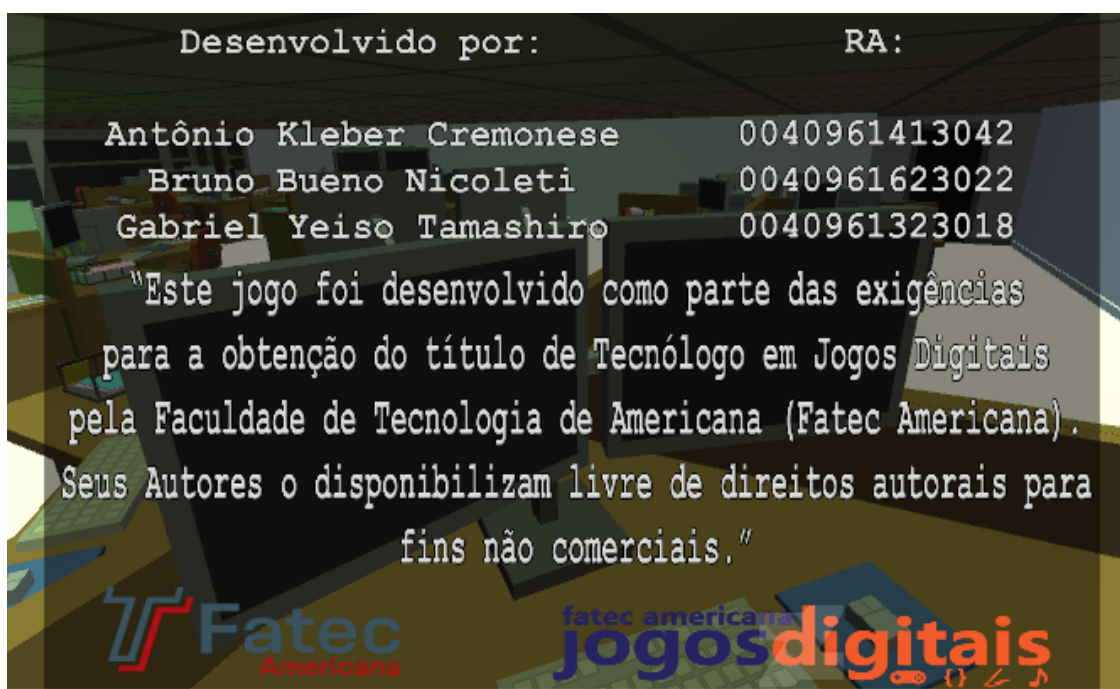
Voltando à tela inicial, e escolhendo a opção Créditos (Figura 12), é apresentado ao jogador informações sobre os desenvolvedores do jogo, juntamente com logo da Fatec Americana e o logo do curso de Jogos Digitais.

Figura 11 – Tela Opções



Fonte: Autoria própria.

Figura 12 – Tela Créditos



Fonte: Autoria própria.

Retornando ao menu inicial e selecionando a opção Jogar, caso seja a primeira vez que o jogo é iniciado, a cena para o início do jogo é exibida e o mesmo começa. Caso o jogador já tenha iniciado anteriormente alguma partida e seleciona a opção

Jogar, é exibido então ao jogador as seguintes opções: Continuar ou Novo jogo. Conforme demonstrado na Figura 13.

Figura 13 – Tela Inicial.



Fonte: Autoria própria.

Iniciando o jogo então, é apresentado ao jogador a seguinte tela (Figura 14), onde é possível observar o personagem caminhando pelo cenário, o mini mapa a direita e o botão de ação logo abaixo.

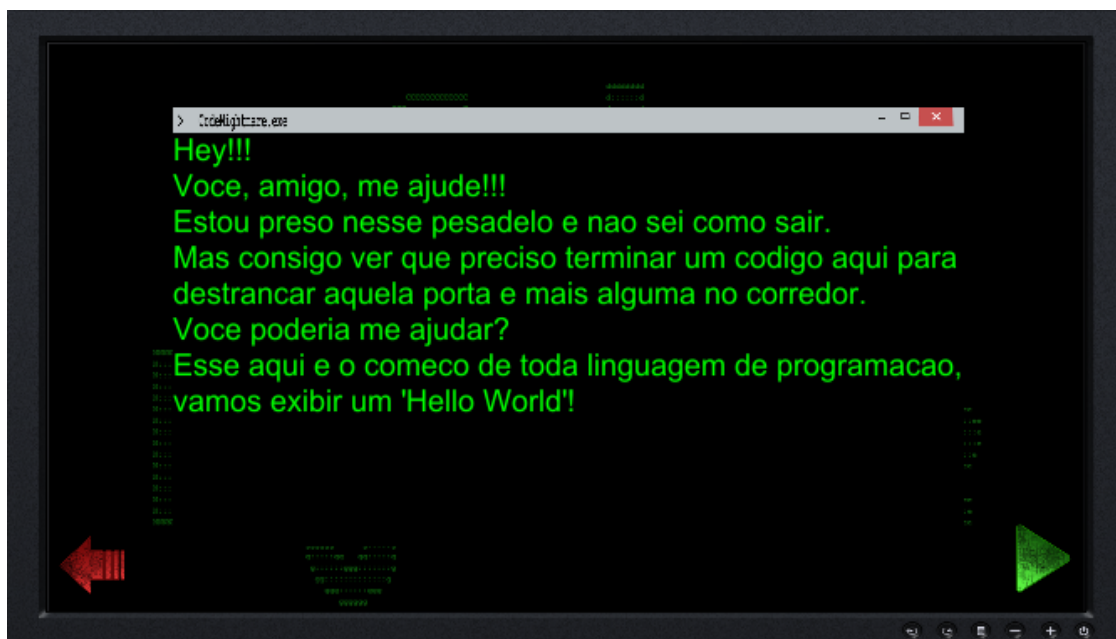
Figura 14 – Tela de jogo.



Fonte: Autoria própria.

Ao realizar a interação com os computadores indicados, é exibida então a tela (Figura 15) onde é explicado ao jogador qual será o desafio que este terá de realizar ao programar com os blocos. Nela é exibida um monitor com um texto dentro de um janela, parecida com um *prompt* de comando e dois botões, um para sair e outro para avançar.

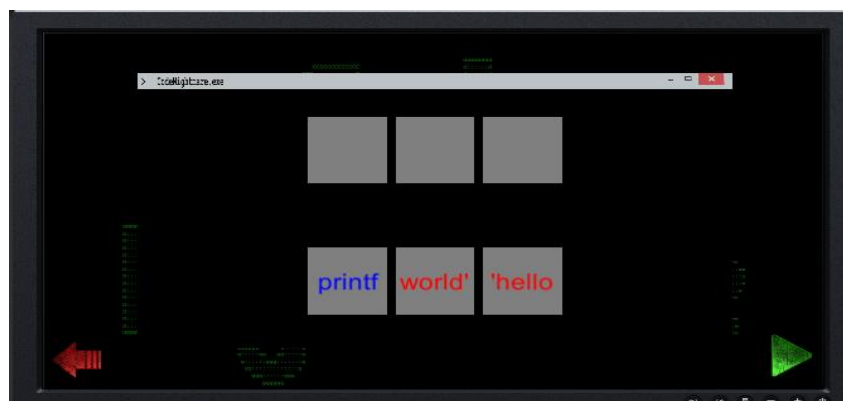
Figura 15 – Tela de interação.



Fonte: Autoria própria.

Dando sequência a tela anterior, após o texto é mostrado então a tela (Figura 16) com os blocos vazios na parte superior e os blocos que possuem as instruções de código na parte inferior. Para solucionar o desafio, o jogador deve arrastar os blocos da parte inferior para a parte superior na ordem correta.

Figura 16 – Tela de programação



Fonte: Autoria própria.

Caso o jogador consiga resolver o desafio proposto é demonstrado então a tela de *feedback* positivo (Figura 17), afirmando que este obteve sucesso. Caso isto não ocorra, o jogador é informado que a ação ainda não foi concluída conforme a figura 18.

Figura 17 – Tela em caso de acerto.



Fonte: Autoria própria.

Figura 18 – Tela em caso de erro.



Fonte: Autoria própria.

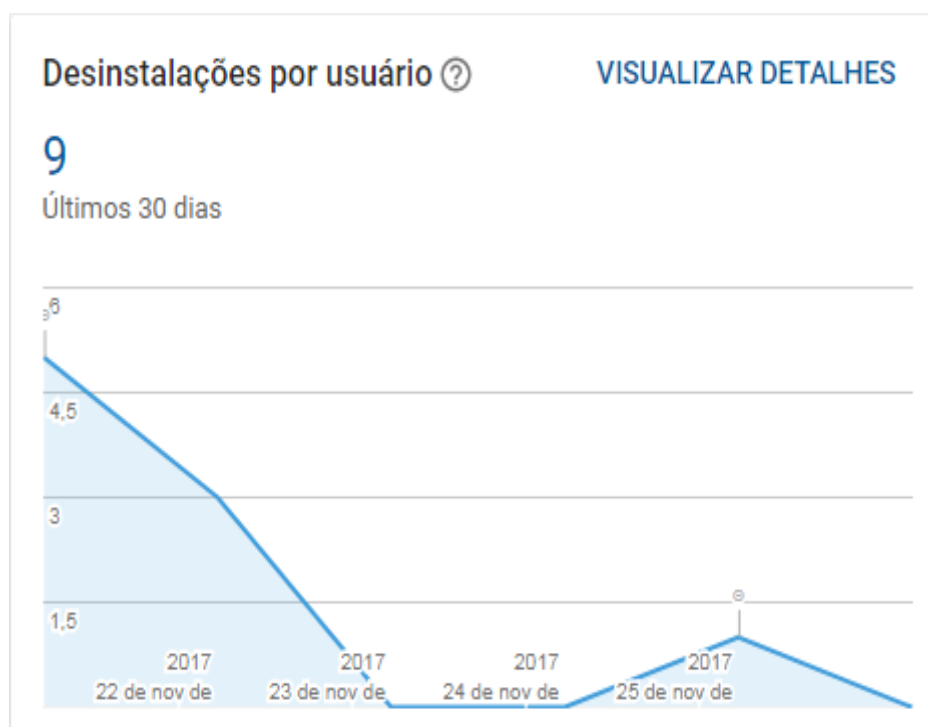
Após o lançamento do das versões beta do jogo na *Google Play Store*, que ocorreu no dia 21/11, os resultados atingidos foram os seguintes, o numero total de instalações, 24, são verificados na Figura 19, e de desinstalações, 9, na Figura 20, assim como o total de instalações por país.

Figura 19 – Total de instalações

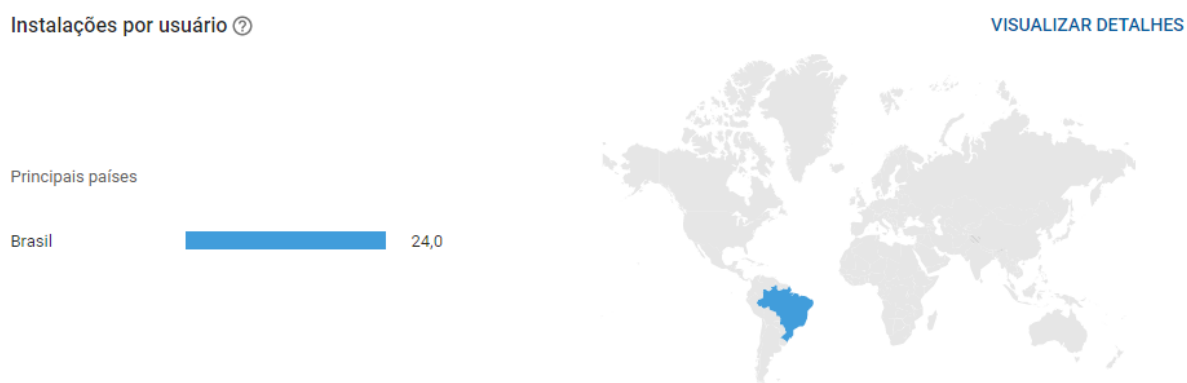


Fonte: Autoria própria.

Figura 20 – Total de desinstalações



Fonte: Autoria própria.

Figura 21 – Total de instalação por país.

Fonte: Autoria própria.

5 CONSIDERAÇÕES FINAIS

Após todo o semestre de desenvolvimento, o grupo se encontra com um jogo de alta complexidade e pode-se dizer que este atingiu um resultado satisfatório. Tendo em vista a grandeza que o projeto atingiu e as limitações técnicas do grupo, o jogo entrega todos os itens previamente idealizados pela equipe, apresentando um jogo que exhibe uma premissa simples, com um desenrolar desafiador e um gráfico minimalista porém de muita qualidade.

A realização deste projeto nos serviu como uma forma de aprendizado enorme, pois até então os membros do grupo não haviam desenvolvido um projeto tão extenso quanto este. Com todos os setores de desenvolvimento do jogo divididos entre os membros do grupo, assim como a arte, som e programação, todos obtiveram um ganho de experiência enorme, mas não só nos campos em que mais atuaram e sim como um todo, graças ao grande sentimento de unidade deste grupo. Todo o processo envolveu diversas dificuldades que foram superadas devido a união e comprometimento do grupo, onde todos estavam dispostos a realizar o projeto, sempre dispostos a ajudar uns aos outros, discutindo ideias e soluções.

Apesar do jogo se encontrar totalmente funcional e com uma bela apresentação, cabe ainda diversos polimentos, como uma inclusão maior de efeitos sonoros, alterações na *HUD*, e nos desafios. O maior motivo destas funções ainda não estarem disponíveis no jogo é devido ao tempo e a falta de experiência do grupo com diversos fatores.

Para versões futuras o grupo pretende acrescentar uma gama maior de efeitos sonoros, ajustar diversas questões dos cenários, como os verificadores de colisão, identificação de itens e locais, adicionar uma quantidade maior de itens de interação, textos explicativos e desafios, melhorando assim de modo geral a experiência com o jogo.

CRONOGRAMA

Atividades	Meses Correspondentes 2017				
	7	8	9	10	11
1. Levantamento bibliográfico sobre a temática	x	x	x	x	x
2. Coleta de dados e de informações	x	x	x	x	
3. Definição História e mecânica do Jogo.	x				
4. Desenvolvimento do Level Design.		x			
5. Implementação Lógica do Game			x		
6. Correção de erros.				x	
7. Referencias Bibliográficas					x
8. Revisão do TCC					x
9. Entrega do TCC pronto					x

REFERÊNCIAS

Blockly Games. **Games for tomorrow's programmers.** Disponível em: <<https://blockly-games.appspot.com/>> Acessado em 12/10/2017.

Free Sound Effects. **License Agreement Explained.** Disponível em: <<http://www.freesoundeffects.com/licence.php>> Acessado em 03/10/2017.

Creative Commons. **Attribution 3.0 Unported (CC By 3.0).** Disponível em: <<https://creativecommons.org/licenses/by/3.0/>> Acessado em 03/10/2017.

Creative Commons. **Public Domain Dedication.** Disponível em: <<https://creativecommons.org/publicdomain/zero/1.0/>> Acessado em 05/10/2017.

ATTLUS, Persona 2: Eternal Punishment (2000).

PIGUINSOFT, Cafe Racer (2016).

MICROSOFT, Age Of Empires II The Age Of Kings (1999).

UBSOFT, Grow Home (2015).