

Fases de Desenvolvimento do Jogo Final Gang War

Rômulo Cadó Dorneles

Departamento de Engenharias e Ciência da Computação – Universidade
Regional Integrada do Alto Uruguai e das Missões (URI)
Caixa Postal 181 – 97.700-000 – Santiago – RS – Brasil

romulocado@gmail.com

Abstract. *This article aims to describe and document the development stages of a 2D beat 'em up game. It will document the game's conceptualization phase, including storyboard creation and the definition of mechanics and gameplay. It will also list the technologies used to implement the required functionalities. Additionally, it will document the stages and animations used in the game and provide an overview of the final product.*

Resumo. *Este artigo tem por objetivo descrever e documentar fases de desenvolvimento de um jogo 2d beat 'em up. Documentar a fase de concepção do jogo como criação de storyboard, definição de mecânicas e gameplay. Listar as tecnologias utilizadas para o cumprimento das funcionalidades necessárias. Também documentar as fases e animações utilizadas no jogo, apresentando como ficou o game pronto.*

1. Introdução

O desenvolvimento de um jogo é um processo complexo e envolve uma série de etapas que devem ser cuidadosamente planejadas para garantir que o jogo atenda às expectativas dos jogadores. Um desses processos fundamentais é o game design, que envolve a definição da jogabilidade, das regras, do foco do jogo, dos personagens, da história e de outras mecânicas importantes. É nesse momento que é definido a essência do jogo e estabelece as bases para a criação de uma experiência envolvente e cativante. É importante que o game design seja iterativo, permitindo ajustes e melhorias ao longo do processo de desenvolvimento, garantindo que o jogo atinja seu potencial máximo.

Ao conceber um jogo, é fundamental que as características mais importantes sejam claramente definidas desde o início, tais como a narrativa e a jogabilidade, com um tema e enredo bem estruturados. É preciso definir com clareza qual será o foco do jogo, se ele será voltado para a jogabilidade, com um enredo menos elaborado como pano de fundo, ou se o enredo será o ponto forte do jogo, com cinemáticas e desenvolvimento de personagem e história bem elaborados. Além disso, é crucial definir as tecnologias a serem utilizadas, bem como decidir se o jogo será em 2D ou 3D, planejando a física dos personagens e objetos com antecedência e criando um storyboard para esboçar as cenas e definir a estrutura do jogo de forma eficaz antes de começar o desenvolvimento. Essas medidas ajudam a garantir um processo de desenvolvimento mais eficiente e a criação de um jogo que atenda às expectativas desde o início, evitando dúvidas em alguma etapa e qual será o foco e objetivo do jogo.

2. Game Design

O game design do jogo a ser desenvolvido Final Gang War, teve como idealização ser um jogo focado na jogabilidade, e seguir um estilo clássico beat 'em up, que é um gênero de jogo de ação onde o jogador controla um personagem que enfrenta hordas de inimigos em uma sucessão de combates, muitas vezes com elementos de plataforma e side-scrolling 2D, onde o personagem anda somente em uma direção até chegar ao fim da fase. O objetivo é avançar através dos estágios, derrotando os inimigos e os chefes, enquanto se recolhe itens para recuperar a saúde e aumentar o poder do personagem. Os jogos beat 'em up costumam ter uma jogabilidade simples e intuitiva, com controles básicos de

ataque, defesa e movimento, além de combos e habilidades especiais. O estilo é popular desde a década de 1980, e tem sido usado em muitos jogos de sucesso, como Final Fight, Streets of Rage e Castle Crashers.

O storyboard foi criado para que se tenha uma visão geral do jogo, facilitando a identificação de problemas e o planejamento das ações necessárias para corrigi-los. Ele também ajuda a definir a aparência visual do jogo, incluindo personagens, cenários e objetos, e permite que a equipe teste a mecânica do jogo e ajuste a jogabilidade antes de iniciar a produção completa. Ele é uma ferramenta importante no desenvolvimento de jogos, especialmente para jogos com enredo e narrativa. Ele é uma espécie de "mapa" visual do jogo, mostrando a sequência de eventos, cenas e diálogos que ocorrem durante o jogo.



Figura 1. Primeiras 3 cenas definidas no storyboard, onde mostra como será o ambiente do jogo 2D side-scrolling, como personagem principal, inimigos e power-ups.



Figura 2. Outras três cenas do storyboard onde mostra a personagem principal chegando no local do chefe principal, onde os inimigos aumentam a dificuldade e é enfrentado o segundo chefe secundário

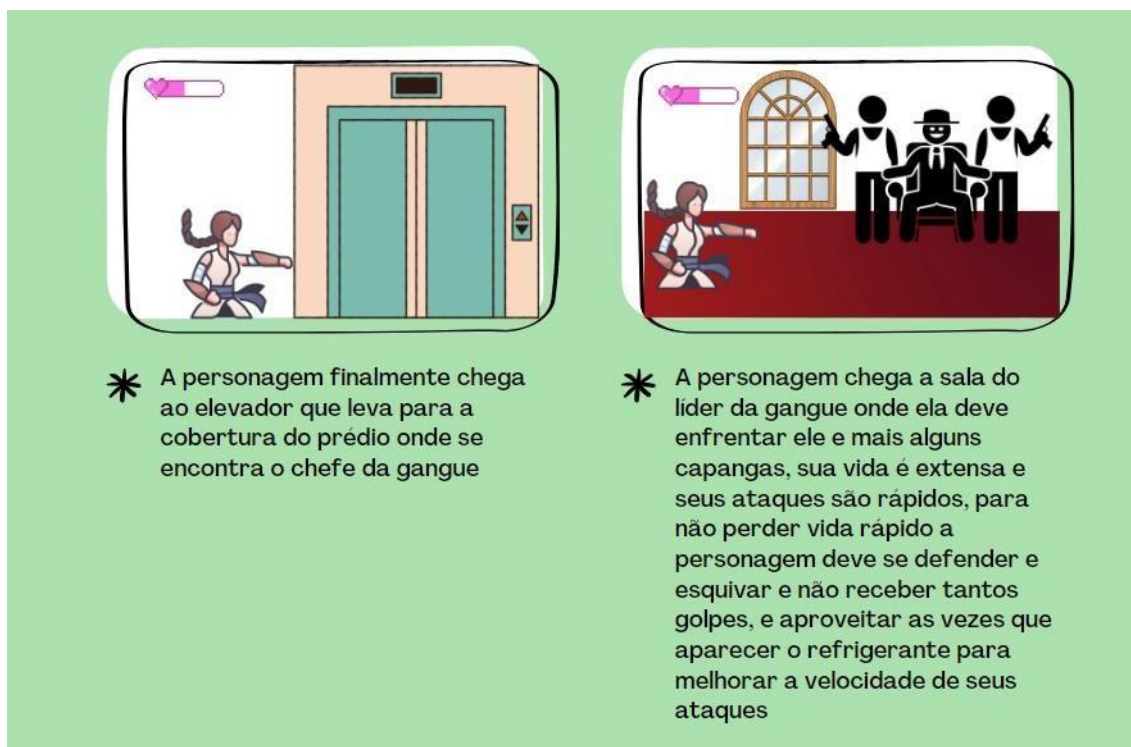


Figura 3. Story Board com as ultimas duas cenas onde mostra a personagem principal encontrando o último chefe e principal desafio do jogo.

3. Tecnologias Utilizadas

Foi escolhido como plataforma de desenvolvimento do jogo a plataforma Unity com linguagem de programação C#. Ao escolher a Unity como plataforma de desenvolvimento, os criadores de jogos têm acesso a uma poderosa engine que oferece recursos abrangentes para criar mundos virtuais, animações, física e interações de personagens. A linguagem de programação C# é amplamente utilizada na Unity e permite criar a lógica do jogo, implementar mecânicas de combate e controlar o comportamento dos personagens. Uma das principais funcionalidades de desenvolver um jogo beat 'em up na Unity é a capacidade de criar sistemas de combate complexos. Pode-se implementar uma variedade de movimentos de ataque, combinações de golpes, habilidades especiais e contra-ataques. Com a ajuda da física da Unity, é possível simular impactos realistas e proporcionar uma sensação autêntica de luta. Além disso, a Unity permite a criação de níveis e ambientes interativos [Unity 2023].

Permite construir cenários detalhados, com elementos destrutíveis, objetos interativos e armadilhas para aumentar a variedade e desafio do jogo. A Unity também oferece suporte a recursos de inteligência artificial, permitindo que os inimigos sejam programados para reagir ao ambiente e ao comportamento do jogador, aumentando a dificuldade e a imersão. A criação de personagens personalizados é outra funcionalidade importante. Permite projetar seus próprios personagens, definir suas habilidades, características e atributos únicos. A Unity oferece ferramentas de animação poderosas, permitindo a criação de movimentos fluidos e expressivos para os personagens, tornando a jogabilidade mais envolvente. A integração de áudio e efeitos sonoros é essencial para criar uma experiência imersiva. A Unity permite a importação e reprodução de trilhas sonoras, efeitos sonoros e vozes para dar vida ao jogo. Isso contribui para a atmosfera e a narrativa, tornando o jogo mais cativante para os jogadores [Unity 2023].

A Unity possui um sistema de física integrado que facilita a detecção de colisões

e a resposta aos eventos de colisão. Ela oferece uma variedade de colisores predefinidos, como colisores de caixa, esfera e cápsula, que podem ser facilmente adicionados aos objetos do jogo. Além disso, fornece uma interface visual intuitiva para definir e ajustar as propriedades das colisões, tornando o processo mais fácil para os desenvolvedores. Também é possível programar colisões personalizadas usando scripts em C#, permitindo maior flexibilidade e controle sobre o comportamento das colisões.

No que se refere às animações, a Unity possui um sistema de animação robusto e amigável. A Unity suporta a criação de animações por keyframes, blend trees, máquinas de estados e animações baseadas em física. A transição entre animações e a sincronização com eventos de jogo também podem ser facilmente configuradas. Com o recurso de animação Mecanim da Unity, é possível criar transições suaves entre animações, controlar parâmetros de animação e até mesmo sincronizar animações com o áudio [Unity 2023].

Esses aspectos de controle de animações e colisões de objetos são extremamente importantes para a mecânica de beat 'em up, onde o personagem principal deve acertar movimentos de lutas nos inimigos, e quando for atingido, levar dano referente ao golpe dado pelo inimigo. Em outras bibliotecas esse controle exige um controle manual matemático, por meio de geometrias de colisões e detecção de objetos, que a Unity facilita pela ferramenta abstrair essa parte manual matemática de posição de objetos e colisões.

4. Assets utilizados

Os assets referem-se a todos os recursos utilizados para compor e construir o mundo virtual do jogo. Eles são elementos essenciais para a criação de gráficos, sons, animações, modelos 2D ou 3D, texturas, scripts, músicas e qualquer outro componente necessário para o funcionamento e a estética do jogo. Para o desenvolvimento do jogo foi utilizado assets gratuitos disponibilizados na web, onde as animações e elementos para montar os mapas são específicos de estéticas beat 'em up, sendo elementos como trilha sonora e alguns mapas do consagrado jogo Street of Rage.

Para animar os personagens, os inimigos, o boss, entre outros elementos gráficos foram utilizados Sprites gráficos. Sprites são um tipo específico de asset usado principalmente em jogos 2D. Eles são imagens bidimensionais que representam objetos, personagens, cenários e elementos visuais dentro do jogo. Os sprites na Unity são usados para criar a aparência visual dos elementos do jogo, como personagens, inimigos, objetos interativos, cenários, efeitos visuais, entre outros. Eles podem ser animados para criar movimento e interação com o ambiente ou outros personagens. Na Unity, os sprites são organizados em spritesheets, que são imagens que contêm várias animações ou quadros individuais. Ao importar um spritesheet na Unity, é possível dividir automaticamente os quadros e criar animações a partir deles. Isso facilita a criação de animações fluídas para os personagens e outros elementos do jogo. Os sprites possuem componentes e propriedades específicas, como colisores, que permitem a detecção de colisões e interações com outros objetos do jogo. Eles são configurados com scripts para controlar seu comportamento e interações com o jogador ou o ambiente. Abaixo estão os sprites utilizados para criação visual do jogo.



Figura 4. Sprites de movimentação e combate da personagem principal e dos inimigos punks

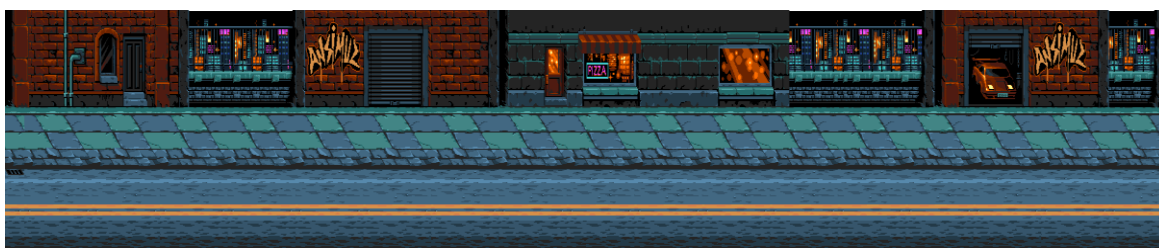


Figura 6. Cenário da fase 1



Figura 7. Cenário da fase 2

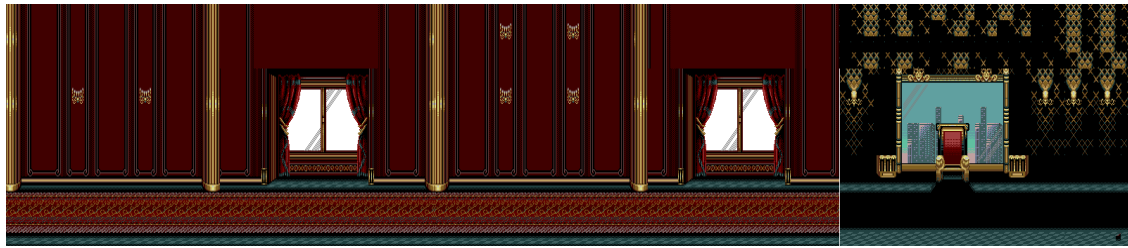


Figura 8. Cenário da fase 3 e última do jogo

Ao criar os mapas para as fases disponíveis no game foi utilizado o TilesetMapper, que é um software de mapeamento de tilesets para criar, editar e organizar mapas ou cenários 2D em jogos que utilizam tiles [TiledsetMapper 2023]. Ele fornece uma interface gráfica para posicionar e configurar os tiles de forma eficiente, permitindo aos desenvolvedores criar ambientes visualmente interessantes para seus jogos, ou seja ele verifica cenários png e divide-os em blocos de acordo com a altura e largura de blocos que terá o mapa desenvolvido. Para animar a movimentação e combate dos personagens foram utilizados o Animator e Animation Controller em conjunto, pois permitem criar animações complexas e controladas para objetos e personagens em um jogo. O Animation Controller atua como um ponto central de controle para gerenciar o fluxo de animações, enquanto o Animator executa as animações com base nas configurações definidas no Animation Controller. Isso proporciona flexibilidade e controle para criar movimentos, transições e interações fluidas e realistas em seu jogo.

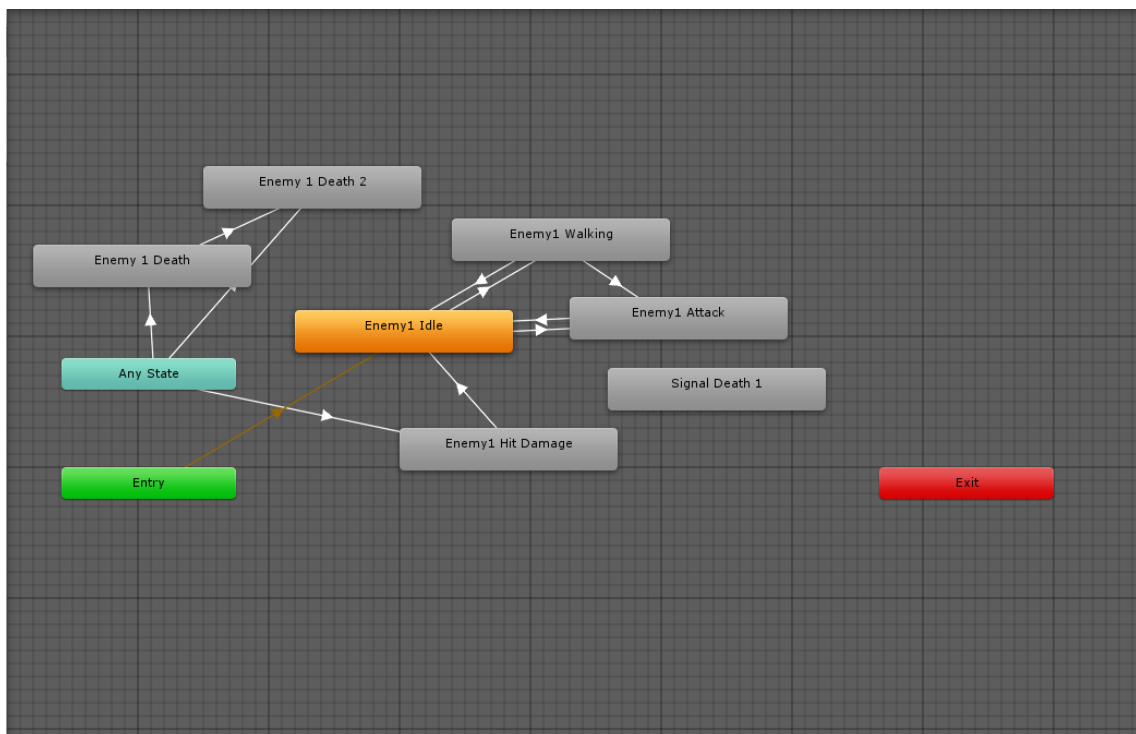


Figura 9. Exemplo do Animation Controller do fluxo de animação do inimigo punk

5. Disposição do jogo e seus elementos

O jogo possui a seguinte disposição e organização de fluxo, primeiramente é apresentada uma tela com a cidade no fundo, e um texto exibindo a breve descrição da história, dando contexto ao jogador da motivação da personagem e situar sobre a jornada que será percorrida, que serve como pano de fundo ao gameplay. Seguindo é exibido a tela principal do jogo com a imagem artística de alta resolução da personagem principal do jogo, e depois o jogador é levado a seleção de personagem que mostra a personagem principal chamada barbie, e seus movimentos de lutas no menu. Os movimentos de luta

da barbie são jabs, assim como socos cruzados, chute, pulo, soco alto com chute e voadora. Os inimigos tem um jab e um cruzado somente, já o chefe das fases possui uma arma como boomerang ao qual ele atira na personagem, assim como movimentos mais rápidos que inimigos comuns.

Posteriormente o jogador inicia a primeira fase, que tem na tela as informações no HUD de vida da barbie e quantidade de tentativas para terminar o jogo, vida dos inimigos quando receberem dano de soco ou chutes da personagem, no mapa são distribuídos frangos que são os elementos de recuperar a vida da personagem. A personagem não consegue ir ou voltar até derrotar todos os inimigos da área, e durante o percurso um aviso é colocado na tela com a seguinte frase: “Para seu principal inimigo derrotar, o chão não deve tocar”. Que significa que quando o jogador chegar o chefe da fase ele deve notar com o aviso e com as tentativas, que o chefe só toma dano de golpes da barbie que são com pulos, ou seja socos altos e voadoras, esse seja o puzzle do game que foi um desafio colocado durante o desenvolvimento. Ao passar pelas três fases e conclui-las com vida suficiente, vem a tela parabenizando o jogador pela conquista e dando desfecho a história da barbie.

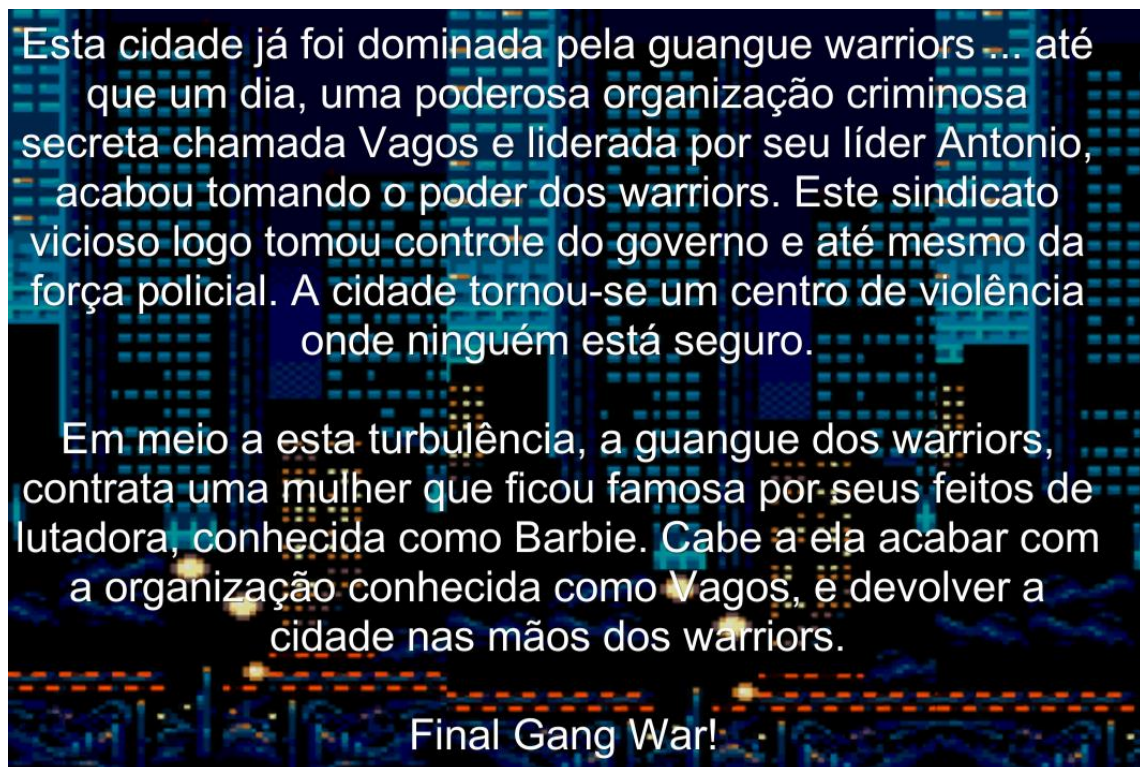


Figura 10. Tela de apresentação inicial e contextualização da história



Figura 11. Tela principal com arte da personagem principal

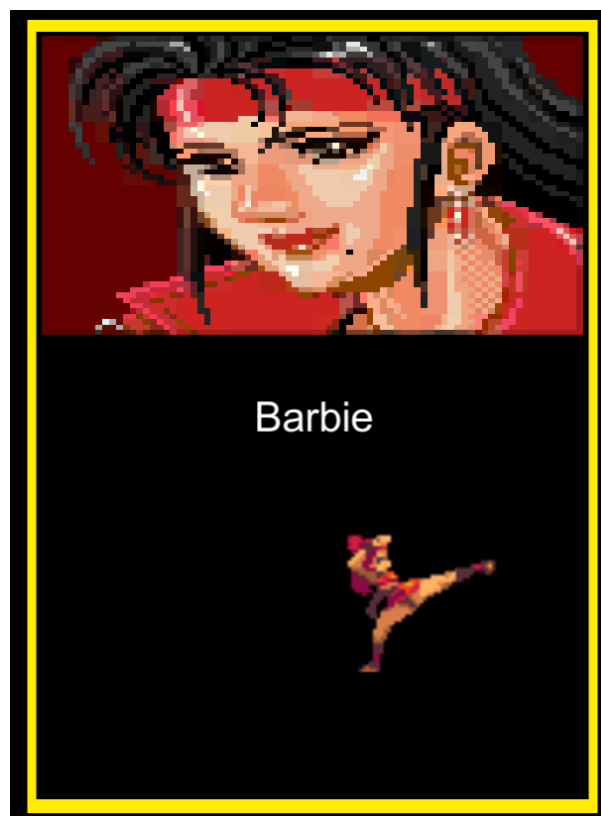


Figura 12. Tela de escolha de personagem



Figura 13. Primeira fase do jogo com o HUD das informações de vida, assim como os elemento de recuperação de vida, e também a dica do puzzle do game sobre como derrotar o boss da fase.

6. Conclusão

É evidente que são inúmeras as etapas e desafios que foram vistos e documentados durante o desenvolvimento de um jogo estilo beat 'em up, com mecânicas e gameplay simples, que mesmo tendo um mecânica simples e uma física mais simples, trouxe inúmeros desafios para o jogo ficar funcional e cumprir o planeamento do story board. Durante a fase inicial de concepção, foi essencial definir uma visão clara do jogo, estabelecendo mecânicas, estética visual, narrativa e objetivos. Isso requer habilidades de planejamento e capacidade de tomar decisões importantes que impactarão todo o processo de desenvolvimento. Tendo o desenvolvimento do story board de como seriam as mecânicas, cenários e historia do jogo desenvolvido que foram definidos semanas antes de realmente começar o código.

A fase de criação e implementação envolve a escolha dos assets do jogo, como modelos 2D, sprites, animações, áudios e desenvolvimento de scripts. Aqui, os desafios estão relacionados ao domínio da ferramenta Unity e da linguagem C#, incluindo o conhecimento de programação orientada a objetos, gerenciamento de estados, lógica do jogo e otimização de desempenho. Assim como domínio matemático de saber construir lógica de colisão e posição de personagens como movimentação no eixo x e y do mapa.

Ao enfrentar esses desafios, adquire-se uma compreensão mais profunda dos processos de desenvolvimento de um jogo, aprimorando habilidades de programação, solução de problemas e pensamento lógico, envolvendo conhecimento matemático com contexto de geometria espacial. É interessante identificar na prática que mesmo sendo um jogo simples, com mecânica bem estabelecida é um desafio criar um jogo com esses elementos pois envolvem aplicação e desenvolvimento inúmeras técnicas até um jogo ficar pronto e funcional.

Referências

Unity(2023). **Unity**. Disponível em < <https://unity.com/pt/learn>>. Acesso em: 21Maio2023.

TiledsetMapper(2023). **TiledsetMapper** Disponível em < <https://doc.mapeditor.org/en/stable/>>. Acesso em: 21Maio2023.

