Curso Tecnológico em Jogos Digitais

Prática em Laboratório UNITY
Flappy Bird Player

Murilo Boratto, Leandro Correia e Fred Adler

1 Resumo

A prática em laboratório a seguir tem como finalidade criar um simples jogo animando personagens utilizando recursos do UNITY. A estrutura consiste em uma estrutura 2D chamada FLAPPY BIRD formada por sprites e personagens definidos.

Parte 1 - Animação da imagem de fundo

1. Abrimos o UNITY. Criamos um novo projeto. Colocaremos como nome do projeto, "ProjetoAnimaSprites". Selecionaremos como local a pasta "ProjetosUNITY/NomeAluno" que criamos no "Desktop". Selecionaremos o modo 2D.

2. Baixamos o arquivo compactado do endereço web: Download [1]. Após baixarmos os arquivos e descompactarmos estamos prontos para criar a cena do nosso jogo, basta arrastar os objetos para o painel Project na pasta Assets.

3. Com os elementos copiados para o UNITY comece arrastando e soltando o fundo chamado denominado "cena" para o Hierarchy. Ele deverá aparecer no painel Scene. Como o painel Scene está definido para mostrar uma exibição 2D, você observará que selecionando a Main Camera na Hierarchy mostrará um preview do que a câmera vai para exibir. Você também pode ver isso na exibição do jogo.

4. Salvaremos a Cena em "File > Save scene as" ... Na pasta "Assets" salvaremos a cena com o nome de "Principal".

5. Arrastamos e soltando também os elementos "bgGame@2x.png" para o painel Hierarchy. No painel Inspector teremos que reodernar a ordem de aparição dos elementos dispostos. Na propriedade Order in Layer colocamos para a cena o valor -1 e mudamos o nome do elemento para "imagemFundoA".

6. Como queremos dar a sensação de que a cena passa pelo jogador daremos ao fundo um efeito de corpo rígido, de tal forma que teremos que colocamos o estatus Kinematic para que ele não sofra a física. Sendo, assim precisamos adicionar um script chamado "MoveFundo" para dar esta sensação.

```
using UnityEngine;
using System.Collections;

public class MoveFundo : MonoBehaviour {

  float larguraTela;

  void Start(){

    SpriteRenderer grafico = GetComponent<SpriteRenderer>();

    float larguraImagem = grafico.sprite.bounds.size.x;
    float alturaImagem  = grafico.sprite.bounds.size.y;

    float alturaTela = Camera.main.orthographicSize * 2f;
    larguraTela = alturaTela / Screen.height * Screen.width;

    Vector2 novaEscala = transform.localScale;
    novaEscala.x = larguraTela / larguraImagem + 0.25f;
    novaEscala.y = alturaTela / alturaImagem;
    this.transform.localScale = novaEscala;

    if (this.name == "imagemFundoB")
      transform.position = new Vector2 (larguraTela, 0f);


     GetComponent<Rigidbody2D>().velocity = new Vector2(-3,0);

  }

  void Update(){

     if(transform.position.x <= -larguraTela){
        transform.position = new Vector2(larguraTela, 0f);
     }

  }

}
```

7. Associaremos este script chamado "MoveFundo" ao elemento "imagemFundoA". Sendo asociado ao elemento pelo Inspector ou simplesmente arrastando-o. A partir do código anterior podemos fazer algumas perguntas:

   Como o algoritmo faz a animação da imagem de fundo, isto é, move o elemento imagemFundoA da direita para a esquerda?

   Explique como se baseia o cálculo para que se possa trabalhar com múltiplas resoluções de tela

Parte 2 - Adicionando o personagem e a física

1. Na pasta com o personagem animado Felpudo selecionamos todos os sprites e os arrastamos soltando o no painel Hierarchy. O UNITY pedira para nomearmos o nome elemento como uma animação. O reposicionaremos no início da cena alinhando o personagem na metade da tela.

2. A idéia é aplicarmos ao personagem Felpudo características de corpo rígido e movimentarmos o personagem através de script.

3. Criaremos um script chamado "ControlaJogador" para implementar esta ação. Sendo asociado ao elemento FelpudoIde na hierarquia.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class ControlaJogador : MonoBehaviour {

  bool comecou;
  bool acabou;
  Rigidbody2D corpoJogador;
  Vector2 forcaImpulso = new Vector2(0, 500f);

  void Start () {
    corpoJogador = GetComponent<Rigidbody2D> ();
  }

  void Update () {

    if (Input.GetButtonDown ("Fire1")) {

      if (!comecou) {
        comecou = true;
        corpoJogador.isKinematic = false;
      }

      corpoJogador.velocity = new Vector2 (0, 0);
      corpoJogador.AddForce(forcaImpulso);
    }

  }
}
```

4. Associaremos este script chamado "ControlaJogador" ao elemento "FelpudoIde". Sendo asociado ao elemento pelo Inspector ou simplesmente arrastando-o. A partir do código anterior podemos fazer algumas perguntas:

   Como o algoritmo faz a associação de movimento do personagem com o recurso do mouse?

   Explique como fazemos poderíamos limitar o personagem enquadrado na cena?

Parte 3 - Inserindo o inimigo

1. Na pasta com o personagem animado Lesmo selecionamos todos os sprites e os arrastamos soltando o no painel Hierarchy. O UNITY pedira para nomearmos o nome elemento como uma animação.

2. Criamos um Prefab do elemento animação selecionado e o renomeamos de inimigoPrefab.

3. A seguir associamos o seguinte script a animação Prefab constituída. A idéia do script CriaInimigos.cs é criar inimigos de forma randômica que se deslocam da direita para a esquerda da tela. O comando Instantiate instancia os elementos Prefabs em coordenadas aleatórias da tela.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class CriaInimigos : MonoBehaviour
{
    public GameObject inimigoPrefab;

    public float intialDelay;
    public float enemyPeriod;
    public float enemyHeight;


    void Start()
    {
        InvokeRepeating("CreateEnemy",intialDelay,enemyPeriod);
    }

    void CreateEnemy()
    {
        float a = Random.Range(-enemyHeight,enemyHeight);
        var enemyTransform = Instantiate(inimigoPrefab,this.gameObject.transform).transform;
        enemyTransform.position += a * Vector3.up;
    }

}
```

Referências

[1] UNITY: Flappy Bird Player,
http://www.muriloboratto.docentes.uneb.br/arquivos/pj/docs/comp/projeto-unity-flappy-bird.zip