Unip-Universidade Paulista

Ciência da Computação

Erik Carobolante Goulart RA: G7623G7

Ronaldo Cesar Del Papa Bofe RA: N2606G9

Julio Cesar Satim RA: G808JB7

**JAVABOYS**

Relatório de Desenvolvimento do Jogo

São José do Rio Preto

2025

**Desenvolvimento do Jogo *JavaBoys***

O presente projeto tem como objetivo o desenvolvimento de um jogo eletrônico intitulado *JavaBoys*, pertencente ao gênero **Beat’em Up**, estilo popularizado nos anos 90 por títulos como *Final Fight* e *Streets of Rage*, conhecido no Brasil como *Briga de Rua*. O gênero é caracterizado por combates em tempo real contra múltiplos oponentes em ambientes urbanos, com movimentação lateral e vertical em um plano bidimensional, proporcionando ao jogador a experiência de enfrentar ondas de inimigos enquanto avança pelos cenários.

O jogo está sendo desenvolvido com foco em uma experiência estética única, por meio da criação de **animações 2D desenhadas à mão**, o que reforça a identidade visual do projeto e o diferencia de outros títulos independentes contemporâneos. Cada quadro de animação é ilustrado individualmente, o que confere aos personagens e ao ambiente uma aparência artística artesanal, próxima de uma animação tradicional.

O enredo de *JavaBoys* ainda está sendo elaborado, mas a mecânica central do jogo já está bem definida. O jogador controlará um personagem principal com habilidades básicas de movimentação (esquerda, direita, cima e baixo) e combate (soco), enfrentando uma variedade de inimigos. Estão planejadas **duas fases regulares** com níveis de dificuldade progressiva, seguidas por uma **fase final mais complexa**, na qual o jogador enfrentará o chefão do jogo. O sistema de combate será baseado em colisões e física 2D, com a utilização do componente Rigidbody2D da engine Unity, facilitando a movimentação fluida dos personagens e o realismo nos impactos durante as lutas.

O elenco de inimigos é composto por **quatro personagens não jogáveis**, com níveis distintos de dificuldade. Dois deles representarão os inimigos comuns, com baixa resistência e ataques simples. O terceiro inimigo apresentará um desafio intermediário, com maior resistência e maior agressividade. Por fim, o chefão final, o qual será enfrentado na última fase, possuirá comportamentos únicos e movimentos diferenciados, incluindo um **ataque aéreo especial (chute)**, além de apresentar maior complexidade em sua inteligência artificial.

Cada personagem do jogo contará com um conjunto exclusivo de **animações**, incluindo: andar, ataque (soco), reação ao dano, estado parado (idle) e morte. O chefão terá uma animação adicional correspondente ao seu ataque especial, diferenciando-o dos demais inimigos e elevando o desafio da fase final.

O estado atual do desenvolvimento encontra-se em estágio funcional inicial. O personagem principal já foi implementado e está completamente controlável, com movimentação e ações básicas operando corretamente. Um dos inimigos já está programado com uma lógica de perseguição ao jogador baseada em vetores de direção e física, utilizando a propriedade linearVelocity para realizar os deslocamentos de forma natural no ambiente bidimensional. O controle de colisões e a lógica de detecção de distância mínima entre personagens também estão em implementação para evitar sobreposições e comportamentos indesejados durante a perseguição.

Paralelamente à programação, os **cenários já foram idealizados** e estão sendo produzidos com base nos conceitos gráficos estabelecidos para o jogo. A **tela inicial** já se encontra em fase de montagem, com layout planejado, identidade visual aplicada e elementos como nome do jogo, menu de opções e transições de cena. A **trilha sonora** do jogo também já foi selecionada, com músicas exclusivas para a **tela inicial**, **fases regulares** e para a **batalha contra o chefão**, de forma a criar uma ambientação sonora coerente com cada momento do jogo.

A proposta do projeto *JavaBoys* é unir áreas distintas do desenvolvimento de jogos digitais, como programação, arte gráfica, design de som e narrativa interativa, resultando em um produto final completo e funcional. Além de proporcionar entretenimento, o desenvolvimento do jogo contribui significativamente para o aprofundamento prático dos conhecimentos adquiridos nas áreas de desenvolvimento de software, lógica de programação, controle de fluxo, manipulação de física em ambientes 2D e uso da engine Unity.

Este projeto visa, portanto, a produção de um jogo que não apenas homenageia os clássicos do gênero Beat’em Up, mas que também apresenta uma identidade própria e original, com foco em jogabilidade acessível, visual marcante e progressão desafiadora.

**Script do Personagem do Jogador**

using UnityEngine;

public class ControleDoJogador : MonoBehaviour

{

    [Header("Referencias Gerais")]

    private Rigidbody2D oRigidbody2D;

    private Animator oAnimator;

    [Header("Movimento do Jogador")]

    [SerializeField]private float velocidadeDoJogador;

    private Vector2 inputDeMovimento;

    private Vector2 direcaoDoMovimento;

    [Header ("Controle do Ataque")]

    [SerializeField] private float tempoMaximoEntreAtaques;

    private float tempoAtualEntreAtaques;

    private bool podeAtacar;

    private void Start()

    {

       oRigidbody2D = GetComponent<Rigidbody2D>();

       oAnimator = GetComponent<Animator>();

    }

    private void Update()

    {

        RodarAnimacoesEAtaques();

        ReceberInputs();

        EspelharJogador();

        MovimentarJogador();

        RodarConotrometroDosAtaques();

    }

    private void RodarConotrometroDosAtaques()

    {

        //controla o tempo entre um ataque e outro

        tempoAtualEntreAtaques -= Time.deltaTime;

        if (tempoAtualEntreAtaques <= 0)

        {

            podeAtacar = true;

            tempoAtualEntreAtaques = tempoMaximoEntreAtaques;

        }

    }

    private void ReceberInputs()

    {

        //Amazena a direção que o Jogador quer seguir

        inputDeMovimento = new Vector2(Input.GetAxisRaw("Horizontal"), Input.GetAxisRaw("Vertical"));

    }

private void RodarAnimacoesEAtaques()

{

    //Rodam as animacoes de parado e andando

    if (inputDeMovimento.magnitude == 0)

    {

        oAnimator.SetTrigger("parado");

    }

    else if (inputDeMovimento.magnitude != 0)

    {

        oAnimator.SetTrigger("andando");

    }

    //Roda as animacoes de soco

    if (Input.GetKeyDown(KeyCode.J) && podeAtacar)

    {

        oAnimator.SetTrigger("socando");

        podeAtacar = false;

    }

}

private void EspelharJogador()

{

    // Faz o Jogador olhar para a direção correta

    if (inputDeMovimento.x == 1)

    {

        transform.localScale = new Vector3(1f, 1f, 1f);

    }

    else if (inputDeMovimento.x == -1)

    {

        transform.localScale = new Vector3(-1f, 1f, 1f);

    }

}

    private void MovimentarJogador()

    {

        // Movimenta o Jogador com basse na sua direção

        direcaoDoMovimento = inputDeMovimento.normalized;

        oRigidbody2D.linearVelocity = direcaoDoMovimento \* velocidadeDoJogador;

    }

}

**Script do Personagem do Inimigo**

using UnityEngine;

public class EnemyControl : MonoBehaviour

{

    [Header ("Refencias Gerais")]

    private Rigidbody2D oRigidbody2D;

    private Animator oAnimator;

    private GameObject oJogador;

    [Header ("Movimento do Inimigo")]

    [SerializeField]private float velocidadeDoInimigo;

    private Vector2 direcaoDoMovimento;

    [Header ("Ataque do Inimigo")]

    [SerializeField] private float tempoMaximoEntreAtaques;

    private float tempoAtualEntreAtaques;

    private bool podeAtacar;

    [SerializeField] private float distanciaParaAtacar;

    [SerializeField] private int quantidadeDeAtaquesDoInimigo;

    private int ataqueAtualDoInimigo;

    private void Start()

    {

        oRigidbody2D = GetComponent<Rigidbody2D>();

        oAnimator = GetComponent<Animator>();

        oJogador = FindAnyObjectByType<ControleDoJogador>().gameObject;

    }

    private void Update()

    {

        RodarCronometroDosAtaques();

        SeguirJogador();

        EspelharInimigo();

    }

    private void RodarCronometroDosAtaques()

    {

        tempoAtualEntreAtaques -= Time.deltaTime;

        if(tempoAtualEntreAtaques <= 0)

        {

            podeAtacar = true;

            tempoAtualEntreAtaques = tempoMaximoEntreAtaques;

        }

    }

    private void EspelharInimigo()

    {

            // Faz o Inimigo olhar para a direção correta

        if (oJogador.transform.position.x > transform.position.x)

        {

            transform.localScale = new Vector3(1f, 1f, 1f);

        }

        else if (oJogador.transform.position.x < transform.position.x)

        {

            transform.localScale = new Vector3(-1f, 1f, 1f);

        }

    }

    private void SeguirJogador()

    {

        //Armazena a posicao do Jogador e movimenta a ele

        if(Vector2.Distance(transform.position, oJogador.transform.position) > distanciaParaAtacar)

        {

            direcaoDoMovimento = (oJogador.transform.position - transform.position).normalized;

            oRigidbody2D.linearVelocity = direcaoDoMovimento \* velocidadeDoInimigo;

            oAnimator.SetTrigger("Andando");

        }

        //Deixa de se movimentar e ataca o Jogador

        else

        {

            oRigidbody2D.linearVelocity = Vector2.zero;

            oAnimator.SetTrigger("parado");

            SortearAtaque();

        }

    }

    private void SortearAtaque()

    {

        // Sorteia os ataques e inicia

        ataqueAtualDoInimigo = Random.Range(0, quantidadeDeAtaquesDoInimigo);

        if (podeAtacar)

        {

            IniciarAtaque();

        }

    }

    private void IniciarAtaque()

    {

        // Muda o seu ataque para o ataque sorteado

        if (ataqueAtualDoInimigo == 0)

        {

            oAnimator.SetTrigger("socando");

        }

        else if (ataqueAtualDoInimigo != 0)

        {

            oAnimator.SetTrigger("voadora");

        }

        /\*Testar com o Boss

        switch (ataqueAtualDoInimigo)

        {

            case 0:

            oAnimator.SetTrigger("socando");

            break;

            case 1:

            oAnimator.SetTrigger("voadora");

            break;

        }\*/

        podeAtacar = false;

    }

}