RELATÓRIO

GRUPO 8: hackerpeople

-Rui Carvalho nº13235

-Andreia Mendes nº13212

-Pedro Rua nº13232

ÍNDICE

Contents

[RELATÓRIO 1](#_Toc478940496)

[GRUPO 8: hackerpeople 1](#_Toc478940497)

[ÍNDICE 2](#_Toc478940498)

[INTRODUÇÃO 3](#_Toc478940499)

[DESCRIÇÃO DAS CLASSES 4](#_Toc478940500)

[Classe "Game 1": 4](#_Toc478940501)

[Classe "Game state": 4](#_Toc478940502)

[Esta classe inicializa e define os elementos do jogo como o rocket, as redes ,os insetos, e contêm o código para as funções para os fazer funcionar como intendido. 4](#_Toc478940503)

[Classe "Input Wrapper": 4](#_Toc478940504)

[Classe "FontSupport": 4](#_Toc478940505)

[Classe "TexturedPrimitive": 4](#_Toc478940506)

[Classe "Camera": 4](#_Toc478940507)

[Classe "Show Vector": 4](#_Toc478940508)

[OBSERVAÇÕES 5](#_Toc478940509)

[CONCLUSÃO 6](#_Toc478940510)

INTRODUÇÃO

Este trabalho consiste num simples jogo bidimensional criado em monogame.

Neste jogo o jogador controla um foguetão com as “arrow keys” e o objectivo é apanhar as abelhas antes que estas desapareçam.

Para realizar a tarefa de programar o jogo foram usadas ferramentas como o:

-IDE:Visual Studio 2015

-Game SDK:Monogame 3.5



DESCRIÇÃO DAS CLASSES

Classe "Game 1":

A classe "MyGame" serve para inicializar as bases do jogo como a *spritebatch* , definir a câmara e a janela, definir o *background* e abrir a janela de jogo.

Classe "Game state":

Esta classe inicializa e define os elementos do jogo como o *rocket,* as redes ,os insetos, e contêm o código para as funções para os fazer funcionar como intendido.

Classe "Input Wrapper":

Esta classe serve para reconhecer o input de certas entradas e executar o que esse input é suposto fazer no programa.

Classe "FontSupport":

A classe "FontSupport" serve para customizar todas as fontes usadas no programa, como a cor ou o tipo de fonte, e o que escrever no ecrã.

Classe "TexturedPrimitive":

A classe "TexturedPrimitive" contêm as funções para gerir todas as *sprites* usadas no jogo, como onde as desenhar, com que tamanho as desenhar, etc.

Classe "Camera":

A classe "Camera" serve para definir os tamanhos e posições da maioria do que aparece no ecra, assim como definir o próprio ecrã, e delimita o limite de colisões para o jogo.

Classe "Show Vector":

Esta classe serve para definir os diferentes vectores usados como o vector de rotação e direção na nave e os vectores da rede.

OBSERVAÇÕES

No decorrer do trabalho encontramos algumas dificuldades que passamos a resolver por exemplo:

-A classe "Camera" não tinha qualquer função "RandomPosition" então foi preciso criar uma que devolvesse posições aleatórias:

static public Vector2 RandomPosition()

{

Random random = new Random();

float x = random.Next(1, (int)sWidth-1);

float y = random.Next(1, (int)sHeight-1);

return new Vector2(x,y);

}

Esta função cria valores aleatórios para as coordenadas x e y no limite da janela e retorna-as como um vector.

-Ao escrever as variáveis em "Camera" não existia qualquer variável para definir a altura da janela que seria suposto usar em algumas funções, pelo que criamos uma variável "sHeight" inicialmente demos os valor de 0f mas como deu alguns problemas alteramos para 100f.

-Na construção do nosso relatório foi preciso pesquisar como fazer um índice do word, visto que alguns dos membros não sabiam sequer da existência de tal mecânica.

CONCLUS**ÃO**

Apesar de termos enfrentado alguns problemas e dificuldades no final fomos capazes de ultrapassa-las e o produto final ficou como tinhamos planeado.

Este trabalho ajudou-nos a compreender melhor monogame e game-design assim como melhorou as nossas capacidades de programação em C#.

