

UNIVERSIDADE FEDERAL DE RORAIMA
CENTRO DE CIÊNCIA E TECNOLOGIA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO
DCC703 - COMPUTAÇÃO GRÁFICA (2024.2)
Prof. LUCIANO FERREIRA SILVA

MARCIA GABRIELLE BONIFÁCIO DE OLIVEIRA - 2020011319
PAULO FERREIRA DA SILVA JÚNIOR - 2019034400

Projeto Final

Poquemon Adventure

Introdução

O "Poquemon Adventure" é um jogo topdown simples desenvolvido na Godot Engine, que integra técnicas de computação gráfica e animação. Este projeto foi concebido com o objetivo de demonstrar, de maneira prática, a aplicação de conceitos fundamentais de desenvolvimento de jogos, como a criação e gerenciamento de sprites, uso de tilemaps para construção de cenários e implementação de colisões. Além disso, o jogo explora a organização de código através de máquinas de estados e a aplicação de técnicas para suavização de movimentos.

Roteiro do Jogo

O jogo inicia com uma tela de menu simples, contendo as opções "Play" e "Quit". Ao selecionar "Play", o jogador é transportado para o universo do jogo, onde assume o controle do personagem principal – o Charizard. Durante a partida, o jogador se depara com dois tipos de inimigos:

- **Wurmple:** Um inimigo corpo a corpo que permanece em estado de "idle" até detectar a proximidade do jogador, momento em que se move em sua direção para atacar.
- Octillery: Um inimigo que ataca à distância. Inicialmente, ele patrulha o cenário de forma aleatória, mas, ao detectar o jogador, passa a disparar projéteis.

O cenário do jogo é composto por um mapa horizontal criado com tilemaps, que inclui elementos como árvores, montanhas e estruturas. Algumas dessas estruturas possuem portas com funções de teletransporte, simulando a entrada e saída de ambientes distintos dentro do mesmo mapa.



Detalhes do Jogo

Personagens e Animações

• Charizard:

- Animações: São utilizadas animações para representar os estados de idle, walk, attack e hurt.
- Máquina de Estados: A gestão dessas animações se dá por meio de uma máquina de estados, que organiza as transições com base na direção (cima, baixo, esquerda, direita e diagonais) e nas ações realizadas, simplificando a estrutura do código.

• Inimigos:

Wurmple:

■ Comportamento: Permanece em estado de "idle" até que o jogador entre em seu raio de detecção. Assim que isso ocorre, o inimigo se move para atacar, utilizando as animações de walk, attack e hurt.

Octillery:

Comportamento: Inicialmente, patrulha o cenário aleatoriamente. Quando o jogador é detectado, o inimigo passa a disparar projéteis à distância.

Cenários e Mecânicas de Jogo

• Tilemaps e Elementos Visuais:

O cenário é construído utilizando tilemaps, permitindo a criação modular e reutilizável dos elementos gráficos como árvores, montanhas e estruturas.

• Sistema de Colisões:

Para a definição de colisões, foi utilizada a ferramenta do Godot que possibilita o desenho de polígonos sobre os elementos do cenário e objetos. Essa técnica garante que os personagens não atravessem barreiras físicas, como árvores e estruturas.

Teletransporte:

Algumas estruturas possuem portas que atuam como pontos de teletransporte, permitindo a transição do jogador entre áreas do mapa e aumentando a dinâmica de navegação.

Interface e Feedback

Menu:

O design da interface é simples e intuitivo, facilitando a navegação. O menu inicial apresenta claramente as opções "Play" e "Quit".



• Feedback Visual das Ações:

As transições entre os estados das animações (por exemplo, quando o personagem passa de *idle* para *attack*) fornecem um feedback visual claro, enriquecendo a experiência do jogador e contribuindo para uma jogabilidade mais imersiva.



Áudio e Trilha Sonora

Menu:

Uma trilha sonora é utilizada na tela de menu, contribuindo para criar a atmosfera inicial e convidar o jogador a iniciar a partida.

• Gameplay:

Durante o jogo, uma música de fundo é empregada para enriquecer a experiência, ajustando o ritmo da jogabilidade e reforçando o ambiente do jogo.

Ferramentas usadas

Godot Engine:

A plataforma central para o desenvolvimento do jogo, responsável pelo gerenciamento de cenas, aplicação de física, execução de animações e controle geral da lógica do jogo.

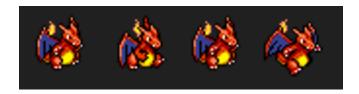
GitHub Desktop:

Utilizado para realizar os commits e gerenciar o versionamento do projeto, o GitHub facilitou a organização do desenvolvimento e a colaboração da equipe, permitindo um controle eficiente das alterações realizadas durante o desenvolvimento do jogo.

Principais técnicas de Computação Gráfica aplicadas

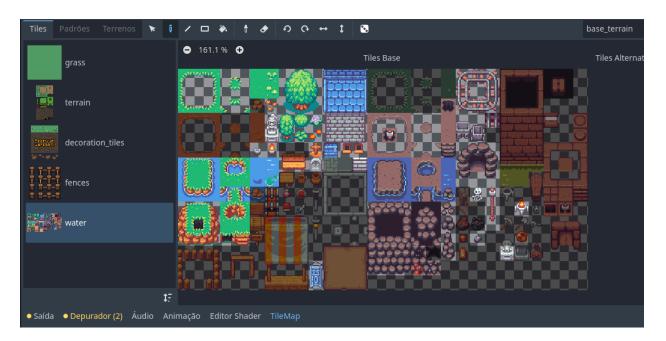
Animação Sprite-Based:

O uso de sprites para animar os personagens, com transições suaves entre os estados de *idle*, *walk*, *attack* e *hurt*, é fundamental para jogos 2D e garante uma experiência visual fluida. Cada personagem foi desenvolvido a partir de sprites individuais, com animações desenhadas para representar diferentes ações e estados.



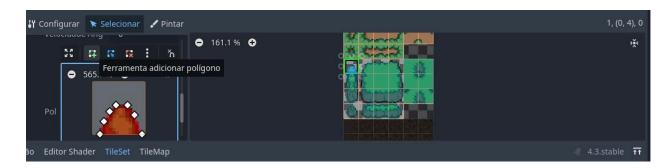
Tilemaps para Construção de Cenários:

A aplicação de "tilemaps" possibilita a criação de cenários ricos e variados, otimizando o processo de desenvolvimento ao permitir a reutilização de elementos gráficos.



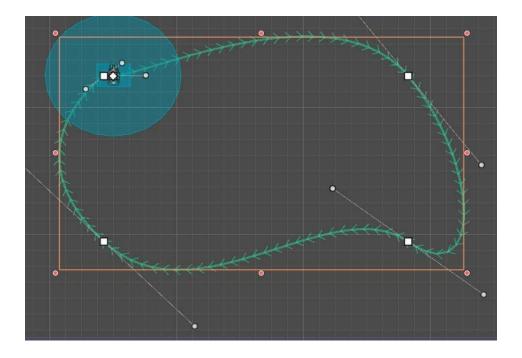
Detecção de Colisões com Polígonos:

A técnica de desenhar polígonos para definir colisões é crucial para a simulação da física no jogo, garantindo interações realistas entre os objetos.



Tentativa com Path2D e Curvas Bézier:

Foi feita a tentativa de suavizar a patrulha do Octillery utilizando Path2D e curvas Bezier utilizando técnicas de interpolação avançadas, mesmo que a solução final tenha sido baseada em direções aleatórias.



Conclusões

Resumo do Projeto:

O "Poquemon Adventure" integra de forma eficaz diversas técnicas de computação gráfica e lógica de jogo, resultando em uma experiência interativa que alia animação e colisões. O uso de sprites, tilemaps, sistemas de colisão por polígonos e uma máquina de estados auxiliaram para uma abordagem organizada e técnica para o desenvolvimento do jogo.

. . .