



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

Alexandre Zaminato Gueleres

MAZE AND GO

Americana, SP

2018



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

Alexandre Zaminato Gueleres

MAZE AND GO

RELATÓRIO TÉCNICO DESENVOLVIDO EM CUMPRIMENTO À EXIGÊNCIA CURRICULAR DO CURSO SUPERIOR DE TECNOLOGIA EM JOGOS DIGITAIS SOB A ORIENTAÇÃO DO PROF. DR. CLEBERSON EUGENIO FORTE.

Americana, SP

2018

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

G96m GUELERES, Alexandre Zaminato

Maze and Go / Alexandre Zaminato Gueleres. – Americana, 2018.

27f.

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

Orientador: Prof. Dr. Cleberson Eugenio Forte

1. Jogos eletrônicos I. FORTE, Cleberson Eugenio II. Centro Estadual de
Educação Tecnológica Paula Souza – Faculdade de Tecnologia de Americana

CDU:681.6

Alexandre Zaminato Gueleres

MAZE AND GO

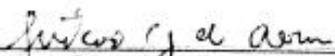
Relatório técnico apresentado como exigência parcial para obtenção do título de Tecnólogo em Jogos Digitais pelo CEETEPS/Faculdade de Tecnologia – FATEC/ Americana.

Americana, 25 de Junho de 2018.

Banca Examinadora:



Cleberson Eugênio Forte (Presidente)
Doutor
Faculdade de Tecnologia de Americana



Gustavo Carvalho Gomes de Abreu (Membro)
Especialista
Faculdade de Tecnologia de Americana



Wagner Siqueira Cavalcante (Membro)
Mestre
Faculdade de Tecnologia de Americana

RESUMO

Com o objetivo de adquirir conhecimento nas áreas de animação, arte, roteiro e sons, foi desenvolvido o jogo Maze and Go, um jogo em plataforma *mobile*, optando pelo sistema operacional Android como mercado alvo para sua distribuição.

Este jogo é um trabalho de gênero híbrido, baseado em quebra cabeça e ação, cuja principal referência e inspiração baseia-se no Pac-Man (Arcade - 1980) e traz desafios em formato de labirinto, com inimigos a serem enfrentados pelo jogador.

O desenvolvimento deste projeto foi baseado em um algoritmo de recursão, disponibilizado pelo programa Maze Generator (2018), o qual é responsável pela geração procedural do mapa, além de posicionar os oponentes e demais itens.

Com a realização deste trabalho foi adquirida experiência em planejar, executar e resolver problemas, além da oportunidade de executar todas as etapas, desde a ideia inicial até a publicação do jogo em uma loja oficial.

Palavras Chave: jogo, quebra cabeça, ação, *mobile*.

ABSTRACT

In order to gain knowledge in the areas of animation, art, script and sounds, the game Maze and Go was developed, a game in mobile platform, opting for the Android operating system as the target market for its distribution.

This game is a puzzle and action genre whose main reference and inspiration is based on Pac-Man (Arcade - 1980) and features maze-shaped challenges with enemies to be faced by the player.

The development of this project was based on a recursion algorithm, provided by the program Maze Generator (2018), which is responsible for the procedural generation of the map, besides positioning the opponents and other items.

With the accomplishment of this work was gained experience in planning, executing and solving problems, besides the opportunity to execute all the steps, from the initial idea until the publication of the game in an official store.

Keywords: *game, puzzle, action, mobile.*

LISTA DE FIGURAS

Figura 1 - Cronograma	11
Figura 2 - Exemplo de Colliders	14
Figura 3 - Itens	14
Figura 4 - Esquema animação do personagem principal	15
Figura 5 - Inimigos.....	16
Figura 6 - Diagrama de máquina de estados	17
Figura 7 - Misturador de som	18
Figura 8 - Botões do menu	18
Figura 9 - HUD do Gameplay	19
Figura 10 - Trecho da história	20
Figura 11 - Implementação Créditos	20
Figura 12 - Menu Opções.....	21
Figura 13 - Menu Pause.....	22
Figura 14 - Game Over	22
Figura 15 - Quantidade de Instalações	23
Figura 16 - Comentários.....	24

GLOSSÁRIO

Assets: São componentes adicionais utilizados em projetos.

Engine: Plataforma de desenvolvimento.

Game Over: Quando o jogador perde

HUD: *Heads-up display*, elementos que a cena contém.

Level Design: Criação da progressão de dificuldades das fases.

Script: Código fonte do jogo.

SUMÁRIO

1.	INTRODUÇÃO	8
1.1.	Sobre o jogo.....	8
2.	METODOLOGIA.....	9
2.1.	Características do jogo	9
2.2.	Ferramentas utilizadas.....	9
2.3.	Análise do mercado.....	10
2.3.1.	Distribuição	10
2.3.2.	Faixa etária.....	10
2.3.3.	Plataforma e sistema Operacional.....	11
2.4.	Cronograma	11
3.	IMPLEMENTAÇÃO	13
3.1.	Gerador de labirinto	13
3.2.	Inteligência artificial	13
3.3.	Personagem principal	13
3.3.1.	Sistema de colisão.....	13
3.3.2.	Itens.....	14
3.3.3.	Controles e jogabilidade	15
3.3.4.	Controle de animação do personagem principal	15
3.4.	Implementação de inimigos.....	15
3.4.1.	Inimigos	16
3.4.2.	Nascimento dos inimigos.....	16
3.5.	Diagrama de máquina de estados.....	17
3.6.	Sistema de controle de fases e história	17
3.7.	Sistema de áudio	17
3.8.	Interface.....	18
3.8.1.	HUD do Jogo	18
3.8.2.	História	19
3.8.3.	Créditos	20
3.8.4.	<i>Level design</i>	20
3.8.5.	Menu de opções	21
3.8.6.	Menu de <i>pause</i> do gameplay	21
3.8.7.	<i>Game over</i>	22

4.	RESULTADOS	23
4.1.	Objetivo inicial	23
4.2.	Resultados obtidos	23
4.3.	Melhorias em relação à pesquisa.....	24
4.4.	Implementações extras	25
4.5.	Link do jogo	25
5.	CONSIDERAÇÕES FINAIS	26
6.	REFERÊNCIAS BIBLIOGRÁFICAS	27

1. INTRODUÇÃO

Um mercado importante para desenvolvedores *indie* é o mercado *mobile*, que abrange um grande público e oferece facilidade para publicar os aplicativos em suas lojas.

Visando aprender mais sobre todos os aspectos que envolvem um game, foi proposta a criação de um jogo para a plataforma *mobile*, com o objetivo de atingir a máxima parcela do mercado que gosta do gênero "Quebra cabeça com elementos de ação".

A utilização da plataforma Unity (2017) permitiu a obtenção de conhecimentos na linguagem de programação C#, além de trabalhar outros elementos, tais como: interface, level design, animações e outros aspectos importantes.

No próximo capítulo será descrita a metodologia utilizada para o desenvolvimento do jogo, exibindo as ferramentas utilizadas, análise de mercado, distribuição, faixa etária e cronograma. Em seguida será abordado a implementação, no qual explica os programas utilizados e passa em todas as etapas da criação do jogo, e por fim será exibido o capítulo com os resultados obtidos, onde é mostrado o objetivo inicial e a melhoria em relação ao feedback dos jogadores.

1.1. Sobre o jogo

No *gameplay*, o jogador controla um personagem, o qual tem objetivo, junto com a auxiliar, borboleta GO, de encontrar a saída de um labirinto gerado proceduralmente, além de encarar inimigos durante o trajeto. O personagem possui uma quantidade de vida pré-definida em cinco corações, perdendo essas vidas de acordo com os ataques recebidos dos inimigos. No decorrer da fase, é possível encontrar poções para recuperá-las, e o jogo acaba quando o jogador chega ao final das duas fases ou perde os cinco corações.

A história do jogo é baseada no personagem principal está tentando encontrar o vilão que o está mantendo preso no labirinto.

2. METODOLOGIA

Nesta seção será descrito o planejamento do jogo, explicando as ferramentas utilizadas, as técnicas e os algoritmos empregados na construção do jogo.

2.1. Características do jogo

O gênero “Quebra cabeça com elementos de ação” foi escolhido porque proporciona ao jogador o desafio de superar sua velocidade de raciocínio e sua capacidade de reflexos, ao receber a missão de passar por um labirinto, gerado proceduralmente, ao mesmo tempo em que enfrenta inimigos por meio de combate corpo a corpo, buscando a maior pontuação possível.

O objetivo principal do jogo é encontrar a saída do labirinto. Para isso, o jogador controla um guerreiro, que possui, como armas padrão, uma espada e um escudo. Para auxiliar o guerreiro a chegar ao final do jogo, existe a "Borboleta GO"-, que o acompanha durante todo o percurso.

O jogo apresenta duas fases, com nível de dificuldade gradativo, portanto, na primeira fase o guerreiro enfrenta três tipos de inimigos, e na segunda fase a quantidade de inimigos é maior e também é adicionada uma armadilha.

Toda a arte do jogo é em 3D, exceto a tela de carregamento e as imagens da tela inicial - o que foi disponibilizado para este trabalho pelo designer gráfico, Heitor Fernandes.

2.2. Ferramentas utilizadas

Para o desenvolvimento deste trabalho foi utilizada a plataforma gratuita Unity, que oferece utilidades para criação e manutenção de interfaces de menu, animações, efeitos sonoros e outros componentes do jogo.

Para a parte artística, além de *assets* gratuitos incluídos na plataforma Unity, foram aplicados recursos disponíveis no site mixamo.com, uma ferramenta para animações de modelos 3D, que também fornece alguns modelos 3D já animados, de forma gratuita. O personagem principal e todos os inimigos foram coletados no Mixamo (2018).

Para o aperfeiçoamento dos áudios e efeitos sonoros, foi empregado o programa Audacity (2018).

O código do jogo, desenvolvido em C#, foi manipulado no Visual Studio (2018). Para o versionamento do código, foi aplicada a ferramenta Git. O código foi hospedado em Github (2018), um site gratuito e aberto a contribuições da comunidade para a melhoria do jogo.

2.3. Análise do mercado

Este trabalho é inspirado no jogo eletrônico japonês Pac-Man, que foi um grande sucesso lançado na década de 1980. Maze and Go assemelha-se ao jogo no qual foi inspirado por possuir um labirinto, inimigos e itens para atingir pontuações. As diferenças de Maze and Go e sua inspiração são a maneira que é exibido o mapa para o jogador, o que, por sua vez, dificulta a percepção da aproximação dos inimigos, além de uma história contada entre as fases. A inspiração em um jogo antigo é proporcionar amistosas competições entre amigos, através da busca de maior pontuação no jogo.

2.3.1. Distribuição

O jogo, em sua versão Beta, está disponível, gratuitamente, na plataforma do Sistema Operacional Android, Google Play Store, gerenciada pela empresa Google.

Segundo pesquisa realizada por Pesquisa Game Brasil (2018), que apresenta como foco principal o levantamento de números referentes ao mercado de games, 70% dos brasileiros optam por jogos gratuitos e 75,5% deles aceitam anúncios publicitários dentro dessas condições. Sendo assim, pretende-se adotar uma forma de monetização, através da publicidade dentro do aplicativo, após a popularização deste jogo.

2.3.2. Faixa etária

De acordo com as recomendações da ESRB: Entertainment Software Rating Board (2018), o acesso a este jogo é indicado e permitido para pessoas com idade acima de 12 anos, visto que o conteúdo do mesmo pode apresentar violência, temas

sugestivos, humor cruel, há sangue, jogos de azar e/ou uso moderado de linguagem forte.

Observa - se, portanto, que este jogo pode atingir a maior parcela da população que se interessa pelos gêneros "Quebra cabeça" e "Ação".

2.3.3. Plataforma e sistema Operacional

De acordo com a Pesquisa Game Brasil, os celulares são a plataforma mais jogada com 62% dos votos.

A publicação para o sistema operacional Android foi selecionada por ter maior facilidade de lançamento via *Google Play Store* e também pelo fato da maioria das pessoas ter acessos a celulares que possuem esse sistema.

2.4. Cronograma

A figura 1 apresenta o cronograma de atividades para o desenvolvimento deste projeto.

Figura 1 - Cronograma

Mês/Tarefas	Fevereiro		Março		Abril		Maio		Junho	
	1º	2º	1º	2º	1º	2º	1º	2º	1º	2º
Planejamento	■	■	■	■						
Desenvolvimento			■	■	■	■				
Animação e Som					■	■				
História						■				
Documentação							■	■		
Testes							■	■		
Entrega									■	
Apresentação e Entrega do Documento										■

Fonte: Próprio Autor

Planejamento: Nesta etapa foram definidas a ideia principal do jogo, bem como foram estudadas as ferramentas que seriam empregadas em seu desenvolvimento.

História: Neste momento foram criadas as histórias do personagem e história sobre o ambiente.

Desenvolvimento: Neste ponto foram estipuladas tarefas para o que o jogo pudesse ter um desenvolvimento rastreável e evitar geração de problemas.

Animação e Som: Neste estágio foram tratados as animações e sons dos personagens, itens e da armadilha.

Documentação: Esta etapa foi reservada para digitação desse relatório técnico.

Testes: Neste ciclo foi selecionado o mesmo período que a etapa de documentação para que se pudesse colher avaliações dos jogadores e munir o relatório com as mesmas.

Montagem da Apresentação: Momento reservado para criação da apresentação do jogo para avaliação.

3. IMPLEMENTAÇÃO

Para o desenvolvimento do jogo foram utilizados dois principais programas que serão abordados em tópicos separados, de modo que descrevem seu conteúdo e aplicação de forma clara.

Além disso, serão abordados tópicos sobre o personagem principal, inimigos, diagrama de máquina de estados, fluxo da fase, sistema de áudio e interface.

3.1. Gerador de labirinto

O *software* Maze Generator (2018), que consiste em um *script* de recursividade que facilita a geração de labirintos aleatórios, foi utilizado para a geração procedural do mapa de todo o jogo, portanto foi fundamental. O código foi aprimorado para prover, também, a criação dos inimigos e itens adicionais.

3.2. Inteligência artificial

O *software* A* Pathfinding Project (2018), que é um *script* que habilita a possibilidade de se configurar elementos no jogo para buscar e chegar até um alvo, foi utilizado especificamente para gerar a inteligência artificial das personagens do jogo. O código deste programa não foi alterado. Apenas ocorreu a configuração dos obstáculos e dos personagens. Portanto, A* Pathfinding Project foi essencial para fazer com que os inimigos encontrassem o personagem principal e também para que a Borboleta Go, auxiliar do jogador, encontrasse o final do labirinto.

3.3. Personagem principal

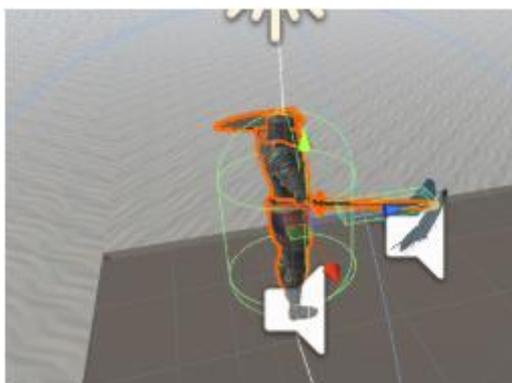
Neste tópico serão descritas as implementações referentes ao personagem principal e suas interações com itens utilitários.

3.3.1. Sistema de colisão

O player tem um tipo de verificação, caso ataque ou defenda golpes, que é o método "*OnTriggerEnter3D*", que acionam diversos códigos para ativar animações,

tais como as relacionadas a recebimento de dano, para remoção da vida, tanto do inimigo, quanto do personagem. Há um sistema de colisão para verificar caso o inimigo ataque o personagem, chamado “*DetectHero*” que, caso seja executado, ativa a animação de ataque do inimigo e, caso o personagem saia dessa distância do *collider*, que é quando ocorre o contato dos personagens, então o inimigo volta a procurar o personagem até ativar novamente essa detecção, e cada arma tem um *collider* (Figura 2) que é verificado caso acerte o alvo.

Figura 2 - Exemplo de Colliders



Fonte: Próprio Autor

3.3.2. Itens

Quando o jogador colidir com algum elemento, tais como a moeda ou a poção, ele receberá pontos. (Figura 3).

Poção: O jogador recupera um coração perdido, e ganha cinco pontos.

Moeda: A moeda dá ao jogador um montante de 10 pontos.

Figura 3 - Itens



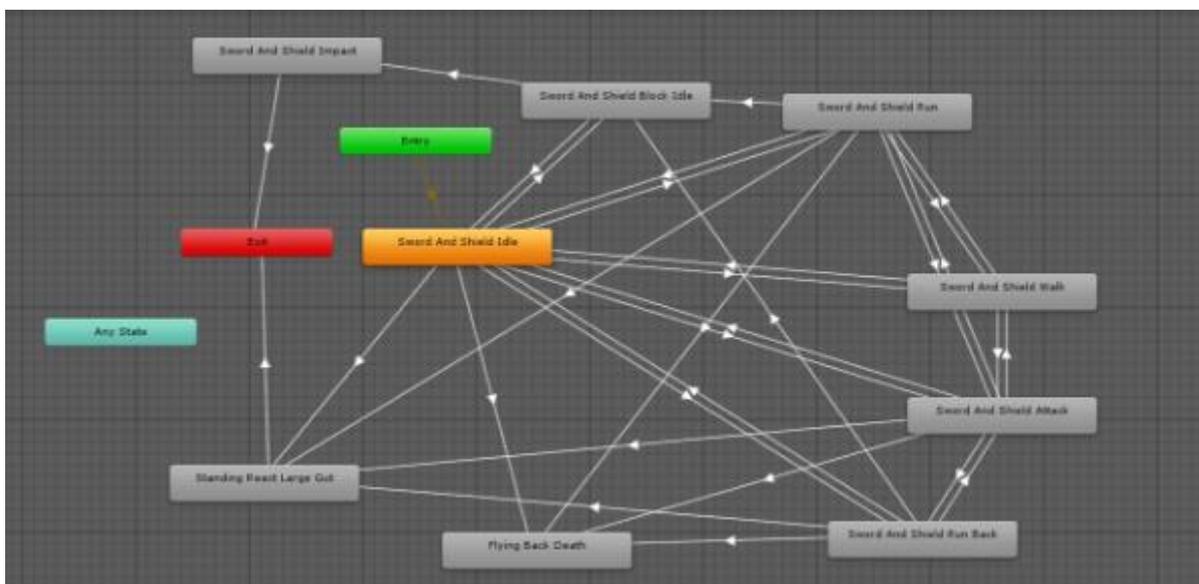
Fonte: Próprio Autor

3.3.3. Controles e jogabilidade

O controle do personagem utiliza o pacote de comandos para dispositivos móveis do próprio Unity, podendo movimentar o personagem para frente, para trás e para os lados, como se fosse um controle analógico e possui três outros botões de ação que ativam o ataque, defesa e também a borboleta para ajudar o jogador a encontrar o final do labirinto, mostrando o caminho correto, com o custo de 100 pontos do jogador.

3.3.4. Controle de animação do personagem principal

Figura 4 - Esquema animação do personagem principal



Fonte: Próprio Autor

Na figura 4 são mostradas as diversas animações do personagem que utilizam variáveis para iniciá-las de acordo com o que ocorre durante o jogo. Cada variável é controlada através de códigos que são verificados com as colisões entre personagens e inimigos. As animações possuem sons distintos para todas as etapas.

3.4. Implementação de inimigos

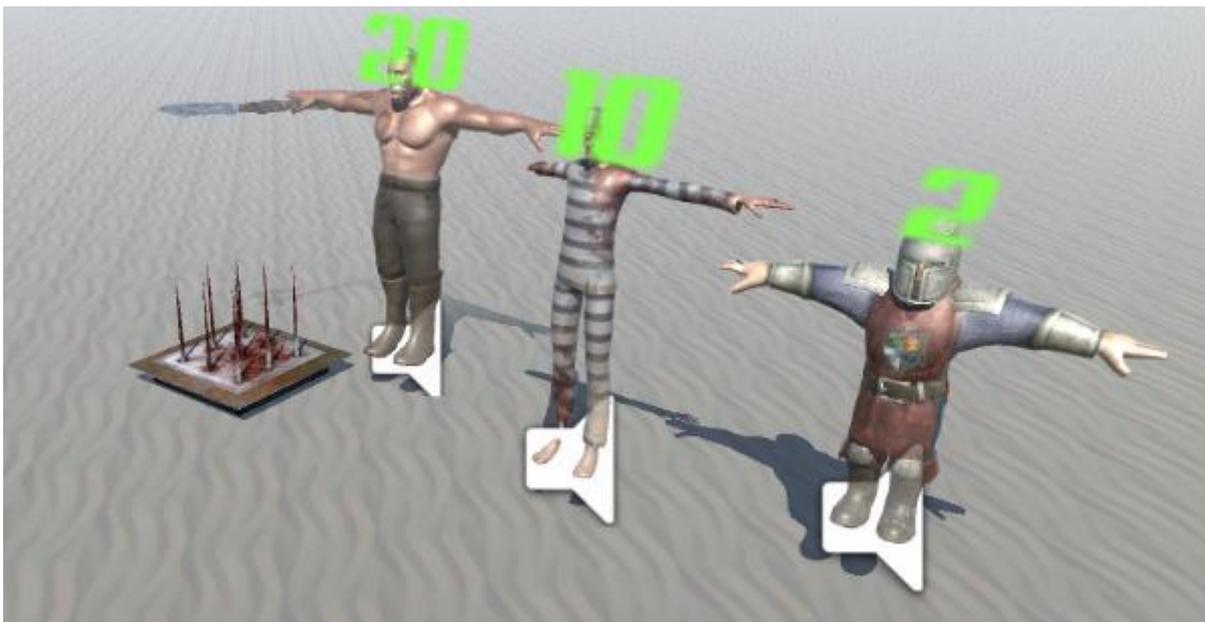
A implementação de inimigos possui menos animações, porém são controladas pela detecção do jogador. Caso um deles passe perto do personagem

principal, é acionado o comando para os outros inimigos buscarem o personagem utilizando o algoritmo A*. Possuem sons de passos, quando recebem ataques e também quando atacam.

3.4.1. Inimigos

O jogo contém três tipos de inimigos (Figura 5), controlados pelo mesmo código, sendo que a diferença entre eles são os modelos 3D, sons, pontos de vida e dano de ataque e a chance de nascerem nas fases. Além desses três inimigos, o jogo possui uma armadilha, que possui animação que, a cada quatro segundos, são disparados e o jogador necessita aguardar o recolhimento da mesma para poder passar sem levar dano.

Figura 5 - Inimigos



Fonte: Próprio Autor

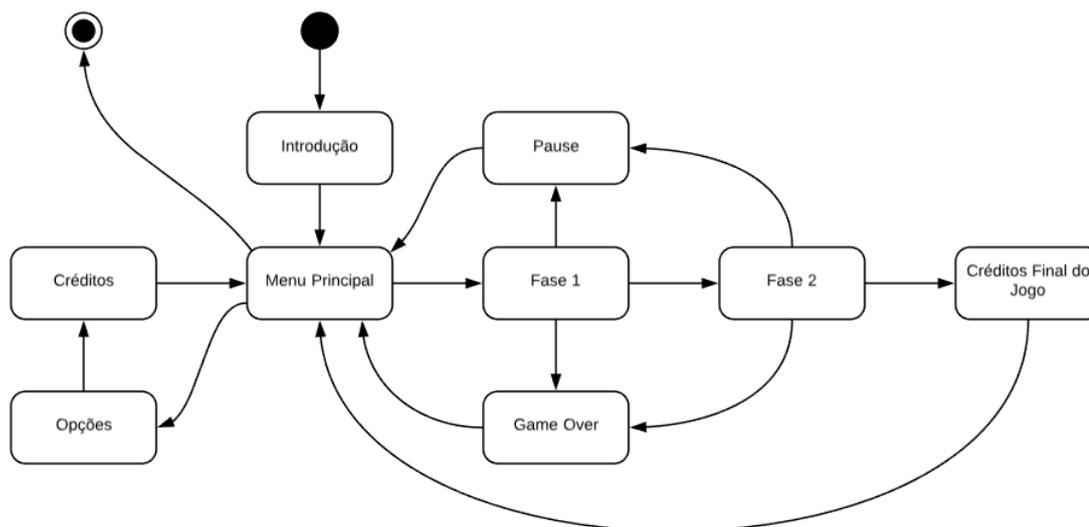
3.4.2. Nascimento dos inimigos

Os inimigos são criados juntamente com a geração do labirinto. É sorteado um número entre zero e dez e, caso o número seja zero, o código entra no modo de gerar objetivos que dão pontos, que no caso são os inimigos, poções e moedas e, se o jogador for à fase dois são adicionadas armadilhas, e seguem uma ordem de

importância, sendo que a chance de nascer uns inimigos mais fracos, moedas e poções é maior do que nascer o mais forte.

3.5. Diagrama de máquina de estados

Figura 6 - Diagrama de máquina de estados



Fonte: Próprio Autor

Na figura 6 é exibido o diagrama de estados das cenas do jogo, que é iniciado pela introdução.

3.6. Sistema de controle de fases e história

O jogo segue o fluxo de não salvar a última campanha do jogador. Após ocorrer o *Game Over*, a única informação salva é a pontuação. Portanto, o jogador sempre iniciará na fase um, além de ter a pontuação zerada.

A história é contada nas telas de carregamento do jogo.

3.7. Sistema de áudio

O sistema de áudio possui um misturador de som (Figura 7), que possui dois canais, sendo eles o de música e de efeitos sonoros, possibilitando o controle

independente dessas duas funcionalidades, por exemplo, remover o som da música do jogo.

O jogo possui quatro músicas, uma para o menu, uma para a fase um, outra para a fase dois e por último uma música para os créditos finais.

Figura 7 - Misturador de som



Fonte: Próprio Autor

3.8. Interface

A interface emprega uma visão minimalista para que o jogador possa realizar as tarefas essenciais de maneira intuitiva. Na figura 8 vê-se que o jogador tem as possibilidades de apertar na engrenagem para mostrar as opções, pressionar no botão de jogar para iniciar, e no "X" para sair do aplicativo.

Figura 8 - Botões do menu



Fonte: Próprio Autor

3.8.1. HUD do Jogo

Na figura 9 é mostrado o *Heads-up display* (HUD) do jogo que foi elaborado para o jogador se sentir confortável com a visão do personagem e ambientes,

exibindo somente os botões e informações necessários para o jogador continuar progredindo no jogo. Um grande desafio enfrentado na criação do HUD foi o tamanho dos botões, pois como são dispositivos móveis, as telas e resolução variam os tamanhos e o jogo sempre é executado no modo paisagem.

Figura 9 - HUD do Gameplay



Fonte: Próprio Autor

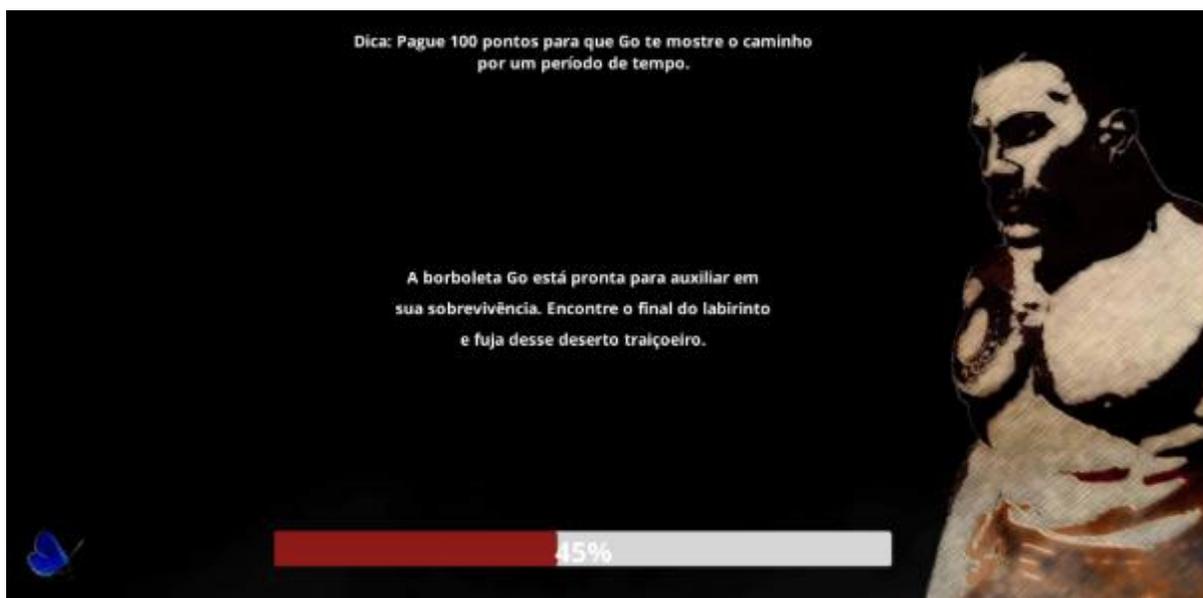
3.8.2. História

A história é um ponto difícil para ser elaborada para dispositivos móveis, pois os jogos dessa plataforma são geralmente casuais.

Com isso, a história está bem simples, de maneira que o jogador busca sair do labirinto com a ajuda da borboleta, porém, quando consegue sair da primeira fase, entra em um novo labirinto e o objetivo se torna encontrar quem está fazendo isso com o personagem. Ao término da segunda fase, ele descobre quem está fazendo isso e será a primeira etapa para a continuação de desenvolvimento desse jogo, a batalha contra o vilão.

Na figura 10 é exibido o exemplo de como é mostrada a história para o jogador.

Figura 10 - Trecho da história



Fonte: Próprio Autor

3.8.3. Créditos

Com a utilização de diversos *assets*, a tela de crédito (Figura 11) ficou com bastante informações e foi distribuída de maneira que ficasse elegante e de fácil leitura para não poluir a tela do jogador.

Figura 11 - Implementação Créditos



Fonte: Próprio Autor

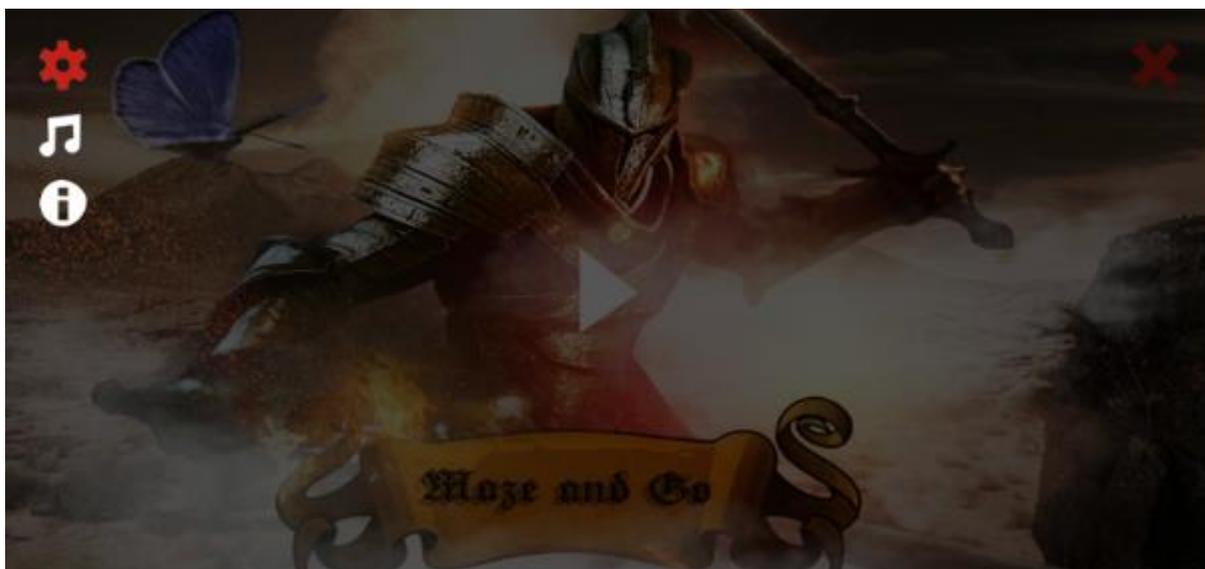
3.8.4. Level design

O *level design* foi criado para que a primeira fase seja apresentada primeiramente os inimigos mais fracos, para depois irem surgindo inimigos mais poderosos, para que o jogador não se sinta intimidado e desista de jogar, e na segunda fase é adicionado um novo elemento já no começo da fase, para entender como funcionará esse novo desafio e a aparição de mais inimigos durante o trajeto.

3.8.5. Menu de opções

A figura 12 mostra a tela de opções com apenas duas possibilidades, que são: Ativar/Desativar a música e a exibição dos créditos.

Figura 12 - Menu Opções



Fonte: Próprio Autor

3.8.6. Menu de *pause* do gameplay

A figura 13 apresenta o menu de *pause* que exibe as opções de ativar/desativar músicas e a opção de voltar ao menu inicial. Esse menu de *pause* não cobre nenhuma outra informação necessária do jogo. Caso o jogador pause, ele irá conseguir ver sua vida, pontuação e botões para jogar.

Figura 13 - Menu Pause



Fonte: Próprio Autor

3.8.7. *Game over*

Na figura 14 é possível ver a imagem de *game over* que aparece centralizada na tela do jogo com as informações de melhor pontuação e pontuação nessa rodada, e também um botão para voltar ao menu inicial.

Figura 14 - Game Over



Fonte: Próprio Autor

4. RESULTADOS

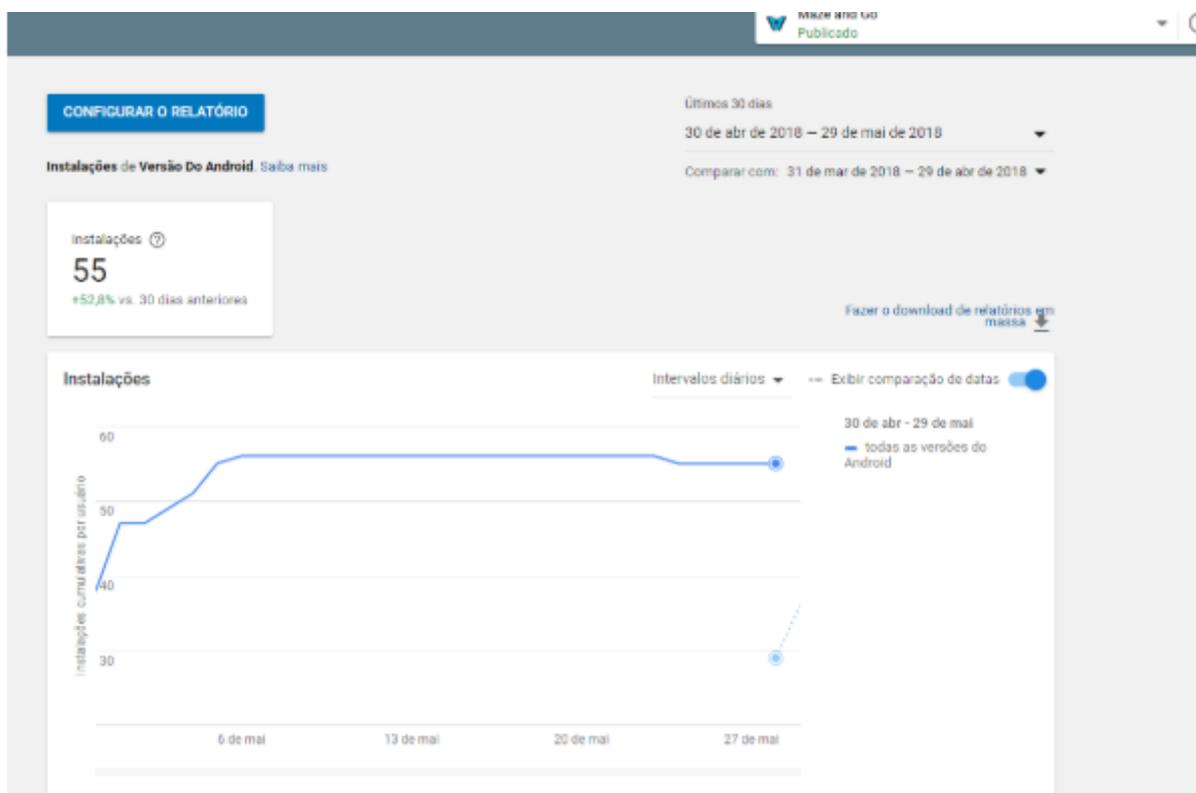
4.1. Objetivo inicial

A ideia inicial era o jogo possuir pelo menos 100 instalações até o final do mês de maio, divulgando-o através apenas de amigos e grupos relacionados a jogos e computação nas redes sociais.

4.2. Resultados obtidos

O jogo, até momento da escrita deste, possui 55 instalações (Figura 15) e 10 comentários. Ressalta-se que não houve investimento financeiro para impulsionar a divulgação do mesmo através das redes sociais, o que poderia ter gerado ainda mais downloads.

Figura 15 - Quantidade de Instalações



Fonte: Próprio Autor

4.3. Melhorias em relação à pesquisa

Assim como se observa na figura 16, os comentários mais recorrentes obtidos como *feedback* dos players que fizeram o *download* e avaliaram o jogo em questão, tratam da sensibilidade de movimentação do personagem.

Figura 16 - Comentários



Gerson Donscoi, 28/04/2018, 14:55

O controle de movimento é rápido para virar e devagar para começar a movimentar para frente, isso dificulta um pouco o direcionamento do boneco, o fato de ser um labirinto somado a sensibilidade de girar a câmera dificulta a orientação do jogador, a partir da quinta curva já não sei mais de onde vim. Os controles de ataque e defesa estão perfeitos e quando precisei apareceu poções de vida, fazendo com que eu continuasse a enfrentar os inimigos. Ótimo projeto

[Postar uma resposta privada](#)

Aparelho: LG G6 (lucye)
Versão do app: 1 (1.0)



Leandro Lima, 28/04/2018, 14:28

Alexandre... Graficamente ficou bem bacana...acho que a Sensibilidade do console ficou muito alta... Meus parabéns pelo projeto

[Postar uma resposta privada](#)

Aparelho: VIBE K6 (K33b36)
Versão do app: 1 (1.0)

Fonte: Próprio Autor

Uma solução possível ao problema detectado seria de apenas diminuir a sensibilidade e rotação do personagem.

A melhor solução, no entanto, seria a de mudar completamente o tipo de controle do personagem, deixando a câmera fixa para que o jogador consiga se lembrar dos caminhos e o controle respeitasse a orientação da câmera para se movimentar ao invés de fazê-lo da perspectiva do personagem.

4.4. Implementações extras

As implementações extras seriam a fase final para a batalha com o vilão da história e desenvolver uma maneira mais atrativa de se contar a história.

4.5. Link do jogo

O jogo está disponível na versão beta para dispositivos android no seguinte endereço:

<https://play.google.com/store/apps/details?id=com.alegueleres.MazeAndGo>

<https://github.com/alegueleres/fatec-game>

5. CONSIDERAÇÕES FINAIS

Considerando que este jogo foi desenvolvido por um único autor, no período de cinco meses, foram atingidos com êxito os objetivos de adquirir conhecimento e aplicar parte do aprendizado obtido no Curso no desenvolvimento deste projeto final, ao ponto de publicá-lo para ser jogado por outras pessoas.

É reconhecido o fato de que este projeto possui pontos que não foram explorados por este autor, como a parte artística e os modelos em 3D, que foram obtidos através da plataforma Unity e do site Mixamo. Apesar disso, foi empregado cuidado com o código para que fosse bem produzido, evitando redundâncias e demais falhas.

Como pontos a serem melhorados estão, à manutenção do código utilizado e a organização das classes, em relação à definição de suas funcionalidades e, principalmente, a resolução da mecânica de movimentação empregada no jogo, de modo que as observações dos usuários que baixaram o jogo sejam atendidas.

Objetiva-se, por fim, lançar uma versão final do jogo para trazer monetização e concluir o ciclo da história da personagem.

6. REFERÊNCIAS BIBLIOGRÁFICAS

A* Pathfinding Project. Version 4.1.16. [S.I.]: Arongranberg, 2018. Disponível em: <<https://arongranberg.com/astar/>>. Acesso em: 25 fev. 2018.

AUDACITY. Version 2.2.2. [S.I.]: Audacity, 2018. Disponível em: <<https://www.audacityteam.org/>>. Acesso em: 07 abr. 2018.

ESRB: Entertainment Software Rating Board. Version 2018. [S.I.]: Entertainment Software Association, 2018. Disponível em: <<http://www.esrb.org>>. Acesso em: 25 mar. 2018.

GITHUB. Version 2018. [S.I.]: GitHub, 2018. Disponível em: <<https://github.com/>>. Acesso em: 31 maio 2018.

MAZE Generator by STYANTON. Version 1.0. [S.I.]: Unity Asset Store, 2018. Disponível em: <<https://assetstore.unity.com/packages/tools/modeling/maze-generator-38689>>. Acesso em: 20 fev. 2018.

MIXAMO. [S.I.]: Adobe Systems Incorporated, 2018. Disponível em: <<https://www.mixamo.com>>. Acesso em: 04 mar. 2018.

PESQUISA Game Brasil. Version 2018. [S.I.]: PGB, 2018. Disponível em: <<https://pesquisagamebrasil.com.br/>>. Acesso em: 21 mar. 2018

UNITY, Version 2017.1.1. [S.I.]: Unity, 2017. Disponível em: <<https://unity3d.com/pt>>. Acesso em: 02 fev. 2018.

VISUAL Studio. Version Community. [S.I.]: Microsoft, 2018. Disponível em: <<https://www.visualstudio.com>>. Acesso em: 03 fev. 2018.