09/11/2023

Projeto Final
The Village

PEDRO CYRNE
LUCAS FERREIRA
GABRIEL FELIPPE
URIEL RELVAS
CELSO VINICIUS

PROFESSOR:
LUIZ FERNANDO

SUMÁRIO

ANAMNESE

Parte gráfica:

O jogo é 2D, ou seja, duas dimensões X e Y (cima e baixo respectivamente).
Devido ao jogo ser 2D o gráfico dele terá base em “sprites” que são, basicamente, imagens em png.

Jogabilidade:

Não terá nenhum tipo de combate direto. O jogo teria como base uma exploração do mundo.

História:

A história é linear, segue uma sequência de acontecimentos e interações simples.

CASO DE USO
Descritivo

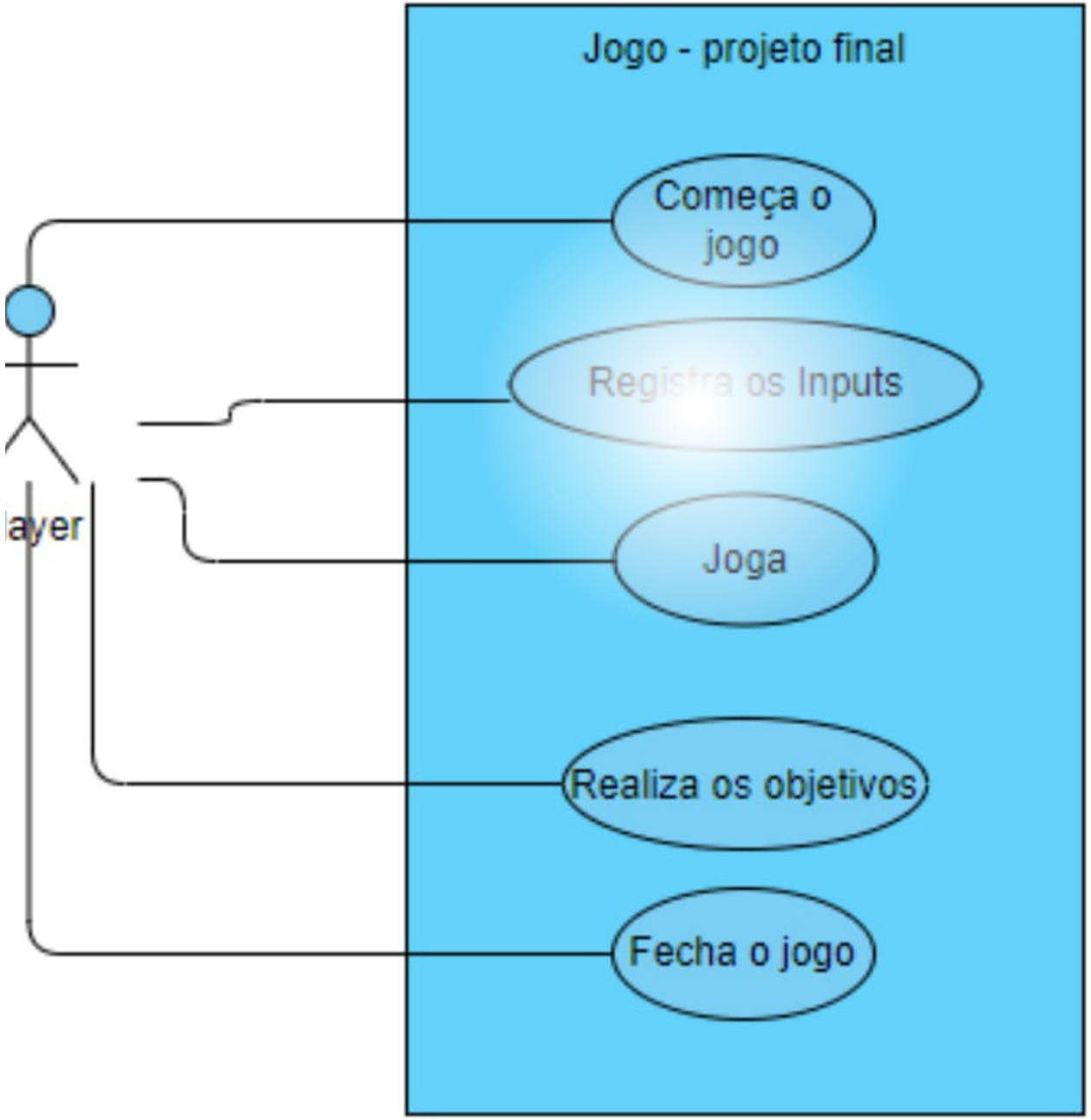
O jogador abre o jogo em seu computador, que logo em seguida, entrará no menu principal. Onde será apresentado algumas opções, como Jogar e Sair. O botão sair encerra o processo do jogo, fechando-o
Já o botão jogar o levará para um nível onde será capaz de começar o jogo em si.

A Game Engine, Unity, é responsável não só pela maior parte dos cálculos mas também pelo registro de "inputs" do usuário. A movimentação é restrita nos eixos X e Y. A mecânica de "vault", que é o que permite o jogador escalar/se pendurar em certos objetos, interagir com objetos, também irá fazer parte da jogabilidade.

Também é possível salvar o progresso, como a última localização que o jogador estava e o seu progresso nas “quests” (objetivos).

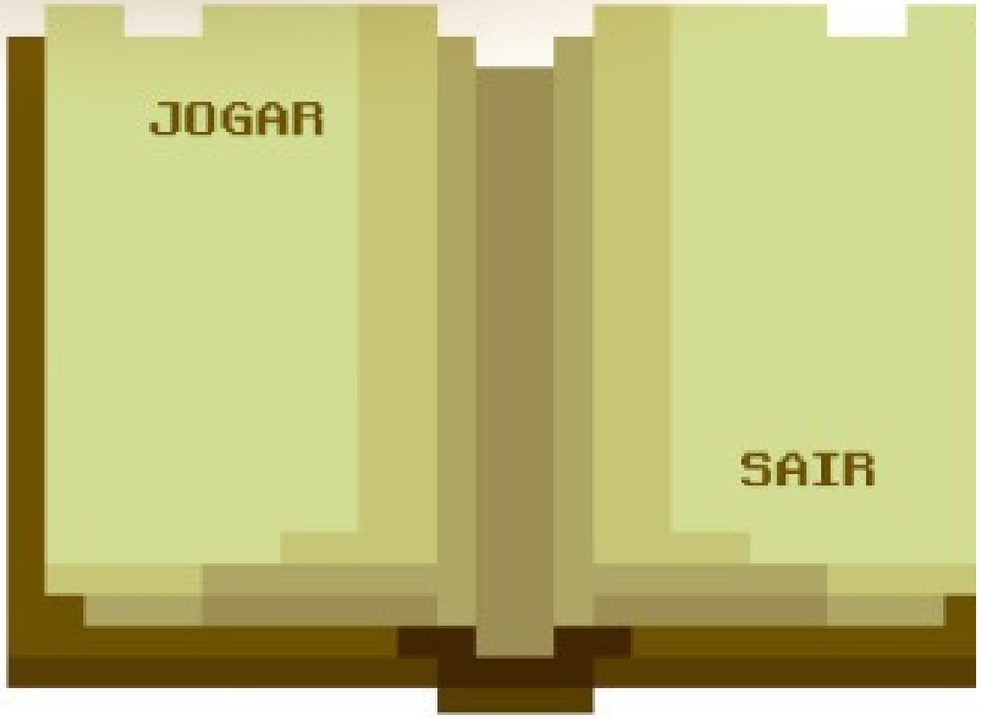
DIAGRAMA DE CASO DE USO:

——————————————->



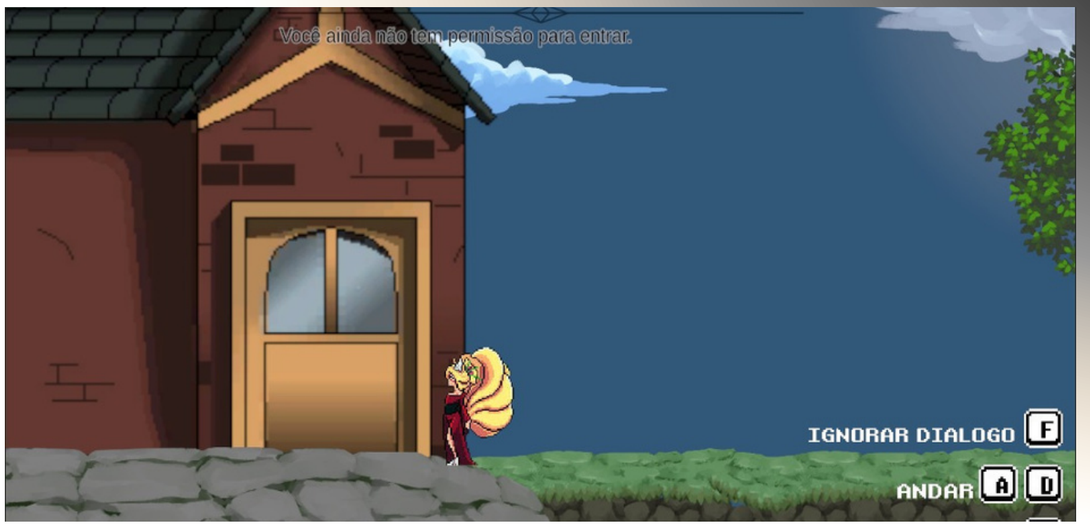
Telas:

Menu Inicial:



Em Jogo:


Poderia pegar uma planta para meus estudos de alquimia?
IGNORAR DIALOGO F
ANDAR A D


Você ainda não tem permissão para entrar.
IGNORAR DIALOGO F
ANDAR A D


JOGO PAUSADO
ANDAR A D

Códigos:

Código responsável pela movimentação dos pássaros:

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

public class Bird : MonoBehaviour
{
private float velocidade;
public float limiteDeDestruicao;
public int right;

private void Start(){
velocidade = Random.Range(0.4f, 2.2f);
right = Random.Range(0, 1);
}

void FixedUpdate()
{
if(right == 1){
transform.Translate(Vector3.right * velocidade *
Time.deltaTime);
if (transform.position.x > limiteDeDestruicao)
{
Destroy(gameObject);
}
}else{
transform.Translate(Vector3.left * velocidade *
Time.deltaTime);
if (transform.position.x < limiteDeDestruicao)
{
Destroy(gameObject);
}
}
}
}
```

Código responsável pelo surgimento dos pássaros na fase:

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

public class BirdSpawner : MonoBehaviour
{
[Header("Valores")]
[SerializeField] private float Tempo_de_Spawn_Min;
[SerializeField] private float Tempo_de_Spawn_Max;

[Header("Componentes")]
[SerializeField] private GameObject Passaro;

private int Randomizer;

void Start()
{
InvokeRepeating("Spawn_Passaro",
Random.Range(Tempo_de_Spawn_Min, Tempo_de_Spawn_Max),
Random.Range(Tempo_de_Spawn_Min, Tempo_de_Spawn_Max));
}

private void Spawn_Passaro(){
Vector2 posicao = new Vector2(transform.position.x,
Random.Range(1, 4));
Instantiate(Passaro, posicao, Quaternion.identity);
}
}
```

Código responsável pelos botões da tela inicial.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using TMPro;
using UnityEngine.SceneManagement;

public class Buttons : MonoBehaviour
{

public AudioClip meuSom;
public AudioSource meuAudioSource;

public void Play(){
meuAudioSource.clip = meuSom;
meuAudioSource.Play();
SceneManager.LoadScene("Loading");
}

public void Settings(){
meuAudioSource.clip = meuSom;
meuAudioSource.Play();
SceneManager.LoadScene("Settings");
}

public void Quit(){
meuAudioSource.clip = meuSom;
meuAudioSource.Play();
Application.Quit();
}

public void Next(){
SceneManager.LoadScene(1);
}
}
```

Códigos

Código responsável pela movimentação da câmera:

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

public class CameraMovement : MonoBehaviour
{

private Vector3 Camera_Offset; //Esse é o valor responsavel pelo offset da camera, nesse caso, o quão longe ela vai se manter do player

[Header("Values")]
[SerializeField] private float Camera_Smooth_Multipliyer; //Esse é o valor que sera responsavel pela suavização

[Header("Components")]
[SerializeField] private Transform Camera_Target_Follow; //esse é onde o unity se referira ao transform que a camera ira seguir, ate momento e geralmente o avatar do player

//Others
private Vector3 Camera_Velocity = Vector3.zero;

private void Awake() => Camera_Offset = transform.position - Camera_Target_Follow.position; //Toda vez que a camera for ativada, isso era ativado junto. Calculo do offset basicamente.

private void LateUpdate() //a ultima coisa calculada depois dos updates
{
Callculus();
}

private void Callculus(){
Vector3 Camera_Targeted_Position = Camera_Target_Follow.position + Camera_Offset; //faz os calculos de qual posição a camera deverá ou deveria estar.
transform.position = Vector3.SmoothDamp(transform.position, Camera_Targeted_Position, ref Camera_Velocity, Camera_Smooth_Multipliyer); // Basicamente faz a movimentação da camera pelo mapa depois de todas as contas
}

}
```

Código responsável pela primeira “quest” do level:

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

public class firstquest : MonoBehaviour
{
[Header("Valores")]
[SerializeField] private int QuestID;
[SerializeField] private int QuestPosition_X;
[SerializeField] private int QuestPosition_Y;

[Header("Componentes")]
[SerializeField] private Animator QuestCache;
[SerializeField] private TextMeshProUGUI Name;
[SerializeField] private GameObject DialogBox;
[SerializeField] private TextMeshProUGUI dialogtext;
[SerializeField] private Transform QuestTracker;

[Header("Dialogo")]
[SerializeField] private string[] lines;
[SerializeField] private float textSpeed;
public int index;
private string clear;

private void Start() {
dialogtext.text = string.Empty;
}

private void OnTriggerEnter2D(Collider2D other){
if(QuestCache.GetInteger("QuestStatus") == QuestID){
DialogBox.SetActive(true);
startdialoge();
QuestTracker.position = new Vector2 (QuestPosition_X, QuestPosition_Y);
QuestCache.SetInteger("QuestStatus", QuestID + 1);
}
if(QuestCache.GetInteger("QuestStatus") == 2){
QuestCache.SetInteger("QuestStatus", 3);
}

}

private void OnTriggerStay2D(Collider2D other){
if(Input.GetKeyDown(KeyCode.F)){
DialogSkipper();
}
}

private void OnTriggerExit2D(Collider2D other){
DialogBox.SetActive(false);
}

private void startdialoge(){
dialogtext.text = clear;
index = 0;
StartCoroutine(TypeLine());
}

private void DialogSkipper(){
index += 1;
dialogtext.text = clear;
StartCoroutine(TypeLine());
}

IEnumerator TypeLine(){
foreach (char c in lines[index].ToCharArray()){
dialogtext.text += c;
yield return new WaitForSeconds(textSpeed);
}
}
```

Obrigado!