



Memories Unveiled

CURSO PROFISSIONAL DE GESTÃO
E PROGRAMAÇÃO DE SISTEMAS
INFORMÁTICOS

Eduardo Miguel Felgueiras Marinho

Trabalho realizado sob a orientação de:

Prof. Manuel Ferreira

Escola Profissional de Braga

Braga, 13 de maio de 2024

Nota Introdutória

A Prova de Aptidão Profissional consiste na apresentação e defesa, perante um júri, de um projeto, consubstanciado num produto, material ou intelectual, numa intervenção ou numa atuação, bem como do respetivo relatório final de realização e apreciação crítica, demonstrativo de saberes e competências profissionais, adquiridos ao longo da formação e estruturante do futuro profissional.

A sua realização constitui-se como um dos imperativos legais para a conclusão do curso profissional de (designação do curso), que confere uma dupla certificação: qualificação profissional de nível IV e o 12º ano como certificação escolar de nível secundário.

A presente prova foi realizada na Escola Profissional de Braga no presente ano letivo e a sua defesa realizada, perante um júri final, nas datas estabelecidas no calendário escolar.

Dedicatória

Dedico a minha Prova de Aptidão Profissional a todos que me apoiaram durante todo este tempo.

Eduardo Miguel Felgueiras Marinho

3

Agradecimentos

Gostaria de expressar a minha gratidão a todos que estiveram ao meu lado durante esta jornada que está longe de acabar da criação do meu jogo. Estar a realizar este projeto significa muito para mim e o apoio de todos foi uma coisa que me ajudou muito.

Aos meus amigos, obrigado por me ouvirem enquanto eu compartilhava as minhas ideias e por oferecerem feedback verdadeiro que me ajudou a melhorar cada etapa que já passou.

À minha família quero agradecer por todo o apoio incondicional e encorajamento ao longo desta jornada.

Aos meus professores agradeço por compartilharem todos os conhecimentos e por me guiarem neste projeto.

Índice

Introdução	11
Objetivos	12
Como surgiu a ideia	12
Inspirações	13
1. Recursos Utilizados	15
Programas Utilizados	15
1.1.1 Unity	15
1.1.2 Visual Studio	16
1.1.3 Blender	16
1.1.4 Notepad++	17
1.1.5 GitHub	17
1.1.6 Amps	18
1.1.7 phpMyAdmin	18
Linguagens utilizadas	19
1.1.8 C#	19
1.1.9 HTML	19
1.1.10 PHP	20
1.1.11 CSS	20
2. Jogo	21
Mapa	21
Jogador	24

Eduardo Miguel Felgueiras Marinho

5

Inventário.....	27
Item	32
Interação “Isqueiro/Paus”	34
Interação “Chave/Porta”	36
Botões.....	38
Luz Código Morse	39
Porta com código.....	41
Botão de saída	44
3. Site	45
Home.....	45
Download	47
Login	47
Registo	48
Base de dados.....	49
4. Alterações Futuras.....	51
Conclusão	52
Bibliografia	53
Anexos	54

Índice de Figuras

1	Imagens de inspiração "When the darkness comes"/"We were here".....	13
2	Unity.....	15
3	Visual Studio	16
4	Blender.....	16
5	Notepad++	17
6	GitHub.....	17
7	Ampps	18
8	phpMyAdmin.....	18
9	C#	19
10	HTML	19
11	PHP.....	20
12	CSS	20
13	Mapa visto de cima.....	21
14	Primeira Sala.....	21
15	Corredor	22
16	Segunda Sala	22
17	Terceira Sala.....	23
18	Script Player Motor.....	24
19	Script Input Manager.....	25
20	Script Player Look.....	26
21	Inventário	27

Eduardo Miguel Felgueiras Marinho

7

22 Item no inventário	27
23 Verifica se o item foi clicado e vai adiciona-lo ao inventário	27
24 Adiciona o item ao inventário	28
25 Verifica se tem o item no inventário	28
26 Toggle Inventory	29
27 Item que larguei.....	29
28 Largar Items.....	30
29 Item noutro slot	30
30 Arrastar item.....	31
31 Item	32
32 Isqueiro	32
33 Exemplo do script "Item"em uso	33
34 Antes e depois da interação	34
35 Chave começa desativada.....	34
36 Verificação de distância.....	35
37 Requisitos para a interação acontecer	35
38 Interação Isqueiro/Paus	35
39 Antes da interação	36
40 Depois da interação.....	37
41 Modificações do código.....	37
42 Editor de animações	37
43 Botões.....	38

44 Interruptor e lâmpada	39
45 Começo da transmissão	39
46 Mensagem em código morse	40
47 Começo e fim da transmissão	40
48 Transmissão	40
49 Code Lock	41
50 Code Lock	41
51 Porta aberta	42
52 Verifica se o jogador está na área	42
53 Adicionar dígito	42
54 Estado do painel.....	43
55 Botão final	44
56 Botão sair.....	44
57 Coleção de imagens.....	45
58 Carrousel de imagens.....	46
59 Sobre o jogo	46
60 Botão download	47
61 Login	47
62 Registo.....	48
63 Tabelas	49
64 Tabela avaliar	49
65 Tabela recolha_dados	50

66 Tabela utilizador	50
----------------------------	----

10 Eduardo Miguel Felgueiras Marinho



Cofinanciado pela
União Europeia



uma escola

Rumos
education

Knowledge
sharing

Introdução

O jogo "Memories Unveiled" é um projeto que surge da minha paixão pelos videojogos e da minha busca por combinar narrativa emocional, exploração e quebra-cabeças em um ambiente não linear. Este jogo é criado como parte da minha prova de aptidão profissional, demonstrando as minhas habilidades na utilização de "Unity" e das linguagens de programação C# e C++.

O objetivo principal de "Memories Unveiled" é proporcionar uma experiência envolvente e emocional para os jogadores. Eles devem explorar ambientes misteriosos, resolver quebra-cabeças desafiadores e, ao fazê-lo, desvendar a história por trás das memórias perdidas do personagem principal.

O jogo "Memories Unveiled" será desenvolvido em "Unity", aproveitando as capacidades dessa engine de desenvolvimento de jogos. As linguagens de programação C# e C++ serão utilizadas para implementar mecânicas de jogo, controlos e sistemas interativos.

"Memories Unveiled" inova ao combinar elementos de exploração, quebra-cabeças e narrativa não linear em um único jogo. Um dos desafios será criar quebra-cabeças criativos e integrá-los perfeitamente na história, mantendo o equilíbrio entre o desafio e a diversão.

O jogo terá um estilo artístico único, com gráficos que criam uma atmosfera misteriosa e cativante. A paleta de cores será usada para refletir as emoções da narrativa, variando de tons sombrios a vibrantes à medida que a história se desenrola.

Objetivos

Os meus objetivos são:

- Promover o conceito de Game Development;
- Desenvolver competências no desenvolvimento de jogos para a plataforma Unity;
- Aumentar a capacidade e domínio da linguagem C# e C++;
- Aplicar conhecimentos e competências adquiridos nas diferentes disciplinas;
- Aumentar a minha capacidade de domínio sobre este tema;

Como surgiu a ideia

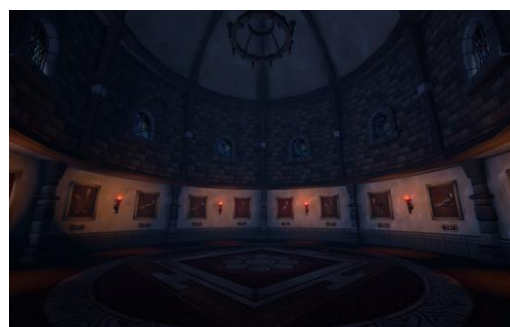
Eu tive esta ideia porque sempre gostei muito de vídeo jogos, e uma das grandes razões para eu ter ingressado neste curso foi o meu gosto por vídeo jogos.

Inspirações

Eu tive como inspirações dois jogos “When the darkness comes” e “We were here”, estes dois jogos foram o porquê de eu querer fazer um jogo.

Os visuais e a história de “When the darkness comes” são o que traz prazer a jogar, não é a sua jogabilidade que o torna um jogo completamente diferente dos outros, são os sentimentos e mensagem transmitida nesse jogo que o tornam um dos melhores jogos que eu já joguei.

Já “We were here” é a inspiração que mais se nota no meu jogo, os puzzles e mecânicas revolucionárias fazem com que tenha sido o melhor jogo desse género que eu já joguei e eu tentei utilizar mecânicas que não tenham sido tão usadas como a implementação de código morse para decifrar um código.



1 Imagens de inspiração “When the darkness comes”/“We were here”

Cronograma		
Fase	Calendarização	Elenco de atividades nas diversas fases
Conceção	Setembro 2023	Conhecimento do regulamento
	Setembro 2023	Definição dos critérios da PAP pela direção
	Setembro 2023	Período de negociação / reflexão / decisão sobre os temas / problemas a explorar
	Setembro 2023	Normas para a elaboração de um trabalho científico
	Até 31 outubro 2023	Entrega do projeto de acordo com as normas do regulamento
Desenvolvimento	Setembro 2023	Idealização da PAP
	Outubro 2023	Criação da história, personagens e cenários.
	Novembro/Dezembro 2023	Início do projeto
	Janeiro/Fevereiro 2024	Testagem do jogo
	Março/Abril 2024	Conclusão do projeto
	Maio 2024	Ajustes finais
Autoavaliação e Relatório	Janeiro 2024	Defesa intermédia da PAP
	13 maio 2024	Conclusão do projeto e entrega do relatório
	Julho 2024	Defesa final da PAP

1. Recursos Utilizados

Durante o desenvolvimento da minha Prova de Aptidão Profissional eu utilizei variados recursos, desde engines de desenvolvimento de jogos, modeladores 3D, editores de texto , linguagens de programação, entre outros.

Programas Utilizados

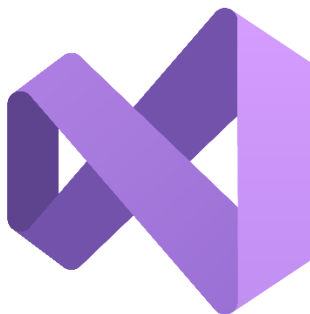
1.1.1 Unity



2 Unity

O Unity é uma plataforma de desenvolvimento de jogos, oferece diversas ferramentas para criar experiências interativas em 2D e 3D. Com suporte a múltiplas plataformas, gráficos avançados e integração de linguagens como o C#, eu utilizei o Unity porque utiliza C# que é uma linguagem que eu já conhecia.

1.1.2 Visual Studio



3 Visual Studio

O Visual Studio é um IDE versátil desenvolvido pela Microsoft, usado para programar em várias linguagens como C#, C++, Python e JavaScript. Oferece recursos avançados de edição de código e integração com outras ferramentas de desenvolvimento.

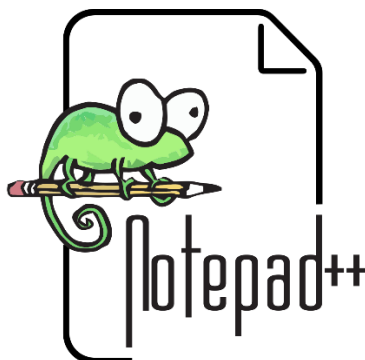
1.1.3 Blender



4 Blender

O Blender é um software usado para modelagem, animação e renderização 3D. Oferece uma ampla gama de ferramentas para criar gráficos 3D, incluindo modelagem de objetos, simulação, texturização e composição, eu escolhi o Blender porque foi a ferramenta que eu mais ouvi falar para modelação 3D.

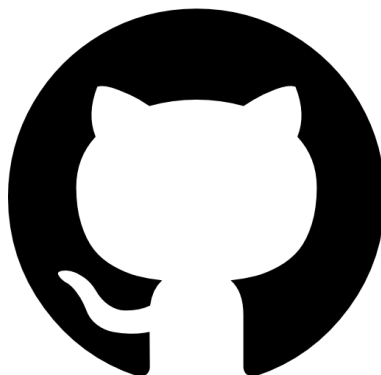
1.1.4 Notepad++



5 Notepad++

O Notepad++ é um editor de texto, ideal para programação e edição de arquivos de texto, eu utilizei o Notepad++ para o desenvolvimento do meu website por ter utilizado durante as aulas de desenvolvimento web.

1.1.5 GitHub



6 GitHub

O GitHub é uma plataforma de hospedagem de código, permite controlar a versão e gerir problemas, facilitando o desenvolvimento e compartilhamento do código.

1.1.6 Ampps



7 Ampps

O Ampps facilita a criação gestão de ambientes de desenvolvimento web local. Ele inclui um pacote integrado que contém o Apache, MySQL, PHP, Perl e outras ferramentas essenciais para o desenvolvimento web.

1.1.7 phpMyAdmin

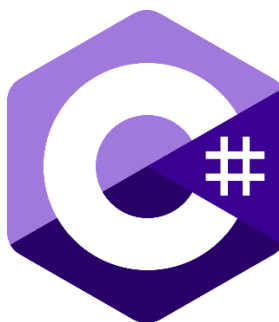


8 phpMyAdmin

phpMyAdmin é uma aplicação web utilizada para gerir bases de dados MySQL através de uma interface gráfica. Permite aos utilizadores criar, modificar e eliminar bases de dados, tabelas, campos e registos diretamente no navegador web.

Linguagens utilizadas

1.1.8 C#



9 C#

C# é uma linguagem de programação moderna e versátil. É utilizada para o desenvolvimento de software, é conhecida pela sua robustez.

1.1.9 HTML



10 HTML

HTML é a linguagem base utilizada para criar e estruturar páginas web, permite definir a estrutura e o conteúdo de uma página, incluindo texto, imagens, links e outros elementos multimédia, é a base fundamental para o desenvolvimento web.

1.1.10 PHP



11 PHP

PHP é uma linguagem de programação utilizada para o desenvolvimento de aplicações web dinâmicas, é especialmente conhecida por ser integrada com HTML e é capaz de se ligar a bases de dados.

1.1.11 CSS



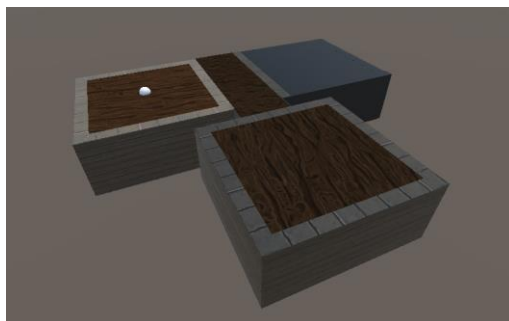
12 CSS

CSS é uma linguagem utilizada para estilizar e formatar elementos HTML em páginas web. Com o CSS, é possível controlar a aparência dos elementos, incluindo cores, fontes, espaçamentos, layout e outros estilos.

2. Jogo

Mapa

O mapa consiste de três salas e um corredor.



13 Mapa visto de cima

A sala principal onde o jogador “acorda” que tem um piano, uma lareira, um isqueiro em cima de uma mesa, e uma porta, nessa sala o jogador tem de pegar no isqueiro, dirigir-se à lareira e fazer uma interação, que vai fazer os paus da lareira “arderem” e aparece uma chave que faz com que a porta abra.



14 Primeira Sala

No corredor tem três portas, uma aberta e duas fechadas e oito botões, uma das portas fechadas não vai a lado nenhum porque é meio que uma distração assim como os botões que apenas se ligam com luzes de várias cores, a outra porta fechada vai dar à última sala, mas para a abrir é preciso um código que vai ser obtido na segunda sala que é a sala de porta aberta.



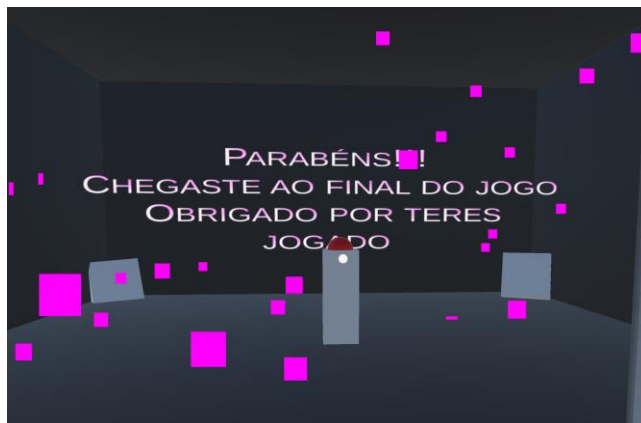
15 Corredor

Eu tentei fazer com que a segunda sala se parecesse com uma pequena biblioteca, tem duas mesas com livros e uma lâmpada em cima, duas cadeiras e três armários com livros, tem também um quadro com uma conta e um interruptor na parede que quando ativado liga a luz que transmite um código que é necessário para decifrar o código para a terceira e última sala.



16 Segunda Sala

A terceira sala é uma sala muito simples em que tem texto escrito pelas paredes, dois blocos em que saiem partículas e um botão em que sai do jogo.



17 Terceira Sala

Jogador

Andar, saltar e olhar são as mecânicas que o jogador mais utilizará.

O jogador consegue andar nas teclas “W” para andar para a frente, “A” para andar para a esquerda, “S” para andar para trás e “D” para andar para a direita.

Consegue saltar no “Espaço” e consegue “olhar” com os movimentos do rato.

```
public class PlayerMotor : MonoBehaviour
{
    private CharacterController controller;
    private Vector3 playerVelocity;
    private bool isGrounded;
    public float speed = 5f;
    public float gravity = -9.8f;
    public float jumpHeight = 1f;

    void Start()
    {
        controller = GetComponent<CharacterController>();
    }

    void Update()
    {
        isGrounded = controller.isGrounded;
    }

    public void ProcessMove(Vector2 input)
    {
        Vector3 moveDirection = Vector3.zero;
        moveDirection.x = input.x;
        moveDirection.z = input.y;
        controller.Move(transform.TransformDirection(moveDirection) * speed * Time.deltaTime);
        playerVelocity.y += gravity * Time.deltaTime;
        if (isGrounded && playerVelocity.y < 0)
            playerVelocity.y = -2f;
        controller.Move(playerVelocity * Time.deltaTime);
        Debug.Log(playerVelocity.y);
    }

    public void Jump()
    {
        if(isGrounded)
        {
            playerVelocity.y = Mathf.Sqrt(jumpHeight * -1.0f * gravity);
        }
    }
}
```

18 Script Player Motor

```
public class InputManager : MonoBehaviour
{
    private PlayerInput playerInput;
    private PlayerInput.OnFootActions onFoot;

    private PlayerMotor motor;
    private PlayerLook look;

    void Awake()
    {
        playerInput = new PlayerInput();
        onFoot = playerInput.OnFoot;

        motor = GetComponent<PlayerMotor>();
        look = GetComponent<PlayerLook>();

        onFoot.Jump.performed += ctx => motor.Jump();
    }

    void FixedUpdate()
    {
        motor.ProcessMove(onFoot.Movement.ReadValue<Vector2>());
    }

    void LateUpdate()
    {
        look.ProcessLook(onFoot.Look.ReadValue<Vector2>());
    }

    private void OnEnable()
    {
        onFoot.Enable();
    }

    private void OnDisable()
    {
        onFoot.Disable();
    }
}
```

19 Script Input Manager

```
public class PlayerLook : MonoBehaviour
{
    public Camera cam;
    private float xRotation = 0f;

    public float xSensitivity = 30f;
    public float ySensitivity = 30f;

    private float originalXSensitivity;
    private float originalYSensitivity;

    public void Start()
    {
        originalXSensitivity = xSensitivity;
        originalYSensitivity = ySensitivity;
    }

    public void ProcessLook(Vector2 input)
    {
        float mouseX = input.x;
        float mouseY = input.y;

        xRotation -= (mouseY * Time.deltaTime) * ySensitivity;
        xRotation = Mathf.Clamp(xRotation, -80f, 80f);

        cam.transform.localRotation = Quaternion.Euler(xRotation, 0, 0);

        transform.Rotate(Vector3.up * (mouseX * Time.deltaTime) * xSensitivity);
    }

    public void SetSensitivity(float sensitivity)
    {
        xSensitivity = sensitivity;
        ySensitivity = sensitivity;
    }

    public void ResetSensitivity()
    {
        xSensitivity = originalXSensitivity;
        ySensitivity = originalYSensitivity;
    }
}
```

20 Script Player Look

Inventário

O inventário pode ser acessado quando o jogador pressionar a tecla “Tab”.



21 Inventário

Quando o jogador clica com o botão esquerdo do rato num item esse item vai ser adicionado ao inventário.



22 Item no inventário

```

88 private void ItemRaycast(bool hasClicked = false)
89 {
90     itemHoverText.text = "";
91     Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);
92
93     RaycastHit hit;
94
95     if (Physics.Raycast(ray, out hit, raycastDistance, itemLayer))
96     {
97         if (hit.collider != null)
98         {
99             Item newItem = hit.collider.GetComponent<Item>();
100             if (newItem)
101             {
102                 if (hasClicked)
103                 {
104                     AddItemToInventory(newItem);
105                 }
106                 else
107                 {
108                     itemHoverText.text = newItem.name;
109                 }
110             }
111         }
112     }
113 }

```

23 Verifica se o item foi clicado e vai adiciona-lo ao inventário

Eduardo Miguel Felgueiras Marinho

27

```

115 private void AddItemToInventory(Item itemToAdd)
116 {
117     int leftOverQuantity = itemToAdd.currentQuantity;
118     Slot openSlot = null;
119
120     for (int i = 0; i < allInventorySlots.Count; i++)
121     {
122         Item heldItem = allInventorySlots[i].getItem();
123
124         if (heldItem != null && itemToAdd.name == heldItem.name)
125         {
126             int freeSpaceInSlot = heldItem.maxQuantity - heldItem.currentQuantity;
127
128             if (freeSpaceInSlot >= leftOverQuantity)
129             {
130                 heldItem.currentQuantity += leftOverQuantity;
131                 Destroy(itemToAdd.gameObject);
132                 return;
133             }
134             else
135             {
136                 heldItem.currentQuantity = heldItem.maxQuantity;
137                 leftOverQuantity -= freeSpaceInSlot;
138             }
139         }
140         else if (heldItem == null)
141         {
142             if (openSlot == null)
143                 openSlot = allInventorySlots[i];
144         }
145     }
146
147     if (leftOverQuantity > 0 && openSlot != null)
148     {
149         openSlot.setItem(itemToAdd);
150         itemToAdd.currentQuantity = leftOverQuantity;
151         itemToAdd.gameObject.SetActive(false);
152     }
153     else
154     {
155         itemToAdd.currentQuantity = leftOverQuantity;
156     }
157 }

```

24 Adiciona o item ao inventário

Existe também uma verificação no inventário para ver se o jogador tem o item.

```

public bool HasItem(GameObject itemPrefab)
{
    string itemName = itemPrefab.name; // Get the name of the item prefab
    foreach (Slot slot in allInventorySlots)
    {
        Item item = slot.getItem();
        if (item != null && item.gameObject.name == itemName)
        {
            return true;
        }
    }
    return false;
}

```

25 Verifica se tem o item no inventário

28 Eduardo Miguel Felgueiras Marinho

Se estiver com o inventário aberto aparece o cursor do rato e a sensibilidade da câmara fica em zero para não mexer, quando fecha o inventário a sensibilidade volta ao normal e o cursor desaparece.

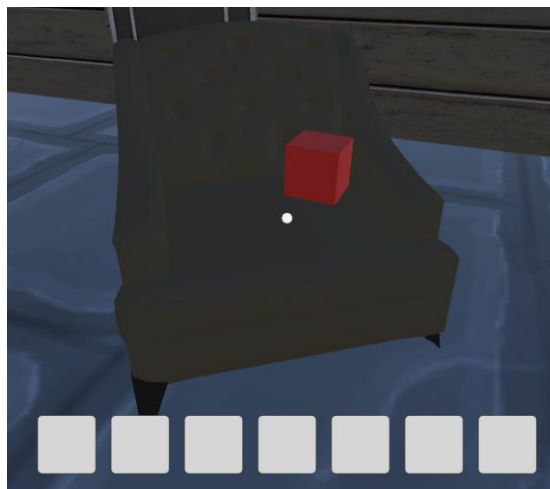
```

173 private void ToggleInventory(bool enable)
174 {
175     inventory.SetActive(enable);
176
177     Cursor.lockState = enable ? CursorLockMode.None : CursorLockMode.Locked;
178     Cursor.visible = enable;
179
180     if (playerLook != null)
181     {
182         if (enable)
183         {
184             Debug.Log("Setting sensitivity to 0");
185             playerLook.SetSensitivity(0f);
186             isSensitivityZero = true;
187         }
188         else if (isSensitivityZero)
189         {
190             Debug.Log("Resetting sensitivity");
191             playerLook.ResetSensitivity();
192             isSensitivityZero = false;
193         }
194     }
195
196     if (!enable)
197     {
198         foreach (Slot curSlot in allinventorySlots)
199             curSlot.hovered = false;
200     }
201 }

```

26 Toggle Inventory

Com o inventário aberto consegue-se largar items quando se clica na tecla “X”.



27 Item que larguei

Eduardo Miguel Felgueiras Marinho

29

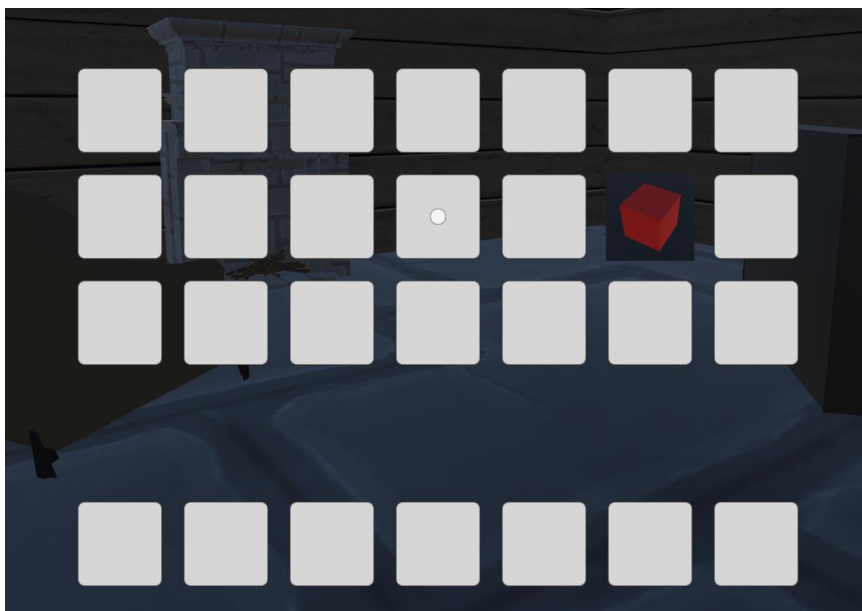
```

204 private void dropItem()
205 {
206     foreach (Slot hotbarSlot in hotbarSlots)
207     {
208         if (hotbarSlot.hovered && hotbarSlot.hasItem())
209         {
210             hotbarSlot.getItem().gameObject.SetActive(true);
211             hotbarSlot.getItem().transform.position = dropLocation.position;
212             hotbarSlot.setItem(null);
213             return;
214         }
215     }
216     foreach (Slot inventorySlot in InventorySlots)
217     {
218         if (inventorySlot.hovered && inventorySlot.hasItem())
219         {
220             inventorySlot.getItem().gameObject.SetActive(true);
221             inventorySlot.getItem().transform.position = dropLocation.position;
222             inventorySlot.setItem(null);
223             return;
224         }
225     }
226 }
227

```

28 Largar Items

Dentro do inventário consegue-se arrastar o icone dos items para outros slots.



29 Item noutra slot

30 Eduardo Miguel Felgueiras Marinho

```

230 private void dragInventoryIcon()
231 {
232     for (int i = 0; i < allInventorySlots.Count; i++)
233     {
234         Slot curSlot = allInventorySlots[i];
235         if (curSlot.hovered && curSlot.hasItem())
236         {
237             currentDragSlotIndex = i;
238
239             currentDraggedItem = curSlot.getItem();
240             dragIconImage.sprite = currentDraggedItem.icon;
241             dragIconImage.color = new Color(1, 1, 1, 1);
242
243             curSlot.setItem(null);
244         }
245     }
246 }

```

30 Arrastar item

Item

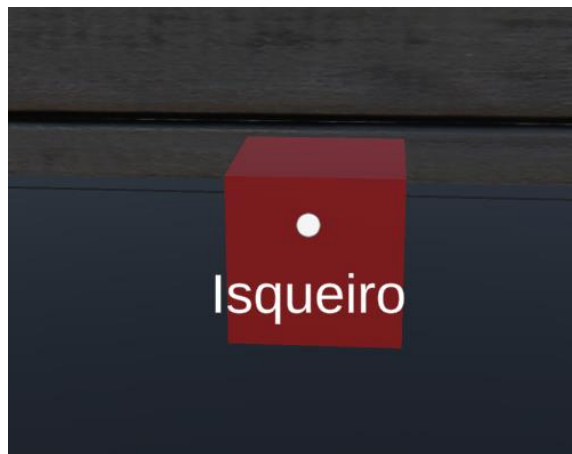
Os items podem ser inseridos no inventário com o clique do botão esquerdo do rato enquanto o jogador “olha” para ele.

```
public class Item : MonoBehaviour
{
    public new string name = "New Item";
    public string description = "New Description";
    public Sprite icon;
    public int currentQuantity = 1;
    public int maxQuantity = 16;

    public int equippableItemIndex = -1;
}
```

31 Item

Ao adicionar o script ao item eu posso modificar o nome, a descrição, o icone, a quantidade atual, a quantidade máxima e o “slot” em que ele vai estar.



32 Isqueiro

Item (Script)	
Script	Item
Name	Isqueiro
Description	Isqueiro
Icon	Isqueiro
Current Quantity	1
Max Quantity	16
Equippable Item Inde	-1

33 Exemplo do script "Item" em uso

Interação “Isqueiro/Paus”

Com o item “Isqueiro” no inventário, o jogador dirige-se à lareira e clicando no “F” acontece a interação que “queima” os paus para aparecer a chave que estava escondida antes.



34 Antes e depois da interação

```
private void Start()
{
    chaveObject = GameObject.FindWithTag("Chave");
    if (chaveObject != null)
    {
        chaveObject.SetActive(false);
    }
}
```

35 Chave começa desativada

Para fazer a interação o jogador tem de estar dentro de um “collider”.

```
private void OnTriggerEnter(Collider other)
{
    if (other.CompareTag("Player"))
    {
        canInteract = true;
    }
}

private void OnTriggerExit(Collider other)
{
    if (other.CompareTag("Player"))
    {
        canInteract = false;
    }
}
```

36 Verificação de distância

Se o jogador estiver com o item no inventário, clicar no “F” e estiver perto o suficiente a interação acontece.

```
private void Update()
{
    if (canInteract && Input.GetKeyDown(interactionKey) && interactionEnabled)
    {
        if (playerInventory.HasItem(requiredItemPrefab))
        {
            PerformInteraction();
        }
    }
}
```

37 Requisitos para a interação acontecer

Ao acontecer a interação os paus são destruídos e a chave aparece.

```
private void PerformInteraction()
{
    GameObject pausObject = GameObject.FindWithTag("Paus");
    if (pausObject != null)
    {
        Destroy(pausObject);
    }

    if (chaveObject != null)
    {
        if (!chaveObject.activeSelf)
        {
            chaveObject.SetActive(true);
        }
    }

    interactionEnabled = false;
}
```

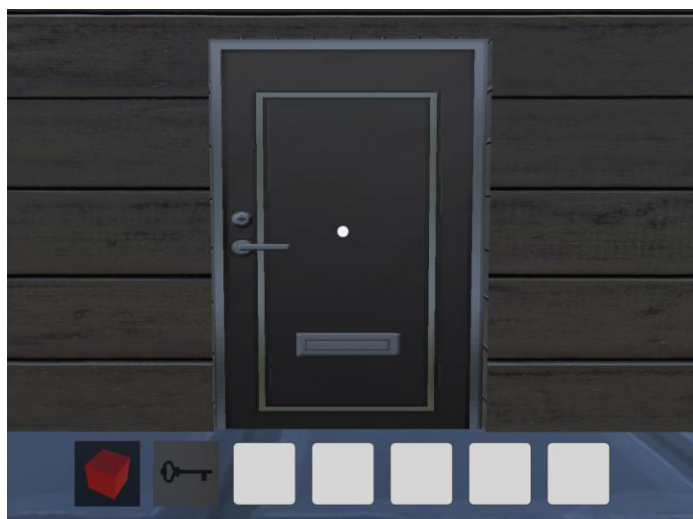
38 Interação Isqueiro/Paus

Interação “Chave/Porta”

Esta interação é muito parecida com a anterior, tem apenas duas diferenças.

A primeira é que em vez de destruir o item ele faz uma animação da porta a abrir, o que fez com que fosse mais difícil, uma vez que tive de criar uma animação para a porta.

E a segunda é que quando a interação acontece não aparece nenhum item novo.



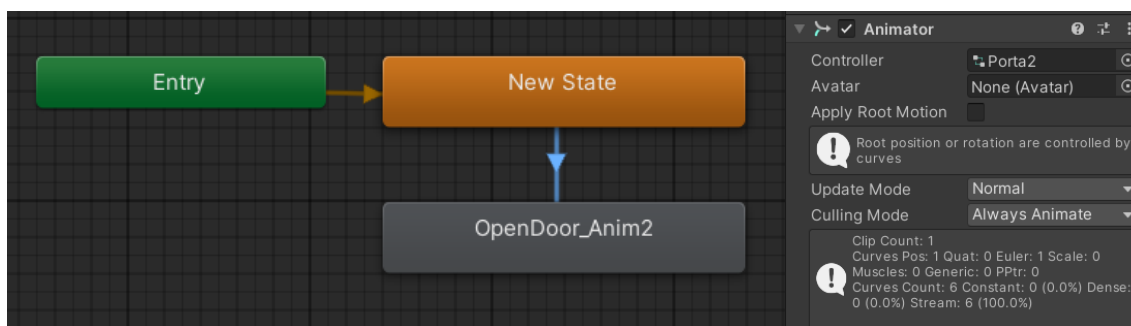
39 Antes da interação



40 Depois da interação

```
if (anim != null)
{
    anim.SetTrigger("OpenDoor2");
}
private Animator anim; anim = GetComponent<Animator>();
```

41 Modificações do código



42 Editor de animações

Botões

No corredor adicionei 8 botões que não fazem nada de especial porque são uma distração.



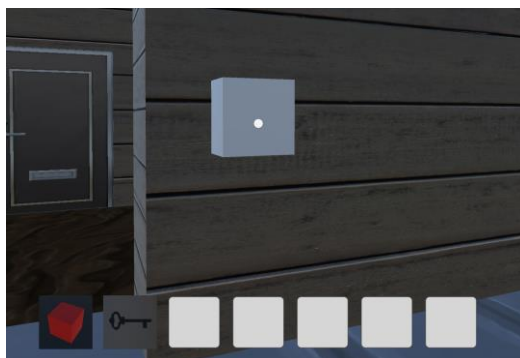
43 Botões

Os botões usam o mesmo sistema que o interruptor e a luz da segunda sala, que vai estar mais à frente.

Luz Código Morse

Para ativar a luz existe um interruptor que era suposto ser uma alavanca.

Um dos scripts verifica se o jogador clicou no botão e começa a transmissão.



44 Interruptor e lâmpada

```
public class LeverInteraction : MonoBehaviour
{
    public MorseCodeLamp lampScript;

    void Update()
    {
        if (Input.GetMouseButtonDown(0))
        {
            Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);
            RaycastHit hit;

            if (Physics.Raycast(ray, out hit))
            {
                if (hit.collider.gameObject == gameObject)
                {
                    if (lampScript != null)
                    {
                        lampScript.StartTransmission();
                    }
                }
            }
        }
    }
}
```

45 Começo da transmissão

A transmissão é feita do tempo que a luz está acesa. Sendo o . meio segundo e o – um segundo.

```
public Light lamplight;
public float shortDelay = 0.5f; // .
public float longDelay = 1.0f; // -
public string morseCodeMessage = "-.... .-.-. -... -.-.-";
```

46 Mensagem em código morse

```
public void StartTransmission()
{
    if (!isTransmitting)
    {
        StartCoroutine(TransmitMorseCode());
    }
}

public void StopTransmission()
{
    if (isTransmitting)
    {
        StopCoroutine(TransmitMorseCode());
        lamplight.enabled = false;
        isTransmitting = false;
    }
}
```

47 Começo e fim da transmissão

Verifica o que recebeu, se foi . ou – e manda o sinal para transmitir

```
IEnumerator TransmitMorseCode()
{
    isTransmitting = true;

    foreach (char c in morseCodeMessage)
    {
        if (c == '.')
        {
            lamplight.enabled = true;
            yield return new WaitForSeconds(shortDelay);
        }
        else if (c == '-')
        {
            lamplight.enabled = true;
            yield return new WaitForSeconds(longDelay);
        }

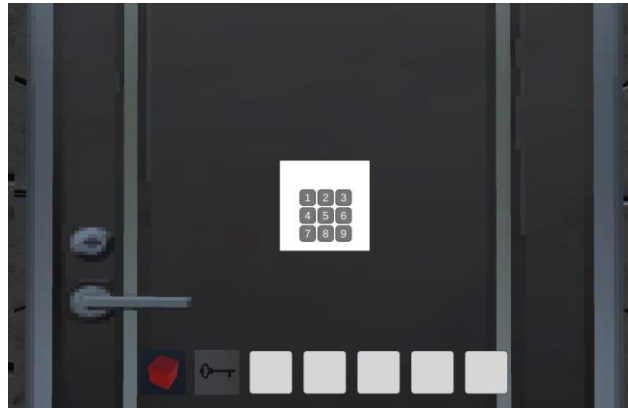
        lamplight.enabled = false;
        yield return new WaitForSeconds(shortDelay);
    }

    isTransmitting = false;
}
```

48 Transmissão

Porta com código

Para entrar na terceira e última sala é preciso um código, quando clicas no “F” abre o menu para inserir o código.



49 Code Lock

Se o jogador estiver perto da porta e clicar no “F” o painel vai abrir, verifica se o código inserido é igual ao código predefinido, não pode ter 5 dígitos, e se o código for igual vai dar a animação da porta a abrir.

```
void Update()
{
    CodeText.text = codeTextValue;

    if (codeTextValue == safeCode)
    {
        anim.SetTrigger("OpenDoor");
        CodePanel.SetActive(false);
    }

    if (codeTextValue.Length >= 5)
    {
        codeTextValue = "";
    }

    if (IsAtDoor && !isPanelOpen && Input.GetKeyDown(KeyCode.F))
    {
        ToggleCodePanel(true);
    }
}
```

50 Code Lock



51 Porta aberta

Verifica se o jogador está dentro da área para abrir o painel, se no início estiver perto, abrir e depois afastar-se o painel vai fechar.

```
private void OnTriggerEnter(Collider other)
{
    if (other.CompareTag("Player"))
    {
        IsAtDoor = true;
    }
}

private void OnTriggerExit(Collider other)
{
    if (other.CompareTag("Player"))
    {
        IsAtDoor = false;
        ToggleCodePanel(false);
    }
}
```

52 Verifica se o jogador está na área

Quando se clica num dos botões adiciona-se esse dígito.

```
public void AddDigit(string digit)
{
    if (isPanelOpen)
    {
        codeTextValue += digit;
    }
}
```

53 Adicionar dígito

Quando o painel estiver ativo, o cursor vai aparecer para poder carregar nos botões.

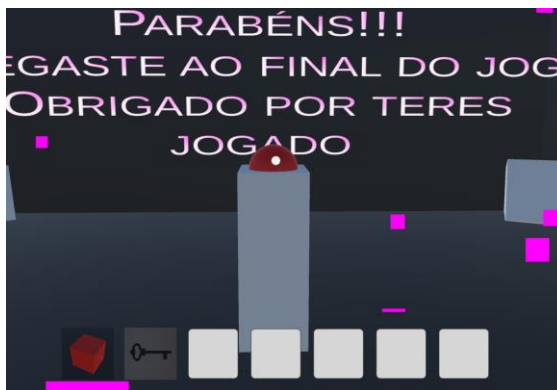
```
void ToggleCodePanel(bool state)
{
    isPanelOpen = state;
    CodePanel.SetActive(state);

    Cursor.visible = state;
    Cursor.lockState = state ? CursorLockMode.None : CursorLockMode.Locked;
}
```

54 Estado do painel

Botão de saída

Na última sala encontra-se um botão que quando se carrega sai do jogo.



55 Botão final

O botão funciona da mesma forma do que o interruptor da segunda sala, só que em vez de fazer a transmissão do código morse, ele sai da aplicação.

```
public class Quit : MonoBehaviour
{
    void Update()
    {
        if (Input.GetMouseButtonDown(0))
        {
            Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);
            RaycastHit hit;

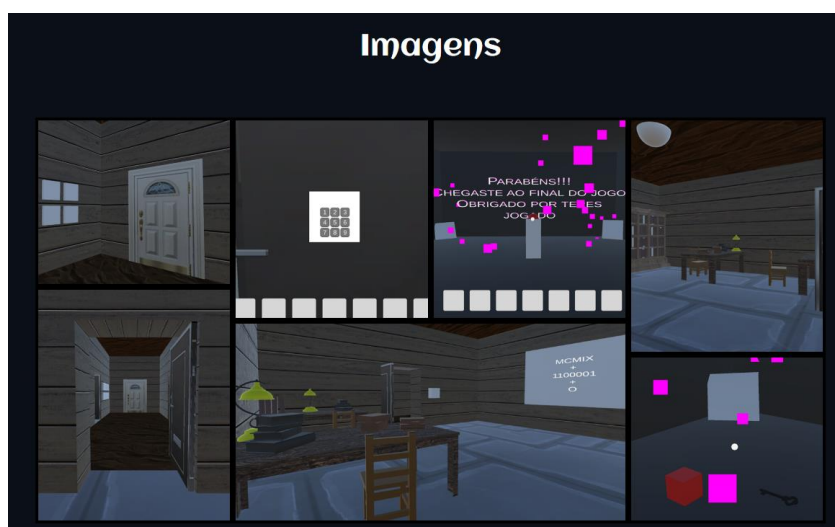
            if (Physics.Raycast(ray, out hit))
            {
                if (hit.collider.gameObject == gameObject)
                {
                    Application.Quit();
                }
            }
        }
    }
}
```

56 Botão sair

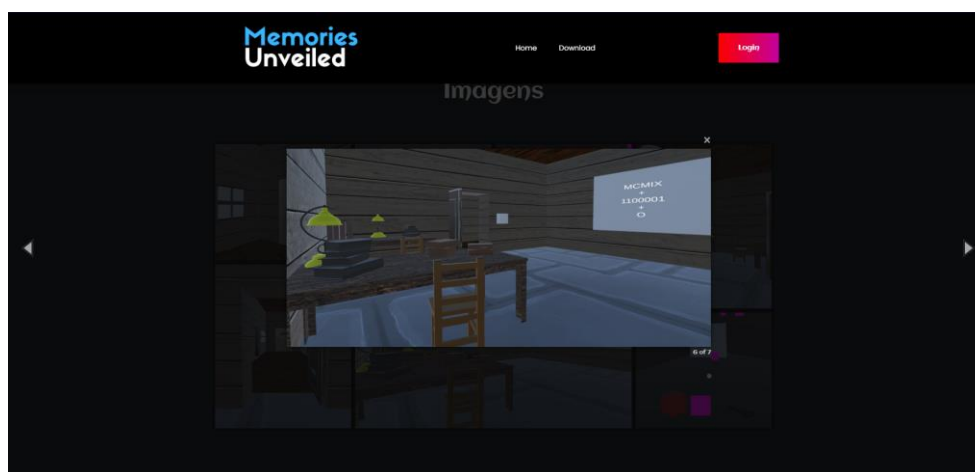
3. Site

Home

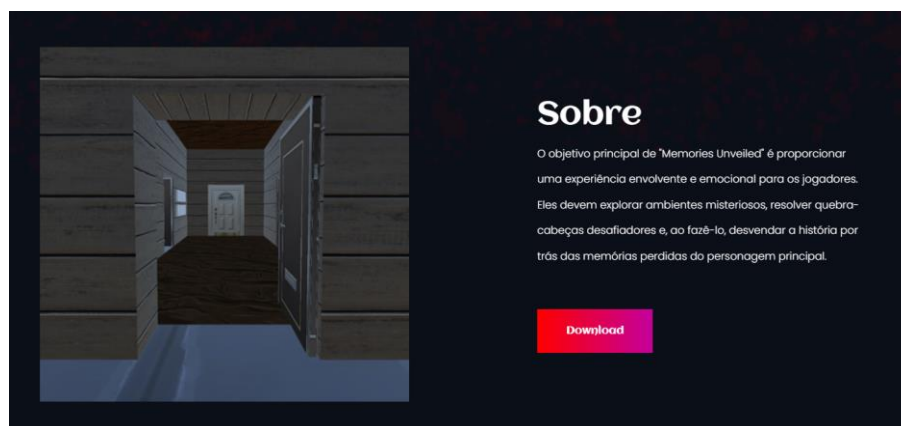
A página principal é uma página simples, que tem informação e imagens sobre o jogo, tem breves descrições sobre o jogo, um botão para ir para a página de download, e um carrousel de imagens com imagens do meu jogo.



57 Coleção de imagens



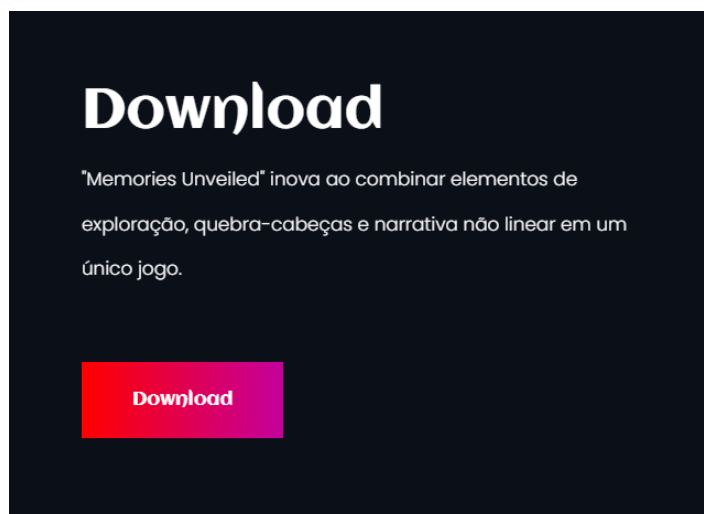
58 Carrousel de imagens



59 Sobre o jogo

Download

Na página de download ao clicar no botão que diz download vai dar acesso ao meu jogo a quem o quiser transferir.



60 Botão download

Login

O meu login é uma página muito rudimentar em que apenas tem os campos “Email or Username” e “Password” e dois botões, um para efetuar o login e outro para a criação de conta que redireciona o utilizador para a página do registo.

User Login

Email or Username:

Password:

Login

Don't have an account? [Create account](#)

61 Login

Registo

A página de registo é parecida com a de login, só que quando o utilizador efetuar o registo os dados vão para a base de dados, e vai diretamente para a página de login.

User Registration

Email:

Username:

Password:

62 Registo

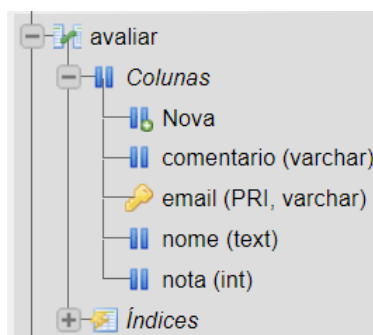
Base de dados

Eu fiz a minha base de dados foi feita em phpMyAdmin, consiste em 3 tabelas, uma para receber as avaliações do jogo, uma para a recolha de dados para fazer o demográfico de jogadores e outra com os utilizadores para poderem efetuar o registo e o login.



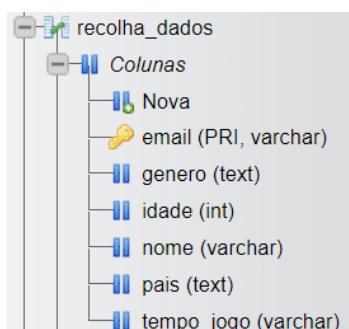
63 Tabelas

Na tabela avaliar tem os campos “nome”, “nota”, “comentario” e “email”.



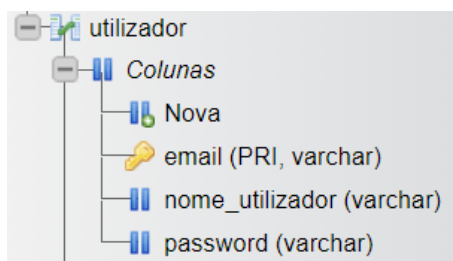
64 Tabela avaliar

Na tabela recolha_dados tem os campos “email”, “genero”, “idade”, “nome”, “pais” e “tempo_jogo”.



65 Tabela recolha_dados

Na tabela utilizador tem os campos “email”, “nome_utilizador” e “password”.



66 Tabela utilizador

4. Alterações Futuras

Até à data da defesa da PAP pretendo adicionar mais texturas, modelos e animações, desenvolver uma história para o jogo visto que não tive tempo de a fazer tendo em conta que não era prioritário, adicionar músicas, efeitos sonoros e opções de aumentar e diminuir volume no menu das opções.

Pretendo também adicionar a página de avaliações e comentários do jogo ao meu website, melhorar o login e registo e aprofundar o perfil de utilizador.

Conclusão

Este percurso foi difícil, mas estamos aqui.

Durante a minha jornada tive muitas dificuldades, mas aprendi a superá-las devido ao trabalho contínuo que os professores, os colegas, os amigos e a família tiveram, foram estas pessoas e eu que me moldaram para ser a pessoa que sou hoje e por isso agradeço novamente a todos que conheci porque sem eles eu provavelmente não estaria aqui ou não seria como sou hoje.

Estes três anos ensinaram-me muitas coisas, mas fundamentalmente a aprender a aprender que é o essencial para continuar a viver.

Bibliografia

<https://www.youtube.com/@HawkesByte>

<https://www.youtube.com/@freecodecamp>

<https://www.youtube.com/@BroCodez>

<https://www.youtube.com/@GMTK>

<https://www.youtube.com/@CodeMonkeyUnity>

<https://www.youtube.com/@CaffeineAlgorithm>

<https://www.youtube.com/@GameAssetWorld>

<https://www.youtube.com/@NattyGameDev>

<https://www.youtube.com/@BMoDev>

<https://github.com/>

Anexos

Anexo 1 – Curriculum Vitae


europass

Eduardo Miguel Felgueiras Marinho

Nacionalidade: Portuguesa **Data de nascimento:** 19/09/2006 **Sexo:** Masculino

Número de telemóvel: (+351) 932274333 **Endereço de email:** 0521186@alunos.epb.pt

SOBRE MIM

Pode inserir aqui a sua descrição...

EXPERIÊNCIA PROFISSIONAL

Estudante
EPB - Escola Profissional de Braga [15/09/2021 – Atual]
 Cidade: Braga
 País: Portugal

Estudante do curso Gestão e Programação de Sistemas Informáticos

Estagiário
Universidade do Minho - Escola de Medicina [05/06/2023 – 25/07/2023]
 Cidade: Braga
 País: Portugal

Elaboração de programas em C#;
 Instalação de sistemas operativos;
 Manutenção de impressoras.

COMPETÊNCIAS LINGÜÍSTICAS

Língua(s) materna(s): português

Outra(s) língua(s):
Inglês
 COMPREENSÃO ORAL B1 LEITURA B1 ESCRIVER B1
 PRODUÇÃO ORAL B1 INTERAÇÃO ORAL B1

Níveis: A1 e A2: Utilizador de base; B1 e B2: Utilizador independente; C1 e C2: Utilizador avançado

VOLUNTARIADO

EMACI 2022 - European Masters Athletics Championships Indoor
 [Braga, 19/02/2022 – 27/02/2022]

Centro de Convívio Sénior de Real
 [Braga, 04/2023 – 04/2023]