

Relatório da Prova de Aptidão Profissional

Puzzle of Blessings

João Pedro Lopes Pais de Andrade, N.º13 da Turma 12.ºP

Professor Orientador:

Sandra Crucho

Diretora de Curso:

Clara Ferreira

Ano Letivo – 2025/2026

Cofinanciado por:



Os Fundos Europeus mais próximos de si.

Agrupamento De Escolas Dr. Ginestal Machado
Curso Profissional Técnico Programador de Informática
Triénio 2023/2026

Prova de Aptidão Profissional

Relatório do projeto do 12.º Ano do Curso de Técnico de Programador de Informática, submetido à Escola Secundária Dr. Ginestal Machado, para cumprimento dos requisitos necessários à obtenção de aprovação na Prova de Aptidão Profissional, elaborado sob a orientação da Professora Sandra Crucho.

Santarém, maio de 2026

Cofinanciado por:



Os Fundos Europeus mais próximos de si.



AGRADECIMENTOS

Durante estes três anos, tenho a sensação que tenho sorte por tudo o que me aconteceu e as pessoas que me passar pelo percurso. Por isso, venho agradecer da minha forma mais sincera.

Quero começar pelo Professor Luís Magalhães, que nos ensinou com atenção e paciência, uma forma de explicar simplificada e fácil de entender tudo. Com ele consegui ter uma noção como funciona o mundo do trabalho na área da programação, como devemos olhar para o futuro e como podemos tirar partido das ferramentas que temos ao nosso alcance.

Quero deixar um dos meus agradecimentos mais especiais à Professora Sandra Crucho, pelo entusiasmo e curiosidade que sempre me fizeram querer desenvolver mais, por todos os projetos criativos que adorei desenvolver, testar e apresentar, pelo interesse genuíno, pelo carinho e dedicação que consigo sentir em todo o momento, uma das professoras que nunca irei esquecer, que marcou meu percurso, e isso levo com muita gratidão.

Outro dos meus agradecimentos mais especiais, vai à Professora Clara Ferreira, a pessoa que mais me marcou e que foi, responsável pela minha paixão pela programação. A minha primeira impressão foi que a dedicação vai muito para além do que normalmente se espera de um professor. Consigo sentir um entusiasmo como quem celebra cada conquista dos alunos, como suas. Sempre muito criativa nas atividades ou desafios, e a forma como acredita nos alunos fizeram toda a diferença em mim. Enxerga o que eles próprios não veem muitas vezes. E eu acredito que, com dedicação, não existem pessoas incapazes com a professora. Foi graças a isso tudo que participei em concursos, entrei em equipas e desafios, desenvolvi projetos e aprendi a comunicar com mais confiança. Acima de tudo, a Professora Clara acreditou em mim quando talvez outros não acreditassem, e isso para mim vale mais do que tudo.

Por fim, também quero agradecer à escola e ao Diretor, pelas oportunidades que me propuseram, especialmente pela possibilidade de participar no programa Erasmus, que me levou a Split, na Croácia. Uma experiência que me marcou imenso, e que me tornou mais maduro, mais confiante no inglês e me deixou memórias que definitivamente vou carregar para a minha vida.



ACRÓNIMOS/SIGLAS

PAP – Prova de Aptidão Profissional

URP – Universal Render Pipeline

LOD – Level of Detail

NPC – Non-Playable Character

HUD – Heads-Up Display

API – Application Programming Interface

IDE – Integrated Development Environment (Ambiente de Interface Integrado)

PREFAB – Pre Fabricated (Pré-Fabricado)



RESUMO

O presente relatório descreve o desenvolvimento do projeto ‘*Puzzle of Blessings*’, no âmbito da Prova de Aptidão Profissional do Curso Profissional de Técnico Programador de Informática. O projeto consiste num jogo digital tridimensional de desafios e puzzles, desenvolvido no motor de jogo Unity com a *Universal Render Pipeline (URP)*, ligado a uma base de dados online Supabase e a um website externo desenvolvido em colaboração com outro aluno do mesmo curso.

O jogo decorre num ambiente composto por ilhas, onde o jogador resolve puzzles de complexidade crescente para progredir entre níveis. O sistema inclui autenticação de utilizadores, gravação de progresso, um conjunto de habilidades desbloqueáveis denominadas Blessings, e um launcher personalizado com atualização automática via nuvem.

O público-alvo principal são crianças e jovens entre os 11 e os 15 anos, tendo o projeto sido orientado por princípios de aprendizagem ativa, com base nas teorias de Jean Piaget e David Kolb. O desenvolvimento decorreu entre setembro de 2025 e maio de 2026, seguindo uma abordagem iterativa e por fases.

Palavras-chave: Unity; C#; Supabase; Jogo educativo; Puzzle; Aprendizagem ativa; Blender.



ABSTRACT

This report describes the development of the ‘Puzzle of Blessings’ project, as part of the Professional Aptitude Test for the Professional Course in Computer Programming. The project consists of a three-dimensional digital game featuring challenges and puzzles, developed using the Unity game engine with the Universal Render Pipeline (URP), connected to a Supabase online database and an external website developed in collaboration with another student on the same course.

The game takes place in an environment consisting of islands, where the player solves puzzles of increasing complexity to progress through the levels. The system includes user authentication, progress saving, a set of unlockable skills called Blessings, and a custom launcher with automatic cloud-based updates.

The main target audience is children and young people aged between 11 and 15 [SC3.1], with the project guided by principles of active learning, based on the theories of Jean Piaget and David Kolb. Development took place between September 2025 [SC4.1] and May 2026, following an iterative and phased approach.

Keywords: Unity; C#; Supabase; Educational game; Puzzle; Active learning; Blender.



Índice

AGRADECIMENTOS.....	III
ACRÓNIMOS/SIGLAS.....	IV
RESUMO.....	V
ABSTRACT	VI
LISTA DE FIGURAS.....	XI
INTRODUÇÃO.....	1
ENQUADRAMENTO TEÓRICO	2
1 - Ferramentas Utilizadas.....	4
1.1 - Motor, IDE e Linguagem.....	4
1.1.1 — Unity.....	4
1.1.2 — Visual Studio	5
1.1.3 — AntiGravity.....	5
1.2 - Base de Dados e Backend	6
1.2.1 — Supabase.....	6
1.3 — Launcher	6
1.3.1 — WPF / .NET 8	6
1.1.1 — Inno Setup	7
1.4 - Design e Recursos Visuais	8
1.4.1 — Blender.....	8
1.4.2 — Sketchfab / Unity Asset Store	9
1.4.3 — Pixabay.....	9
1.4.4 — Google Flow	10
1.4.5 — Pixlr	10
2 - Procedimento.....	11
2.1 — Prototipagem, Brainstorm e Testes de Ligação à Base de Dados	11
2.1.1 — Brainstorm.....	11
2.1.2 — Criação da História do Jogo.....	12



2.1.3 – Prototipagem e Testes.....	12
2.2 — Prefab de <i>Player</i> e Mecânica de Agarrar Objetos	14
2.2.1 – <i>Player</i>	14
2.2.2 – Mecânica de Agarrar (<i>Grab System</i>)	14
2.2.2 – <i>GrabbableObject.cs</i> – Definição dos objetos interativos	15
2.3 — Primeiro Menu e Planeamento do Login.....	16
2.4 — Sistema de Login com Base de Dados	17
2.4.1 – Implementação da Interface de Login.....	17
2.4.2 – Funcionamento do Sistema de Login.....	17
2.4.3 Proteção das palavras-passe através de hashing	18
2.5 — Criação do Mapa Principal (Ilhas)	19
2.6 — Tutorial: Chave e Portão	21
2.7 — Limitações, <i>Respawn</i> por queda em água	22
2.7.1 – Funcionamento do <i>WaterRespawn.cs</i>	22
2.8 — Menu em Tempo de Jogo e Sistema de Salvar Progresso	23
2.8.1 — Menu e Definições em Jogo	23
2.8.2 — Sistema de Salvar Jogo.....	23
2.8.3 - Scripts desenvolvidos para o sistema de gravação	24
2.8.4 – Estrutura dos dados guardados em <i>SaveData.cs</i>	24
2.8.5 – Funcionamento do <i>SaveGameController.cs</i>	24
2.9 — Nível 1: Puzzle de Alavancas	26
2.10 — Áudio e Animações	27
2.10.1 – Funcionamento do <i>AudioSettings.cs</i>	27
2.11 — Reformulação Completa do Menu	28
2.11.1 – Fluxo de funcionamento do menu	29
2.12 — Implementação da Personagem Guia Interativa (Fantasma).....	30
2.13 — Pop-ups em World Space.....	31
2.14 — Sistema de <i>Puzzle Coins</i>	32
2.15 — Blessing Manager: Sistema de Habilidades	33
2.15.1 – Funcionamento <i>BlessingManager</i>	34



2.16 — Artefactos Colecionáveis	35
2.17 — <i>Memory Teleporter</i>	35
2.18 — Mecânica de Voo Livre (<i>NoClip</i>).....	36
2.19 — Puzzle “Simon Says” (<i>Memory Spheres</i>)	37
2.19.1 — Geração do Puzzle	37
2.19.2 — Funcionamento do Puzzle.....	37
2.19.3 — Integração com a base de dados.....	38
2.20 — Power-up de Velocidade (<i>Speedup</i>)	39
2.21 — Fase Final do Puzzle Of Blessings.....	40
2.21.1 — Transição Narrativa (<i>EndgameTrigger</i>).....	42
2.21.2 — Persistência e New Game+	42
2.22 — Arquitetura de Distribuição e Launcher Personalizado.....	43
2.22.1 — Launcher Desenvolvido em WPF	43
2.22.2 — Integração com Supabase	43
2.22.3 — Download e extração dos ficheiros	44
2.22.4 — Tratamento de erros e experiência do utilizador	44
2.22.5 — Instalação do Launcher	44
2.23.1 — Sistema de Auto-Update do Launcher	45
2.23.2 — Controle de versões Supabase.....	45
2.23.3 — Verificação automática ao iniciar	45
2.23.4 — Substituição dos ficheiros em uso	45
2.23.5 — Registo de Diagnóstico.....	45
2.23.6 — Permissões de Administrador	46
3 — Manual do Utilizador	47
3.1 — Instalação do Jogo	47
3.1.1 — Etapa 1	47
3.1.2 — Etapa 2	50
3.1.3 — Arranque do Jogo	51
3.2 — Habilidades / Blessings do Jogo	54
3.2.1— Descrição Blessings	55
3.3 — Objetos Seguráveis.....	56



3.3.1 – Dado Azul.....	56
3.3.2 – Chave Tutorial	56
3.3.3 – Memory Fragment.....	57
3.4 – Colecionáveis.....	58
3.4.1 – Cassete VHS.....	58
3.4.2 – Troféu de Escalada	58
3.4.3 – Almofada dos Sonhadores.....	59
3.5 – Níveis do Jogo.....	59
3.5.1 – Tutorial	59
3.5.2 – Nível 1 (Genesis Field).....	60
3.5.3 – Nível 2 (Deep Exploration).....	60
3.5.4 – Nível 3 (Spirit Realm)	61
4 – TRABALHO FUTURO.....	62
RESULTADOS E DISCUSSÃO.....	63
Considerações Finais.....	64
BIBLIOGRAFIA	65
Anexo 1 – Avaliações	ii
Anexo 1.1 – Primeira Avaliação Acompanhamento PAP	ii
Anexo 1.2 – Segunda Avaliação Acompanhamento PAP	iii
Anexo 1.3 – Terceira Avaliação Acompanhamento PAP.....	iv
Anexo 1.4 – Autoavaliação.....	v

LISTA DE FIGURAS

Figura 1 - Esboço da ideia com o Google Flow.....	1
Figura 2 - Dois livros de referência sobre aprendizagem e desenvolvimento humano: um sobre a psicologia da criança e outro sobre a aprendizagem experiencial.	3
Figura 3 - Unity	4
Figura 4 - Visual Studio.....	5
Figura 5 - Supabase	6
Figura 6 - .NET8	6
Figura 7 - Inno Setup	7
Figura 8 - Blender	8
Figura 9 - Modelo 3 de Puzzle (desenvolvido por mim).....	8
Figura 10 - Sketchfab.....	9
Figura 11 - Pixabay.....	9
Figura 12 - Google Flow.....	10
Figura 13 - Pixlr	10
Figura 14 - Processo Creativo e Brainstorm	11
Figura 15 - Processo de Construção de um Modelo Inicial	13
Figura 16 - Contador e Comunicação com a Base de Dados.....	13
Figura 17 - Protótipo e cena de testes	14
Figura 18 - Mecânica de agarrar atualizada	15
Figura 19 - Desenvolvimento do menu principal	16
Figura 20 - Sistema de Login e Hashing	18
Figura 21 - Criação do mapa	19
Figura 22 - Visual final do mapa	20
Figura 23 - Planejamento de Níveis	20
Figura 24 - Desenvolvimento do Tutorial.....	21
Figura 25 - Protótipo Menu em Jogo.....	25
Figura 26 - Sistema de guardar progresso	25
Figura 27 - Desenvolvimento Primeiro Nível	26
Figura 28 - Funcionamento da Alavanca.....	27
Figura 29 - Tela de Configurações de Jogo	28
Figura 30 - Sistema de Funcionamento do Menu.....	29
Figura 31 - Modelo Fantasma	30
Figura 32 - Tutoriais.....	31



Figura 33 - Puzzle Coin	32
Figura 34 - Sistema de Cartas	33
Figura 35 - Desafio Simon Says	38
Figura 36 - Powerup de Velocidade	39
Figura 37 - Desafio Floresta Assombrada	40
Figura 38 - Desafio Final Spirit Anchor e Duplication	41
Figura 39 - Estrutura de Funcionamento do Launcher	46
Figura 40 - Download no Site	47
Figura 41 - Abrir Executável	47
Figura 42 - Smartscreen Windows	48
Figura 43 - Instalador	48
Figura 44 - Criar Executável no Ambiente de Trabalho	49
Figura 45 - Instalar	49
Figura 46 - Abrir Launcher	50
Figura 47 - Instalar Jogo	50
Figura 48 - Confirmar	51
Figura 49 - Abrir Jogo	52
Figura 50 - Abrir Login	52
Figura 51 - Iniciar Sessão Ou Criar Conta	53
Figura 52 - Novo Jogo ou Continuar	53
Figura 53 - Cartas Disponíveis	54
Figura 54 - Dado Azul	56
Figura 55 - Chave Tutorial	56
Figura 56 - Fragmento de Memória	57
Figura 57 - Cassete VHS	58
Figura 58 - Troféu de Escalada	58
Figura 59 - Almofada dos Sonhadores	59
Figura 60 - Tutorial	59
Figura 61 - Genesis Field	60
Figura 62 - Deep Exploration	60
Figura 63 - Spirit Realm	61

INTRODUÇÃO

‘Puzzle Of Blessings’ procura um público-alvo de crianças entre os 11 e os 15 anos, incluindo crianças com autismo, através de desafios que procuram captar a atenção e promover a aprendizagem ativa por meio dos jogos. A escolha deste projeto resultou de uma motivação pessoal: criar uma experiência de jogo acessível, tendo como referência próxima um familiar com perturbações ligeiras do espectro do autismo. Apesar de esse não ser o objetivo principal do projeto, essa foi a principal motivação para a sua realização. O meu projeto foi desenvolvido com recurso à Unity, utilizando a *Universal Render Pipeline (URP)*, está ligado a uma base de dados Supabase e inclui ainda um website próprio, desenvolvido pelo aluno da minha turma número 18, Salvador Felipe, e existindo assim uma colaboração com vista ao cumprimento dos objetivos de cada Prova de Aptidão Profissional (PAP). O jogo decorre num ambiente composto por ilhas interligadas por pontes ou teletransportes, onde o jogador resolve puzzles simples para avançar entre os diferentes níveis.

Figura 1 - Esboço da ideia com o Google Flow



ENQUADRAMENTO TEÓRICO

O desenvolvimento do projeto *Puzzle of Blessings* baseia-se em princípios da aprendizagem ativa, suportados por teorias construtivistas da educação, nomeadamente as contribuições de Jean Piaget e David Kolb.

Segundo Jean Piaget, a aprendizagem é um processo ativo no qual o indivíduo constrói o seu próprio conhecimento através da interação com o meio envolvente. O autor defende que as crianças não são apenas recetoras passivas de informação, mas participantes ativas no processo de aprendizagem, explorando, experimentando e resolvendo problemas. (Piaget, 1969)

Esta perspetiva é fundamental no contexto de jogos educativos, uma vez que permite que o utilizador aprenda através da ação e da descoberta, reforçando competências cognitivas como o raciocínio lógico, a atenção e a resolução de problemas. (Piaget, 1969)

Por sua vez, David Kolb desenvolveu a teoria da aprendizagem experiencial, que descreve a aprendizagem como um ciclo contínuo composto por quatro etapas: experiência concreta, observação reflexiva, conceptualização abstrata e experimentação ativa. Este modelo destaca a importância da experiência direta como ponto de partida para a construção do conhecimento, sendo especialmente relevante em ambientes interativos como os jogos digitais. Através deste ciclo, o utilizador aprende com a prática, refletindo sobre as suas ações e aplicando novos conhecimentos em situações futuras. (Kolb, 1984)

A aprendizagem ativa, conceito central neste projeto, caracteriza-se pela participação direta do indivíduo no processo de aprendizagem. Em vez de uma abordagem passiva, o utilizador é incentivado a explorar, tomar decisões e resolver desafios. Este tipo de aprendizagem tem-se revelado particularmente eficaz em contextos educativos digitais, onde o envolvimento e a motivação desempenham um papel crucial. (Minnesota, 2023)

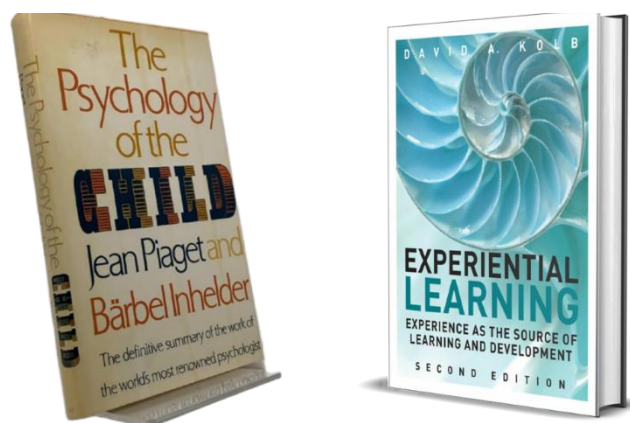


No contexto do público-alvo deste projeto, que inclui crianças entre os 11 e os 15 anos e perturbações do espectro do autismo, a utilização de jogos como ferramenta de aprendizagem pode trazer benefícios significativos. Ambientes estruturados, objetivos claros e feedback imediato são características que facilitam a concentração e o envolvimento, contribuindo para uma experiência de aprendizagem mais eficaz e acessível. (Society, National Autistic, 2023)

Neste sentido, o jogo *Puzzle of Blessings* foi construído com base nestes princípios teóricos, promovendo a aprendizagem através da interação, da experimentação e da resolução de puzzles.

(Piaget, 1969) (Kolb, 1984)

Figura 2 - Dois livros de referência sobre aprendizagem e desenvolvimento humano: um sobre a psicologia da criança e outro sobre a aprendizagem experiencial.



Fonte da Imagem: [AbeBooks](#) e [Institute for Experimental Learning](#)



1 - FERRAMENTAS UTILIZADAS

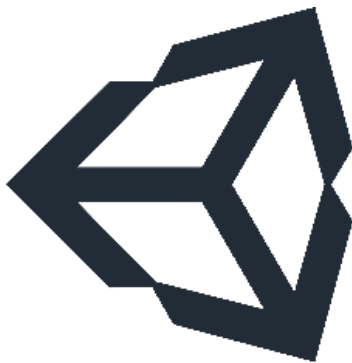
No desenvolvimento deste projeto foram utilizadas diversas ferramentas e tecnologias, cada uma com um papel específico na construção e funcionamento do jogo. A seleção destas ferramentas teve em consideração critérios como a facilidade de utilização, desempenho, compatibilidade e adequação aos objetivos do projeto.

1.1 - Motor, IDE e Linguagem

1.1.1 — Unity

O motor de jogo utilizado foi o Unity, com o pipeline de renderização URP (Universal Render Pipeline). A escolha do Unity deveu-se à sua ampla utilização na indústria, à grande quantidade de recursos disponíveis e à sua versatilidade no desenvolvimento de jogos 2D e 3D. A utilização do URP permitiu alcançar um bom equilíbrio entre desempenho e qualidade gráfica, sendo especialmente adequado para garantir fluidez em diferentes tipos de hardware. Para a programação da lógica do projeto foi utilizada a linguagem C#, integrada no próprio Unity, escolhida pela sua robustez, organização e compatibilidade direta com o motor de jogo.

Figura 3 - Unity



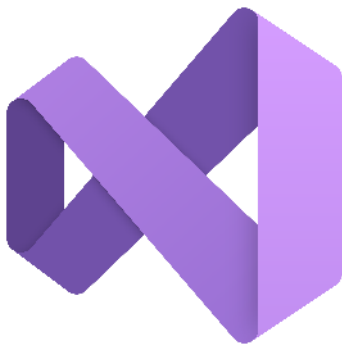
Fonte da Imagem: [WorldVectorLogo](https://www.worldvectorlogo.com/logo/unity-vector-illustration.html)



1.1.2 — Visual Studio

Como ambiente de desenvolvimento, foi utilizado o Visual Studio, devido à sua integração eficiente com o Unity e ao conjunto de funcionalidades que disponibiliza, como o destaque de sintaxe, depuração de código (debugging) e sugestões automáticas (IntelliSense), que facilitam o desenvolvimento e a deteção de erros.

Figura 4 - Visual Studio

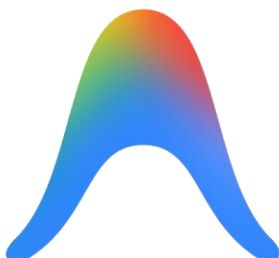


Fonte da Imagem: [WIKIMEDIACOMMONS](#)

1.1.3 — AntiGravity

Para ferramenta de apoio ao desenvolvimento, foi utilizado o AntiGravity, devido à sua utilidade na identificação e correção de bugs em scripts, bem como na aceleração do processo de criação e na realização de adaptações ao código. Organizando o fluxo de trabalho, tornando o desenvolvimento mais eficiente e facilitando a implementação de alterações de forma mais rápida e prática.

Figura 5 - AntiGravity



Fonte da Imagem: [AntiGravity](#)



1.2 - Base de Dados e Backend

1.2.1 — Supabase

Para a base de dados escolheu-se o Supabase, Plataforma de base de dados na nuvem baseada em *PostgreSQL*, utilizada para gerir a autenticação dos utilizadores, sincronizar as Bênções desbloqueadas e guardar dados de progresso entre sessões. Foi escolhida por oferecer uma *API REST* simples de integrar com Unity e no website, por ser gratuita para projetos de pequena escala e por permitir sincronização em tempo real sem necessidade de desenvolver um servidor próprio.

Figura 6 - Supabase



Fonte da Imagem: [Supabase](https://supabase.com)

1.3 – Launcher

1.3.1 — WPF / .NET 8

Windows Presentation Foundation, é a *Framework* utilizada para desenvolver o Launcher do jogo, a aplicação responsável por instalar, atualizar e lançar o executável do Puzzle of Blessings. Foi escolhida por ser nativa do ecossistema Windows, por permitir criar interfaces gráficas modernas com *XAML* e por se integrar facilmente com o .NET 8, garantindo desempenho e compatibilidade com sistemas Windows 10 e 11.

Figura 7 - DotNET8



Fonte da Imagem: [DontPanicLabs](https://dotnet.microsoft.com)

1.1.1 — Inno Setup

A ferramenta utilizada para criar o executável foi o Inno Setup (*LauncherInstaller.exe*). Permite empacotar a aplicação num único ficheiro executável de instalação, com opções de escolha de pasta de destino, criação de atalhos e assistente passo a passo. Foi escolhida por ser gratuita, amplamente utilizada para distribuição de software *Windows* e por gerar instaladores leves e compatíveis com todas as versões modernas do sistema operativo.

Figura 8 - Inno Setup



Fonte da Imagem: [Wikipedia](https://en.wikipedia.org/wiki/Inno_Setup)

1.4 - Design e Recursos Visuais

1.4.1 — Blender

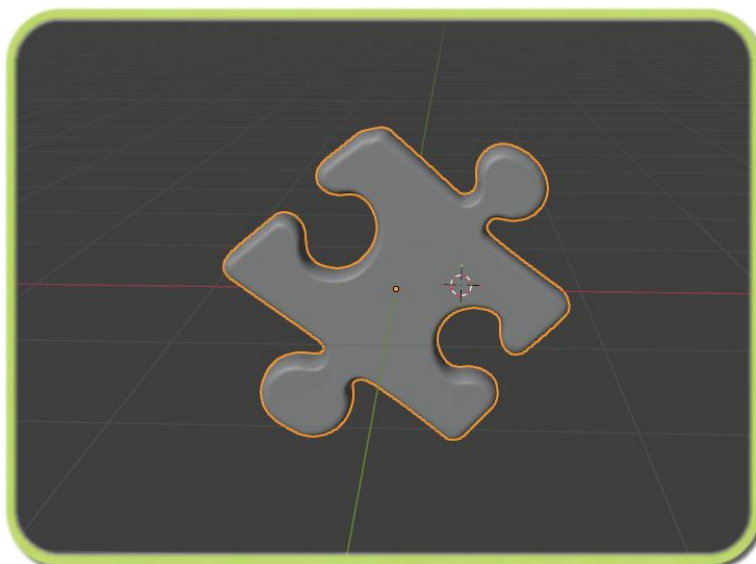
Na componente gráfica e de modelação, recorreu-se ao Blender para a criação e edição de modelos 3D. Esta ferramenta foi escolhida por ser gratuita, de código aberto e bastante completa, permitindo desenvolver modelos personalizados e ajustados às necessidades do projeto.

Figura 9 - Blender



Fonte da Imagem: [WIKIMEDIACOMMONS](https://commons.wikimedia.org/wiki/File:Blender_2.78_Linux_Icon.png)

Figura 10 - Modelo 3 de Puzzle Desenvolvido por mim



1.4.2 — Sketchfab / Unity Asset Store

Para complementar os recursos visuais, foram utilizados assets gratuitos provenientes do Sketchfab, bem como elementos adicionais obtidos através da Unity Asset Store, incluindo prefabs e componentes de interface. A utilização destes recursos permitiu acelerar o desenvolvimento, reduzindo o tempo necessário para criar conteúdos desde o início, mantendo ao mesmo tempo uma qualidade visual adequada.

Figura 11 - Sketchfab



Fonte da Imagem: [Sketchfab](https://sketchfab.com/)

1.4.3 — Pixabay

O Pixabay foi escolhido como plataforma de recursos multimédia gratuitos e livres de direitos de autor. Foi utilizada para obter efeitos sonoros e músicas de fundo presentes no jogo. Escolhido por disponibilizar uma vasta biblioteca de conteúdos com licença *Creative Commons Zero (CC0)*, permitindo a sua utilização em projetos académicos e comerciais sem restrições.

Figura 12 - Pixabay



Fonte da Imagem: [WorldVectorLogo](https://www.pixabay.com/)



1.4.4 — Google Flow

A ferramenta utilizada para o planeamento e organização do fluxo de desenvolvimento de assets 2D e imagens, para o jogo, foi o Google Flow. Devido a sua ramificação e fluxo de trabalho.

Figura 13 - Google Flow



Fonte da Imagem: [Emergent](#)

1.4.5 — Pixlr

Para o design de elementos de interface (UI), foi utilizada a ferramenta Pixlr, escolhida pela sua simplicidade e acessibilidade, permitindo criar e editar gráficos de forma rápida sem necessidade de software mais complexo.

Figura 14 - Pixlr



Fonte da Imagem: [CDNLOGO](#)

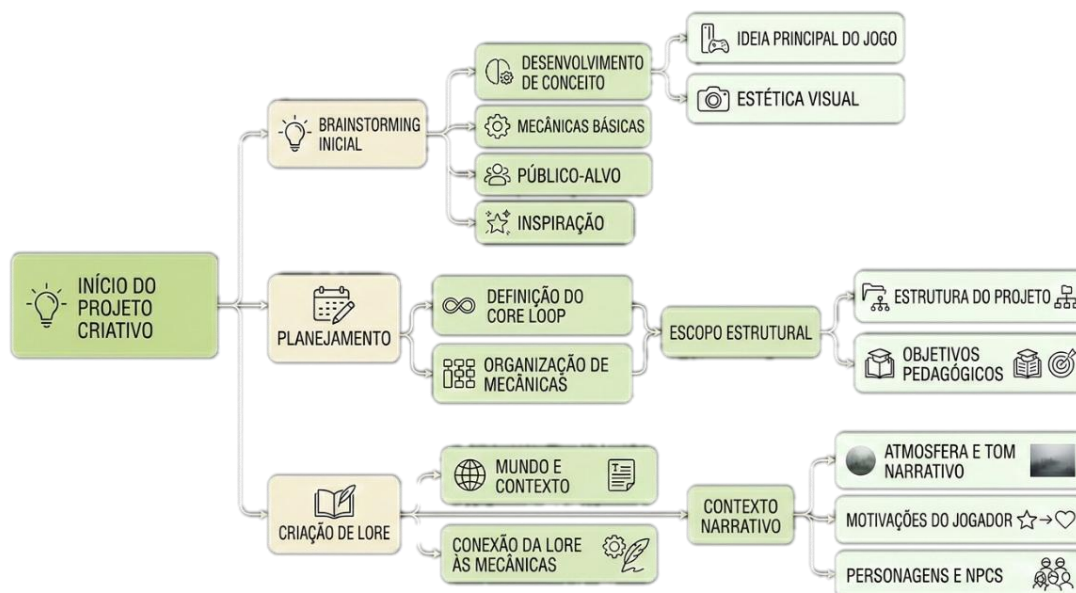
2 - PROCEDIMENTO

2.1 — Prototipagem, Brainstorm e Testes de Ligação à Base de Dados

2.1.1 – Brainstorm

O projeto teve origem num brainstorm inicial centrado na criação de um jogo de puzzle 3D com uma estética low-poly estilizada, cores suaves e efeitos visuais discretos, de modo a proporcionar uma experiência clara e acolhedora. A proposta destacou-se desde o início pela utilização de cartas/benções como ferramentas cognitivas, permitindo ao jogador interagir com o ambiente, manipular objetos e resolver desafios de forma criativa. Em paralelo, foram pensados fantasmas/NPCs que recompensam com cartas, pistas e pequenos fragmentos narrativos. A inspiração em jogos como Portal, The Witness e Half-Life 2 serviu de base para uma experiência assente na lógica espacial, na manipulação física e na introdução progressiva de novas mecânicas, com uma narrativa misteriosa, mas suave e confortável. A dimensão pedagógica também foi considerada desde o início, com objetivos bem definidos, com dinâmica website essencial do meu colega, perfis e conteúdos pedagógicos.

Figura 15 - Processo Criativo e Brainstorm



2.1.2 – Criação da História do Jogo

A partir desta base, foi desenvolvida a história do jogo, situada nas Ilhas do Entre-Sonho, um lugar raro e misterioso que existe entre o real e a memória. Estas ilhas são compostas por elementos naturais e mágicos, onde as regras do mundo exterior não se aplicam da mesma forma. Nesse espaço vivem sobretudo os fantasmas, que preservam ecos de histórias antigas. O protagonista desperta neste local sem memória completa, não por acaso, mas porque foi chamado pelas próprias ilhas, as quais reconhecem nele uma ligação antiga e especial.

Ao longo da aventura, as suas memórias encontram-se fragmentadas e espalhadas pelo cenário, ficando protegidas por puzzles, mecanismos e desafios. Assim, a progressão do jogador não se limita a encontrar uma saída, mas envolve também lembrar, compreender e aceitar. As Blessings surgem neste contexto como poderes concedidos por uma entidade antiga que observa as ilhas e que recompensa o esforço do viajante, funcionando simultaneamente como elemento de gameplay e como avanço narrativo. Do mesmo modo, os puzzles justificam-se como consequência de um mundo “trancado” por camadas de lembrança, onde partes da realidade se encontram misturadas com sonho, tornando cada desafio um passo necessário para abrir caminhos, restaurar memórias e aproximar o protagonista da verdade sobre si próprio.

2.1.3 – Prototipagem e Testes

O objetivo principal neste ponto de criação é testar a comunicação entre o Unity e a base de dados Supabase, sem ainda incluir qualquer elemento definitivo do jogo, criado no Unity com o URP (*Universal Render Pipeline*) tecnologia gráfica da Unity responsável pelo processamento visual do jogo, permitindo melhor desempenho, iluminação otimizada e uma qualidade gráfica mais consistente em diferentes dispositivos., garantindo desde logo uma base preparada para desempenho e qualidade gráfica. De seguida, foi criado um modelo simples de um *player* no Blender, juntamente com um mapa de testes básico que permitisse validar interações dentro do ambiente.

Figura 16 - Processo de Construção de um Modelo Inicial

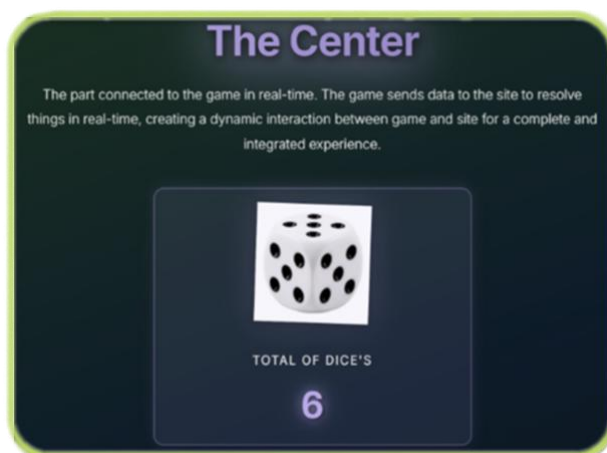


Paralelamente, foi criada a tabela jogador no Supabase, contendo os campos necessários para armazenar a informação do utilizador. Para permitir a comunicação entre o Unity e a base de dados, foi implementado o script SupabaseIntegration.cs, responsável por efetuar pedidos HTTP à API do Supabase. No ambiente de testes do Unity, foi ainda criado um cubo com uma zona de *trigger* associada ao script *CoinCollector.cs*.

Quando o player colidia com este objeto, o sistema lia o valor atual do número de moedas (nmoedas) na base de dados, incrementava esse valor em uma unidade e atualizava o registo correspondente. Após a recolha, o objeto desaparecia da cena.

Como resultado, o teste revelou-se bem-sucedido, uma vez que ao apanhar o cubo no Unity, o número de moedas era atualizado em tempo real na base de dados e refletido corretamente no website. Desta forma, ficou validado o funcionamento do sistema de comunicação entre Unity, Supabase e o portal *web*.

Figura 17 - Contador e Comunicação com a Base de Dados



2.2 — Prefab de *Player* e Mecânica de Agarrar Objetos

Após confirmar a ligação à base de dados, o foco passou para o *player* e para a mecânica central do jogo: pegar e interagir com objetos.

2.2.1 – *Player*

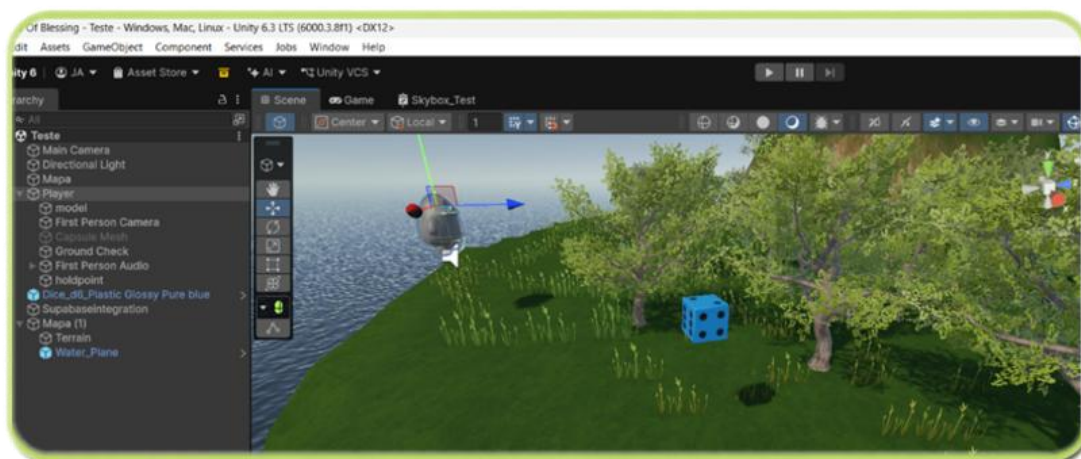
Nesta fase, o modelo de teste foi substituído por um prefab de *player* já existente, que foi posteriormente adaptado e configurado de acordo com as necessidades do projeto. Esta alteração permitiu dar continuidade ao desenvolvimento com uma estrutura mais estável e funcional.

2.2.2 – Mecânica de Agarrar (*Grab System*)

De seguida, foram desenvolvidos dois scripts essenciais para o sistema de interação com objetos, que constituem a base da mecânica de agarrar.

O *GrabController.cs* é o script responsável pela gestão da mecânica de agarrar objetos. Quando o jogador prime a tecla “E” ou clica com o botão esquerdo do rato, o jogo realiza uma verificação esférica na zona frontal, por forma a identificar objetos suscetíveis de serem agarrados. Caso seja encontrado um objeto compatível, este é preso a um ponto invisível localizado à frente do mesmo, denominado HoldPoint.

Figura 18 - Protótipo e cena de testes

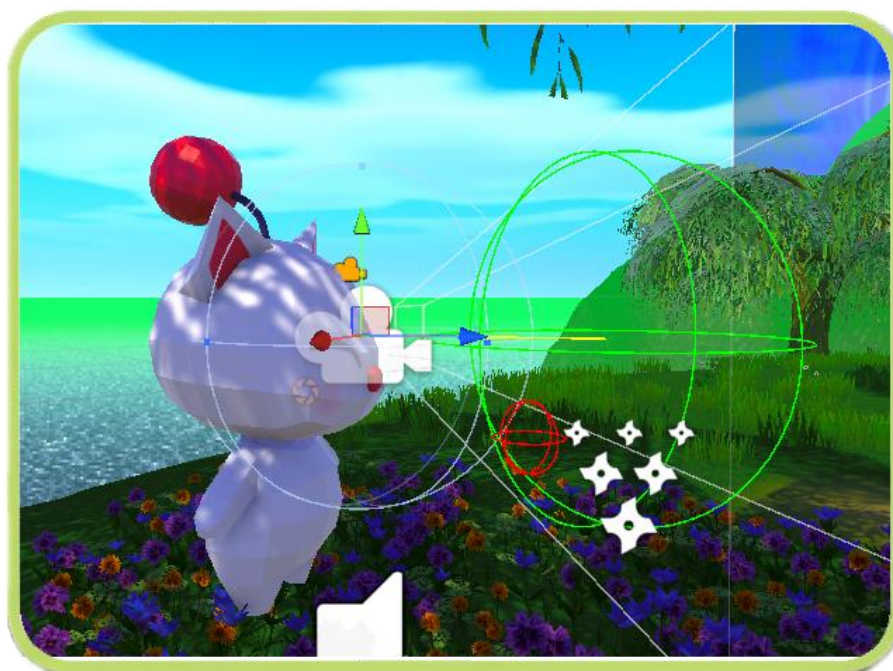


Enquanto o objeto está a ser segurado, acompanha o movimento e a orientação do utilizador. Durante este processo, o mesmo pode ainda ajustar a distância do objeto, através da roda, do rato ou das teclas “I” e “K”, sendo aplicada verificação de colisão para evitar que o objeto atravessasse paredes ou outros elementos do cenário. É igualmente possível rodar o objeto livremente ao manter premido o botão do meio do rato, ficando a câmara estática durante essa ação. Por fim, o objeto pode ser largado com o botão direito do rato ou lançado com força, consoante o tempo de pressão do botão.

2.2.2 – *GrabbableObject.cs* – Definição dos objetos interativos

Este script é atribuído a todos os objetos do cenário que possam ser agarrados. Quando um objeto é interagido, a sua física é temporariamente desativada, passando para o estado kinematic, o que permite que siga suavemente a posição definida em frente ao jogador. Ao ser largado, a física do objeto é novamente ativada, podendo este recuperar o comportamento físico normal e, no caso de um lançamento, receber a velocidade correspondente à ação executada pelo mesmo.

Figura 19 - Mecânica de agarrar atualizada



2.3 — Primeiro Menu e Planeamento do Login

Nesta fase, foi iniciada a criação do primeiro menu do projeto, com o objetivo de estabelecer uma base funcional para a interface do utilizador. Para acelerar este processo, foi integrado um *prefab* de menu denominado *SlimUI*, permitindo poupar tempo na criação de animações e de elementos visuais de interface.

O menu foi utilizado apenas de forma temporária, servindo como estrutura inicial enquanto se planeava uma reformulação completa numa fase posterior do desenvolvimento. Assim, foram mantidas as animações e as posições dos elementos de interface, tendo sido removido o conteúdo visual original e substituído pela lógica própria do projeto. Nesta etapa, foi ainda tomada a decisão de não investir, de imediato, na criação de *assets* próprios para o menu, sendo dada prioridade à funcionalidade e à continuidade do desenvolvimento do projeto.

Figura 20 - Desenvolvimento do menu principal





2.4 — Sistema de Login com Base de Dados

Foi desenvolvido o sistema de autenticação de utilizadores, permitindo a ligação entre o jogo e a base de dados.

2.4.1 – Implementação da Interface de *Login*

Foi criada uma janela de login por completo, composta por dois campos de entrada de texto (*TextMeshPro InputFields*), destinados à introdução das credenciais do utilizador: Utilizador do jogo (*jo_user_jogo*); Senha do jogo (*jo_password_jogo*). Esta interface permite ao utilizador inserir os seus dados de forma simples e intuitiva antes de aceder ao menu principal.

2.4.2 – Funcionamento do Sistema de *Login*

O processo de autenticação é gerido através do script *MenuLoginController.cs*. Ao iniciar o jogo e selecionar a opção *Play*, é apresentado ao utilizador um painel de login com uma animação de abertura.

Após a introdução dos dados e ao clicar na opção *Login*, o sistema envia as credenciais para a base de dados através do módulo de integração. A verificação é realizada na tabela jogador, onde é procurado um registo com correspondência entre o nome de utilizador e a respetiva senha. Caso exista correspondência, o login é considerado válido.

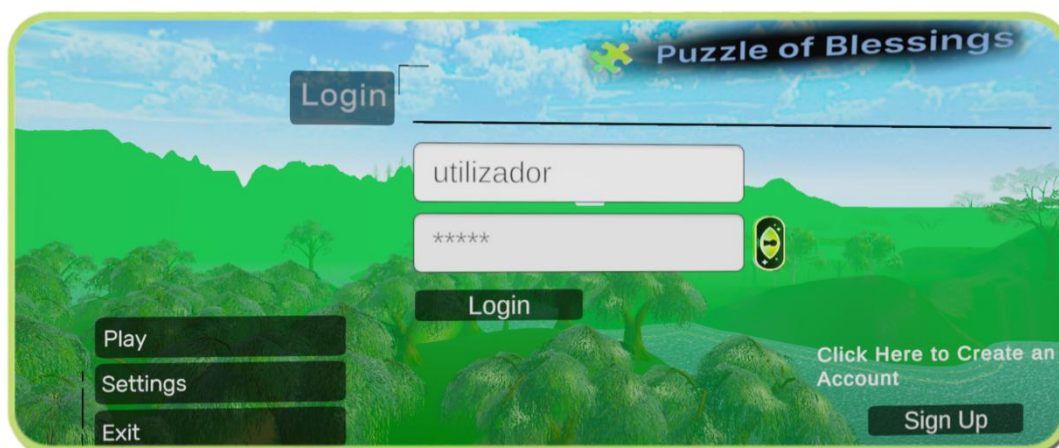
Quando a autenticação é bem-sucedida, os dados do utilizador são armazenados temporariamente em memória através de um gestor de sessão (*GameSessionManager*). Este utiliza o método *DontDestroyOnLoad*, permitindo preservar a informação mesmo durante a transição entre diferentes cenas do jogo. Após o *login*, é apresentado o menu principal, disponibilizando opções como Novo Jogo e Continuar.

Importa referir que o registo de novos utilizadores não é realizado dentro do jogo, sendo efetuado exclusivamente através do website do meu colega. O sistema implementado no jogo permite apenas a autenticação de utilizadores já existentes.

2.4.3 Proteção das palavras-passe através de hashing

Numa fase posterior do desenvolvimento, foi prevista a implementação de um sistema de hashing para as palavras-passe, com vista a reforçar a segurança dos dados dos utilizadores. Esta medida foi pensada como forma de proteção da informação armazenada na base de dados, evitando que as senhas sejam guardadas em texto simples. O hashing consiste na transformação da palavra-passe introduzida pelo utilizador num valor codificado através de um algoritmo criptográfico. Assim, mesmo que exista acesso indevido à base de dados, a palavra-passe original não fica visível de forma direta. No momento do login, o valor introduzido pelo utilizador é novamente convertido e comparado com o hash guardado na base de dados. Se ambos coincidirem, a autenticação é validada.

Figura 21 - Sistema de Login e Hashing



2.5 — Criação do Mapa Principal (Ilhas)

Após a interrupção temporária do trabalho no menu, o desenvolvimento centrou-se novamente no mapa principal do jogo. O mapa foi concebido com base numa estrutura composta por ilhas interligadas, tendo sido definidos três níveis principais para esta fase do projeto: o tutorial na primeira ilha, na segunda ilha o Nível 1, na terceira ilha o Nível 2 e posteriormente nível 3 e fim de jogo.

Para a construção do cenário, recorreu-se às ferramentas de terreno do Unity, através das quais foi possível modelar as ilhas e a zona envolvente de água. Foram também adicionados elementos de vegetação, como árvores e arbustos, utilizando a ferramenta *Paint Trees/Details*.

De forma a orientar a progressão do utilizador, foi igualmente criado um caminho que liga as três ilhas. Durante este processo, surgiram alguns problemas técnicos, nomeadamente erros associados ao pipeline de renderização, devido à incompatibilidade de certos shaders com o *URP*. Verificaram-se também dificuldades relacionadas com o sistema de *LOD (Level of Detail)*, que provocava pop-in visual em alguns modelos. Estes problemas foram resolvidos através da conversão manual dos materiais para *URP/Lit* e da correção de aspetos relacionados com colisões e movimentação da câmara. Como resultado, foi obtido um mapa com uma boa componente visual e ambiental, apoiado pela utilização do *URP*, *fog*, *skybox* personalizada e iluminação ambiente ajustada ao estilo pretendido para o jogo.

Figura 22 - Criação do mapa

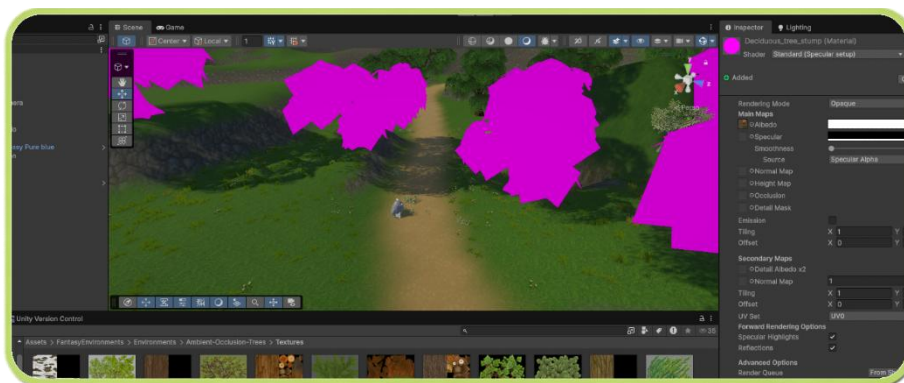


Figura 23 - Visual final do mapa

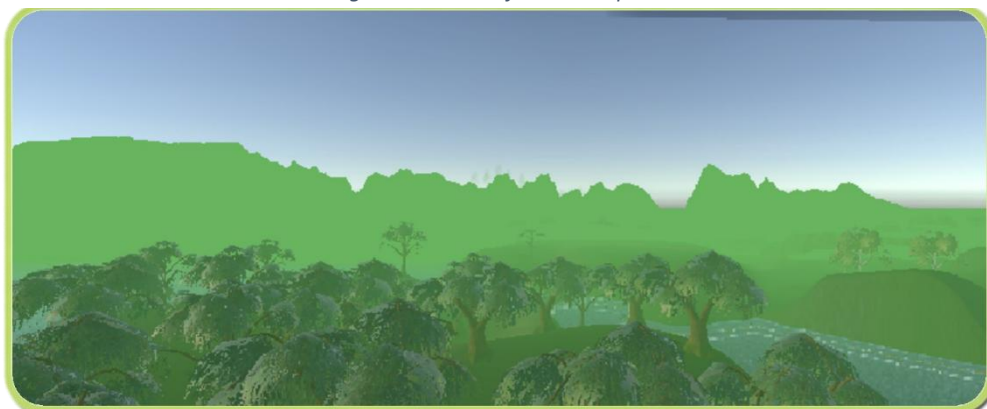


Figura 24 - Planejamento de Níveis



2.6 — Tutorial: Chave e Portão

Tendo em vista a introdução do jogador à mecânica de agarrar e largar objetos e explicar conceitos básicos, foi introduzido um pequeno tutorial inicial antes da progressão para puzzles mais complexos. O tutorial foi estruturado de forma simples e intuitiva, permitindo que se aprendesse a utilizar a mecânica principal do jogo sem necessidade de instruções extensas. Para isso, foi colocada uma chave no cenário, em posição visível e acessível. Ao utilizar a mecânica de agarrar, através da tecla “E”, e consegue-se pegar na chave e transportá-la até ao portão que bloqueia a passagem para a ponte. Quando a chave é aproximada do portão dentro de uma distância mínima definida, é desencadeada automaticamente a animação de abertura. Nesse momento, a chave é removida do cenário, permitindo ao mesmo atravessar a ponte e avançar para a ilha seguinte, concluindo assim o tutorial.

Figura 25 - Desenvolvimento do Tutorial





2.7 — Limitações, *Respawn* por queda em água

Para evitar que o utilizador ficasse preso ao cair na água, foi implementado um sistema de *respawn* automático. Este sistema permite restaurar o jogador para uma posição segura sempre que ocorre uma queda na água, garantindo uma melhor experiência de jogo e evitando interrupções desnecessárias.

2.7.1 – Funcionamento do *WaterRespawn.cs*

Este script é responsável por monitorizar continuamente a posição de quem joga, guardando, em intervalos regulares de 0,2 segundos, as últimas posições consideradas seguras, ou seja, posições em que o jogador se encontra no solo. Quando é detetado contacto com a água, é iniciado um efeito visual de transição, com a tela a escurecer temporariamente. De seguida, este é teleportado para a última posição segura registada e, após essa reposição, a imagem volta gradualmente ao normal. Durante este processo, é também reproduzido um som associado à queda e o movimento é temporariamente desativado, visando tornar a transição mais natural e coerente.



2.8 — Menu em Tempo de Jogo e Sistema de Salvar Progresso

Nesta fase procedeu-se ao desenvolvimento de dois elementos fundamentais para a experiência de utilização: o menu de pausa acessível em jogo através da tecla “ESC” e o sistema de gravação do progresso. Esta etapa assumiu particular relevância no contexto do projeto, uma vez que exigiu a articulação entre a interface do utilizador, a gestão da sessão de jogo e o armazenamento persistente dos dados do jogador.

2.8.1 — Menu e Definições em Jogo

Foi integrado um novo painel de definições, acessível durante a execução do jogo através da tecla **ESC**. A sua criação teve como objetivo disponibilizar ao utilizador um conjunto de opções essenciais sem interromper de forma definitiva a sessão de jogo. Este menu permite ajustar parâmetros relacionados com o áudio, nomeadamente o volume geral e o volume da música, contribuindo para uma maior personalização da experiência de jogo. Foram ainda integradas opções que permitem guardar o progresso e regressar ao menu principal, reforçando a navegabilidade e a funcionalidade da interface em contexto de utilização.

2.8.2 — Sistema de Salvar Jogo

De seguida, foi desenvolvido o sistema de gravação do progresso do utilizador, com base na criação de ficheiros *JSON* associados ao utilizador autenticado. Esta solução foi adotada por permitir uma estrutura de armazenamento simples, organizada e facilmente recuperável. Inicialmente, considerou-se a implementação de um sistema de gravação exclusivamente local. No entanto, essa abordagem foi considerada inadequada, uma vez que não assegurava a continuidade do progresso entre diferentes dispositivos nem garantia total consistência com a base de dados. Assim, optou-se por associar cada ficheiro de gravação ao identificador único do jogador, o `jo_cod`, assegurando a correspondência entre o progresso guardado e o registo respetivo na base de dados. Para garantir a preservação da informação entre cenários distintos, recorreu-se ao *GameSessionManager*, configurado com *DontDestroyOnLoad*, permitindo manter os dados da sessão ativos durante toda a execução do jogo.



2.8.3 - Scripts desenvolvidos para o sistema de gravação

Script FunçãoSaveData.cs Define a estrutura dos dados a guardar, incluindo posição, rotação, cena, nível, utilizador e data/hora.

SaveGameController.cs Responsável por executar a gravação do progresso durante o jogo.

PlayerPositionRestorer.cs Reposiciona a posição do jogador durante o carregamento da cena.

LoadGameController.cs Permite retomar o jogo a partir de um ficheiro de gravação existente.

2.8.4 – Estrutura dos dados guardados em *SaveData.cs*

playerUser — utilizador associado ao ficheiro de gravação

currentScene — última cena acedida

coinsCollected — número de moedas recolhidas

puzzlesSolved — número de puzzles concluídos

lastSaveTime — data e hora do último registo

posX, posY, posZ — posição tridimensional do jogador

rotY — rotação horizontal

playerLevel — nível atual do utilizador

2.8.5 – Funcionamento do *SaveGameController.cs*

Quando o utilizador seleciona a opção de guardar, o script identifica automaticamente o objeto do jogador através da tag *Player*, recolhe a sua posição atual e obtém o nível em execução por intermédio do *GameSessionManager*. Posteriormente, todos estes dados são serializados e gravados num ficheiro JSON associado ao código do utilizador. Concluído o processo, é apresentado um feedback visual temporário, de modo a confirmar ao utilizador que o progresso foi registado com sucesso.

Figura 26 - Protótipo Menu em Jogo



Figura 27 - Sistema de guardar progresso



2.9 — Nível 1: Puzzle de Alavancas

O primeiro puzzle do Nível 1 foi concebido, localizado na segunda ilha do mapa. O espaço foi delimitado com muros obtidos através da *Unity Asset Store*, enquanto a alavanca utilizada corresponde a um modelo 3D gratuito da Sketchfab, posteriormente ajustado no Blender e no Unity. A porta do nível foi animada através do *Animator* do Unity.

O objetivo deste nível consistiu em introduzir um desafio lógico, no qual se é desafiado e tem de descobrir a combinação correta de quatro alavancas distribuídas pela ilha. Cada alavanca pode assumir dois estados, Cima ou Baixo, sendo necessário testar diferentes combinações até encontrar a solução adequada. Este puzzle reforça a componente exploratória e experimental do jogo, incentivando o jogador a observar, testar e deduzir a solução correta, ao mesmo tempo que contribui para uma progressão gradual da dificuldade.

O script *LeverInteraction.cs* foi aplicado a cada alavanca e é responsável por detetar a proximidade do utilizador, e alternar o estado da alavanca ao premir a tecla “E”. Sempre que ocorre essa alteração, é reproduzido um som e a respetiva animação associada ao movimento da alavanca.

Por sua vez, o script *LeverPuzzleController.cs*, colocado na porta ou num objeto de controlo, monitoriza continuamente o estado das quatro alavancas e compara-o com a combinação definida como correta. Quando essa combinação é alcançada, é acionada a abertura da porta, é reproduzido o som correspondente e o progresso é atualizado, sendo também efetuado o auto-save do estado do jogo.

Figura 28 - Desenvolvimento Primeiro Nível



2.10 — Áudio e Animações

Procedeu-se à integração dos elementos de áudio e ao refinamento das animações associadas às principais interações do jogo, com o objetivo de melhorar a imersão e fornecer um feedback mais claro ao jogador.

Foram aplicados efeitos sonoros livres de direitos de autor a diversos elementos interativos, incluindo o som de acionamento das alavancas, a abertura de portas, a interação com a chave no tutorial e a queda na água.

Paralelamente, foram desenvolvidas e ajustadas animações no *Animator* do Unity, nomeadamente a abertura do portão do tutorial (associada ao *trigger OpenGate*), a abertura da porta do Nível 1 (associada ao *trigger OpenDoor*) e as animações das alavancas, controladas pelos *triggers PullUp* e *PullDown*, sem repetição em ciclo. Estas animações permitem uma representação visual coerente das interações realizadas no jogo.

Figura 29 - Funcionamento da Alavanca



2.10.1 – Funcionamento do *AudioSettings.cs*

Este script é responsável pela gestão dos volumes geral e da música através de sliders integrados na interface de definições. Os valores selecionados pelo utilizador são guardados localmente através de *PlayerPrefs*, permitindo que as preferências de áudio sejam mantidas entre sessões de jogo sem necessidade de recorrer à base de dados.

2.11 — Reformulação Completa do Menu

Foi realizada a reformulação integral do sistema de menu do jogo, substituindo completamente a solução inicial baseada no *prefab SlimUI* por uma interface original e adaptada às necessidades do projeto. Esta intervenção representou uma das alterações mais significativas ao nível visual e funcional.

A nível gráfico, foram desenvolvidos novos elementos de interface com recurso ao Google Flow e editado no Pixlr, permitindo criar uma identidade visual própria e mais coerente com o estilo do jogo. Paralelamente, a lógica associada ao menu foi reestruturada, tendo sido reescritos os scripts de interface, mantendo apenas como referência algumas animações e posicionamentos previamente existentes.

Foram também introduzidas novas funcionalidades, nomeadamente a integração completa do sistema de login com a base de dados, a implementação das opções Novo Jogo e Continuar, bem como um painel de definições com controlo de volume geral e da música através de sliders. A opção Novo Jogo permite iniciar uma nova sessão, eliminando o progresso anterior, enquanto a opção Continuar apenas se encontra disponível quando existe um ficheiro de gravação associado ao utilizador autenticado.

Adicionalmente, foi incluída uma opção de registo (*Sign Up*) no ecrã de login, que redireciona o utilizador para o website externo, onde é possível criar uma nova conta. Foram ainda implementadas animações de transição entre os diferentes painéis, contribuindo para uma navegação mais fluida.

Figura 30 - Tela de Configurações de Jogo

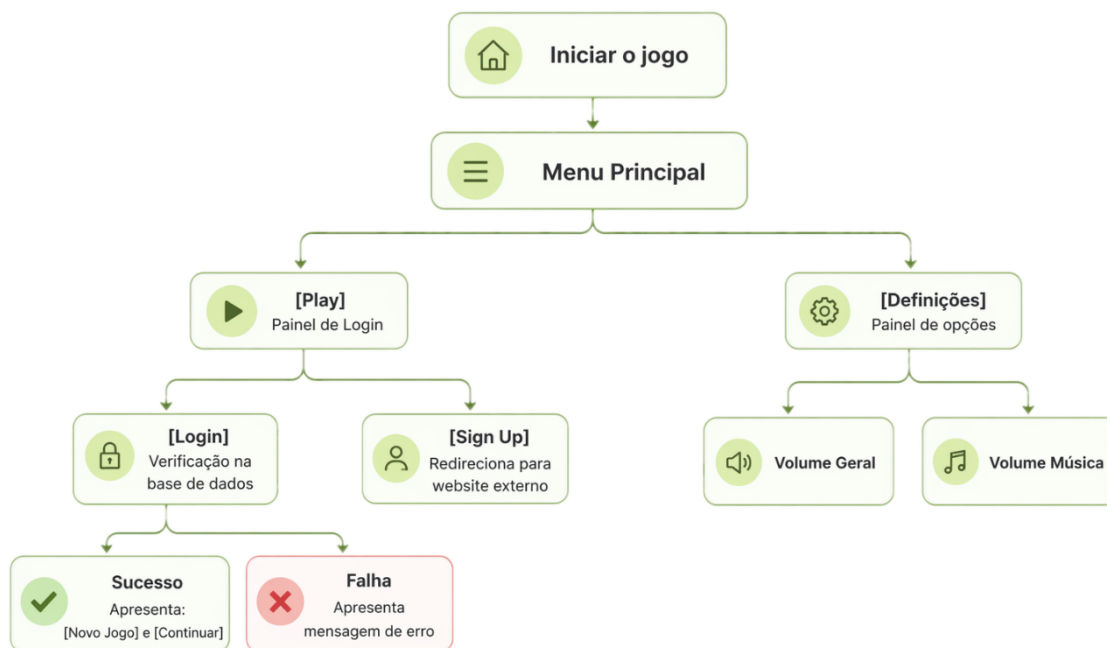


Por fim, procedeu-se à reorganização da estrutura de pastas do projeto no Unity, separando adequadamente scripts, assets e prefabs, visando melhorar a organização e manutenção do projeto.

2.11.1 – Fluxo de funcionamento do menu

De forma a compreender a navegação e as interações disponíveis para o utilizador, foi desenvolvido um diagrama de fluxo que ilustra a estrutura do menu do jogo. Este esquema detalha o percurso desde a inicialização até às diferentes secções principais, como o painel de autenticação e as configurações de áudio, conforme se pode observar na figura seguinte:

Figura 31 - Sistema de Funcionamento do Menu



A opção “Continuar” apenas se encontra disponível quando existe um ficheiro de gravação associado ao utilizador autenticado.

2.12 — Implementação da Personagem Guia Interativa (Fantasma)

Durante esta fase foi desenvolvida a personagem guia do jogo, representada por um fantasma interativo, com a função de orientar o jogador ao longo da progressão da história e conduzi-lo até ao nível seguinte. Para a implementação desta funcionalidade foi utilizado um modelo 3D obtido através da *Sketchfab*, posteriormente integrado e adaptado ao ambiente do jogo.

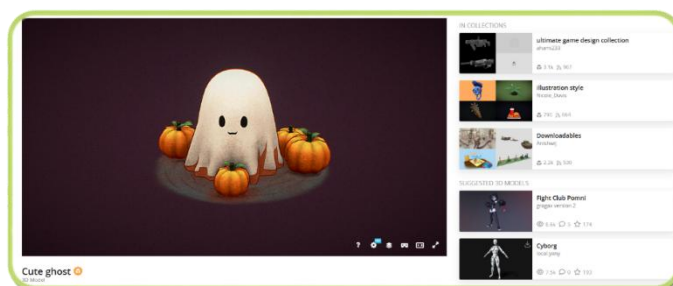
(License: CC Attribution).

A criação desta personagem envolveu desafios técnicos específicos, uma vez que foi necessário adaptar diferentes sistemas de interação e comportamento. Foi implementado um efeito de flutuação para reforçar a sensação de levitação da personagem e melhorar a sua integração visual no ambiente do jogo.

Adicionalmente, foi criado um sistema de som associado ao fantasma, permitindo reproduzir efeitos sonoros durante os diálogos e interações, tornando a experiência mais imersiva e natural.

Foi também criada uma lógica personalizada para a orientação da personagem, garantindo que esta acompanhasse corretamente o utilizador sem apresentar rotações incorretas do modelo 3D. Paralelamente, foram adicionadas funcionalidades ao sistema de diálogos, permitindo associar determinados eventos a falas específicas, incluindo o aparecimento programado de objetos e itens durante a interação com o NPC. Esta fase contribuiu de forma significativa para a narrativa, ambiente e experiência geral do jogo, tornando a progressão mais guiada e interativa.

Figura 32 - Modelo Fantasma



Fonte da Imagem: [Cute Ghost Sketchfab](#)

2.13 — Pop-ups em World Space

Procedeu-se à implementação de elementos de *feedback* visual no ambiente tridimensional do jogo, com especial foco na criação de um sistema de ajuda adaptado ao público-alvo.

Foram desenvolvidos pop-ups recorrendo a Canvas configurados em modo *World Space*, permitindo que estes elementos sejam apresentados diretamente no espaço 3D e integrados no cenário do jogo. Estes *pop-ups* surgem em momentos específicos, como a introdução de novas mecânicas ou a interação com determinados objetos, funcionando como ferramenta de apoio.

Para além do *feedback* visual básico, estes elementos foram também utilizados como *pop-ups* de ajuda, apresentando instruções simples, diretas e de fácil compreensão. Esta abordagem permite guiar de forma intuitiva, sem interromper o fluxo de jogo com menus complexos ou explicações extensas.

Cada pop-up inclui texto informativo e uma animação de escala, na qual o elemento surge progressivamente (aumentando de tamanho) e desaparece após um curto período de tempo. Esta animação foi implementada através do script *PopupAnimation.cs*, contribuindo para uma apresentação mais dinâmica e apelativa.

A utilização deste sistema revelou-se particularmente adequada para o público-alvo definido, uma vez que facilita a aprendizagem das mecânicas do jogo de forma visual e interativa, promovendo uma experiência mais acessível, orientada e envolvente.

Figura 33 - Tutoriais



2.14 — Sistema de *Puzzle Coins*

Foi também desenvolvido um sistema de moedas, designado por *Puzzle Coins*, com o objetivo de incentivar o percurso e introduzir um elemento adicional de progressão no jogo.

As *Puzzle Coins* correspondem a moedas distribuídas pelo cenário, que podem ser recolhidas durante a exploração. Cada moeda recolhida é registada e associada ao utilizador, sendo posteriormente utilizada como contador de progresso, armazenado na base de dados.

Do ponto de vista visual, foi criado um modelo 3D próprio para este elemento, inspirado na identidade do jogo, utilizando como base a forma do logótipo. O objeto foi complementado com um material de aparência dourada, de forma a torná-lo visualmente apelativo e facilmente identificável no ambiente de jogo.

Paralelamente, foi concebido um elemento de interface (HUD) que apresenta o número de moedas recolhidas. Este contador é atualizado dinamicamente sempre que uma nova moeda é apanhada, incluindo um efeito visual de incremento, de forma a reforçar o *feedback* da ação realizada. A integração deste sistema contribui para aumentar o envolvimento de quem joga, promovendo a exploração ativa do cenário e fornecendo um indicador claro de progresso ao longo do jogo.

Figura 34 - *Puzzle Coin*



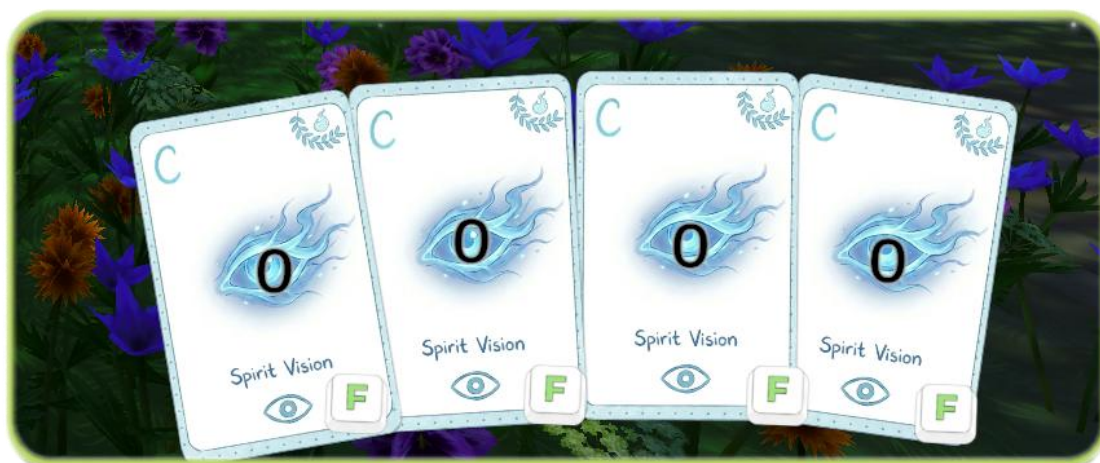
2.15 — Blessing Manager: Sistema de Habilidades

Com vista a centralizar a gestão das habilidades do jogador, foi desenvolvido para a gestão das *blessings*, ou seja, das habilidades que podem ser desbloqueadas e utilizadas ao longo do jogo. O objetivo principal passou por tornar este sistema mais organizado, mais fácil de expandir e com ligação direta à base de dados, para que as habilidades ficassem associadas ao progresso de cada utilizador.

Em vez de cada habilidade funcionar de forma isolada, passou a existir um controlo centralizado, o que facilitou a gestão do sistema e a comunicação com outros elementos do projeto.

O funcionamento destas habilidades passou também a incluir uma mecânica de *cooldown*. Isto significa que, depois de uma habilidade ser utilizada, ela não pode ser ativada novamente de imediato, tendo de aguardar um determinado tempo até voltar a estar disponível. Esta abordagem foi importante para equilibrar a jogabilidade e para evitar que as habilidades fossem usadas de forma contínua e sem restrições. Sempre que ativar uma blessing, a respetiva carta na interface entra em estado de recarga, mudando visualmente para indicar que a habilidade está temporariamente indisponível. Durante esse período, o jogador consegue perceber claramente que terá de esperar alguns segundos até poder voltar a usá-la.

Figura 35 - Sistema de Cartas





2.15.1 – Funcionamento *BlessingManager*

O script *BlessingManager.cs* é o principal responsável por todo este sistema. De forma resumida, este script verifica quais as blessings que o utilizador já desbloqueou, organiza-as na interface e controla quando podem ser ativadas. Quando uma habilidade é usada, o script trata de atualizar o seu estado, iniciar o cooldown e alterar a aparência da carta para que se tenha *feedback* visual imediato. Este comportamento pode ser observado no excerto de código apresentado, onde se percebe a ligação entre a lógica do sistema e a interface do jogo. Além disso, foi introduzido um sistema de reposição total do progresso no modo Novo Jogo, garantindo que, ao iniciar uma nova sessão, sejam removidos os dados guardados anteriormente, incluindo as habilidades desbloqueadas, o progresso e outros elementos associados ao jogador. Este sistema estabeleceu a base técnica para toda a árvore de habilidades do jogo, mais claro para o utilizador e mais preparado para futuras expansões do projeto.



2.16 — Artefactos Colecionáveis

O sistema de artefactos colecionáveis foi construído para incentivar a exploração do mapa e aumentar a interação com o ambiente do jogo.

Para esta funcionalidade foi criado o script *CollectibleItem.cs*, responsável por controlar o comportamento dos colecionáveis no cenário. Cada item possui um identificador próprio (*cl_id*), definido no Inspector do Unity, permitindo associá-lo corretamente à base de dados.

Os colecionáveis apresentam um efeito contínuo de rotação e flutuação, tornando-os mais visíveis e apelativos. Quando o jogador recolhe um item, o sistema envia automaticamente os dados para o Supabase, registando o colecionável associado ao utilizador.

Este sistema foi também integrado com a opção Novo Jogo, garantindo que todos os colecionáveis anteriormente obtidos são removidos da base de dados quando se reinicia completamente o progresso.

2.17 — *Memory Teleporter*

Foi implementada a mecânica *Memory Teleporter*, baseada na interação entre dois objetos específicos necessários para ativar o teletransporte. O sistema foi implementado através do script *MemoryTeleporter.cs*, responsável por verificar a ligação entre os objetos e executar a transição para a nova localização.

Quando o puzzle é concluído, é iniciado um efeito visual de transição com um flash branco no ecrã e o jogador é automaticamente teleportado para outra zona do mapa. Durante este processo, a física dos objetos envolvidos é temporariamente desativada e o sistema de áudio ajusta automaticamente o volume dos sons do ambiente para reforçar o efeito da transição. Paralelamente, foram corrigidos problemas relacionados com os objetos da visão especial (*SpecialVisionObjects*). Sempre que uma habilidade especial termina enquanto se segura um objeto invisível ou mágico, o sistema força automaticamente o objeto a ser largado, evitando erros de gameplay e situações de exploração indevida das mecânicas do jogo.



2.18 — Mecânica de Voo Livre (*NoClip*)

A habilidade *NoClip* foi implementada como uma ferramenta reservada a administradores, destinada a facilitar o desenvolvimento, os testes e as apresentações do projeto. Esta bênção permite ao utilizador atravessar a geometria do nível e movimentar-se livremente pelo mapa, sem as limitações normais da física.

A funcionalidade foi integrada de forma modular no sistema de blessings, através da interface *IBlessingAbility*, ficando compatível com o *BlessingManager*. O script *NoClip.cs* desativa temporariamente os controlos normais de movimento, a gravidade e os colliders do utilizador, permitindo a passagem por paredes, chão e outros objetos do cenário. A movimentação é feita de forma livre, tendo a câmara como referência, o que facilita a exploração rápida do mapa. Foram ainda adicionados controlos para subir, descer e aumentar a velocidade de deslocação, tornando esta ferramenta útil para verificar zonas do jogo e apresentar o projeto de forma mais eficiente.

Quando a habilidade é desativada, o sistema repõe os controlos normais do jogador e reativa a física e os colliders anteriormente desativados, garantindo o regresso ao comportamento habitual da personagem.



2.19 — Puzzle “Simon Says” (*Memory Spheres*)

Um dos desafios técnicos mais exigentes do projeto foi o puzzle de memória, inspirado no clássico “Simon Says”. Esta mecânica representou um dos maiores desafios técnicos do projeto, uma vez que exigiu a combinação de geração automática de objetos, som criado por código, lógica de sequência, interface de apoio e integração com a base de dados.

2.19.1 – Geração do Puzzle

Foi criado o script `MemorySpherePuzzle.cs`, responsável por montar o puzzle sempre que a cena é carregada. O script gera cinco esferas, posicionando-as em círculo e orientando-as para o jogador.

2.19.2 – Funcionamento do Puzzle

A mecânica segue a lógica clássica de um jogo de memória. O puzzle começa com uma sequência curta, com apenas uma cor, e vai aumentando progressivamente até cinco cores.

Durante cada ronda, o jogo mostra a sequência correta através da ativação das esferas, e o objetivo é repetir essa mesma ordem. A interação é feita através da tecla “E” ou do clique esquerdo do rato, com deteção por raio.

Quando se acerta, a esfera reage com uma pequena animação e com um som de confirmação. Se errar, o puzzle emite um som de falha e reinicia a sequência, obrigando o utilizador a recomeçar.

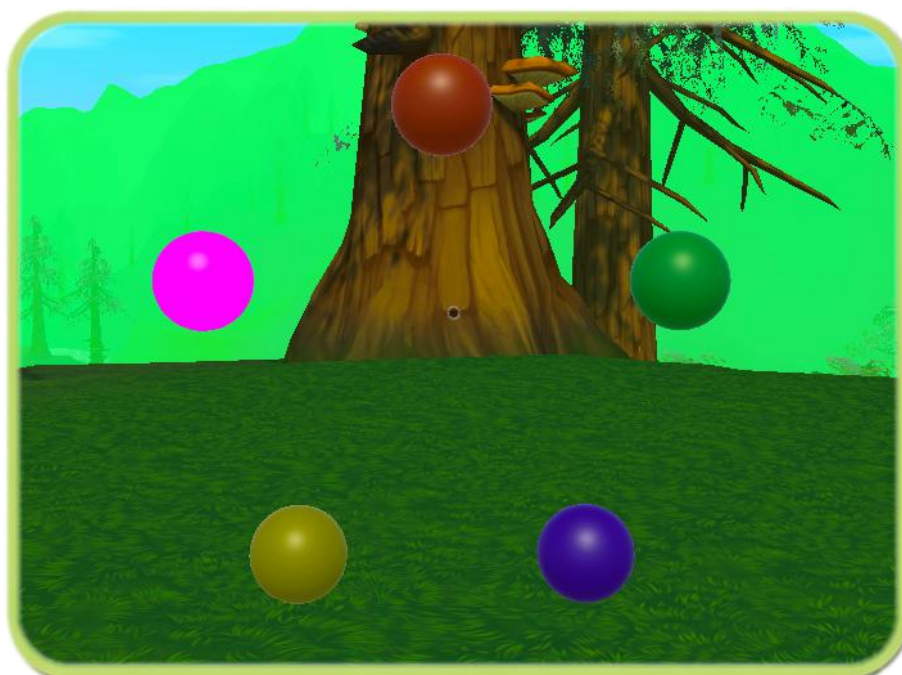
Para equilibrar a dificuldade, foi ainda introduzido um sistema de piedade, pensado para evitar frustração excessiva. Se o jogador falhar várias vezes no mesmo puzzle, o sistema passa a ser mais tolerante e pode impedir que certas penalizações mais exigentes sejam ativadas de imediato. Desta forma, o desafio mantém-se difícil, mas sem se tornar demasiado frustrante.



2.19.3 – Integração com a base de dados

Por fim, o puzzle foi integrado com o Supabase, permitindo verificar se o jogador já tinha desbloqueado a recompensa associada a este desafio. Caso a recompensa já tivesse sido obtida anteriormente, o puzzle não era criado novamente, evitando desperdício de recursos e garantindo que o progresso era respeitado.

Figura 36 - Desafio Simon Says





2.20 — Power-up de Velocidade (*Speedup*)

Para enriquecer a experiência de movimento, foi desenvolvido um *power-up* de velocidade, através do script `SpeedupPowerup.cs`, que ao ser recolhido altera temporariamente a movimentação do utilizador. Quando este entra em contacto com o *power-up*, a sua velocidade passa a ser superior ao normal e o movimento frontal fica automaticamente reforçado, criando um efeito de aceleração contínua.

Ao mesmo tempo, o salto é desativado durante o efeito, com o intuito de manter o controlo da mecânica e evitar conflitos com a deslocação rápida. Para tornar a experiência mais apelativa, o objeto foi ainda animado com um efeito de flutuação e rotação, e foi adicionado feedback visual e sonoro no momento da recolha. Após alguns segundos, o efeito termina e a movimentação volta ao comportamento normal.

Figura 37 - Powerup de Velocidade



2.21 — Fase Final do Puzzle Of Blessings

Antes do end game, foi criada uma floresta que tenta criar um clima mais sombrio e noturno através da alteração repentina do ambiente, com vista a preparar a atenção do jogador para a reta final do jogo. Esta parte exigiu vários ajustes ao nível da jogabilidade, da orientação do mesmo e da integração das mecânicas já existentes, uma vez que era necessário garantir que o percurso final fosse desafiante, mas ao mesmo tempo claro e funcional.

Nesta zona, encontram-se armadilhas que o obrigam a carregar numa sequência de teclas apresentada no ecrã através de uma imagem central. Embora não exista uma penalização imediata por falhar, fica-se temporariamente mais lento, sendo a forma de ultrapassar esta dificuldade o uso da habilidade *Spirit Vision*. A implementação desta zona exigiu bastante afinação, porque foi necessário equilibrar a dificuldade da mecânica com a capacidade de perceber e superar sem frustração.

Figura 38 - Desafio Floresta Assombrada



O utilizador passa a receber indicações visuais do caminho através de setas posicionadas, por forma a orientar a progressão até à zona seguinte. Depois de concluir o puzzle Simon Says e obter a bênção de duplicação, o mesmo volta a falar com o fantasma na zona seguinte, que lhe entrega uma nova blessing de apoio chamada Spirit Anchor.

A partir desse ponto, o jogador é obrigado a usar as duas blessings, Duplication e Spirit Anchor, para recolher duas tábuas perto da casa do fantasma. Com essas tábuas, este constrói uma ponte para alcançar o outro lado, chegando assim à ilha final do jogo. No final desse percurso, sobe-se até à zona escura na montanha, desencadeando a sequência de créditos. Esta sequência final exigiu um esforço de integração considerável, porque foi necessário garantir que todas as mecânicas funcionassem em conjunto sem falhas, especialmente numa fase em que o utilizador já se encontra muito perto de concluir o jogo.

Figura 39 - Desafio Final Spirit Anchor e Duplication





2.21.1 – Transição Narrativa (*EndgameTrigger*)

O *EndgameTrigger.cs* foi criado para marcar o momento final da aventura. Ao entrar neste gatilho, o jogo ativa uma transição mais cinematográfica, desligando o movimento de quem joga e os sons do cenário, para que possa destacar o momento de conclusão.

Durante esta sequência, o ecrã escurece e surgem várias frases narrativas no centro, com efeito visual de aparecimento e desaparecimento suave. Esta sincronização revelou-se particularmente exigente, dado que qualquer desfasamento comprometeria a imersão.

No final da sequência, o sistema comunica com a base de dados para atualizar o estado de conclusão e, depois disso, regressa ao menu principal.

2.21.2 – Persistência e *New Game+*

Foi também adicionado o campo booleano *NewGame+* na tabela *player* da base de dados Supabase. Este campo permite identificar os jogadores que já concluíram o jogo e que, por isso, podem aceder a uma nova opção no menu inicial.

No *MenuLoginController.cs*, quando o sistema deteta que o jogador já desbloqueou este estado, passa a aparecer o botão *New Game+* acima da opção *Continuar*.

Ao iniciar este modo, o jogo reinicia apenas o progresso principal, como a posição de save e o nível atual, mas mantém os elementos já desbloqueados, e as blessings. Desta forma, este pode voltar a jogar desde o início, mas com acesso às habilidades conquistadas anteriormente, tornando a segunda passagem pelo jogo mais livre e criativa.

2.22 — Arquitetura de Distribuição e Launcher Personalizado

Nesta fase final, o projeto deixou de se focar apenas no conteúdo desenvolvido dentro do Unity e passou também a considerar a forma como o jogo seria entregue ao utilizador. Para isso, foi criado um launcher próprio, com o objetivo de facilitar o acesso ao jogo e permitir futuras atualizações sem necessidade de descarregar manualmente novos ficheiros.

2.22.1 – Launcher Desenvolvido em WPF

O launcher foi criado fora do Unity, utilizando WPF (Windows Presentation Foundation) e .NET 8. O WPF é uma tecnologia da Microsoft usada para criar janelas e interfaces no Windows, e neste projeto serviu para construir a aplicação de arranque do jogo.

A interface foi feita em XAML, que é a linguagem usada para desenhar os elementos visuais da aplicação, como botões, textos, barras de progresso e menus. Assim, foi possível criar um launcher com aspeto mais profissional e adaptado à identidade visual do jogo.

A lógica da aplicação foi desenvolvida de forma assíncrona, por forma a que o programa não ficasse bloqueado enquanto faz downloads ou verificações. Isto permite que o utilizador continue a ver a interface a funcionar normalmente enquanto o launcher trata das tarefas em segundo plano.

2.22.2 – Integração com Supabase

O launcher foi ligado ao Supabase, que funciona como base de dados online. A partir daí, a aplicação consegue verificar se existe uma nova versão do jogo e saber onde está guardado o ficheiro de download.

Desta forma, sempre que o jogo é atualizado, o launcher consulta a base de dados e descarrega automaticamente a versão mais recente, evitando que o utilizador tenha de procurar ficheiros novos manualmente.

2.22.3 – *Download* e extração dos ficheiros

Durante esta fase também foi necessário resolver problemas relacionados com o download dos ficheiros. Alguns serviços de alojamento, como o Dropbox, nem sempre devolvem o ficheiro diretamente. Para contornar isso, o launcher foi preparado para reconhecer esse tipo de ligação e ajustá-la automaticamente.

Depois do download, o sistema verifica se o ficheiro recebido está válido e só depois procede à extração. Isto ajuda a evitar erros e garante que o conteúdo é colocado corretamente no computador do utilizador.

2.22.4 – Tratamento de erros e experiência do utilizador

Em caso de erro, o launcher foi programado para mostrar mensagens simples e diretas dentro da própria janela, em vez de apresentar avisos técnicos do sistema operativo. O objetivo foi tornar a experiência mais clara e menos confusa para o utilizador.

Quando ocorre algum problema, o botão de jogar é escondido, a barra de progresso muda de cor e é apresentada uma mensagem a informar que o download não está disponível no momento.

2.22.5 – Instalação do *Launcher*

Para finalizar, o launcher foi publicado como uma aplicação autónoma, visando qualquer utilizador de Windows o possa instalar sem precisar de ferramentas adicionais de programação. Depois disso, foi criado um instalador com o Inno Setup, permitindo distribuir o programa de forma simples, com atalho no ambiente de trabalho e instalação automática.



2.23.1 — Sistema de Auto-Update do Launcher

Após o launcher estar funcional, surgiu a necessidade de implementar um sistema de atualização automática, com o intuito de também ele pudesse ser melhorado no futuro sem obrigar o utilizador a reinstalar tudo de novo. Esta solução foi pensada para dar mais escalabilidade ao projeto e tornar a manutenção mais simples.

2.23.2 – Controle de versões Supabase

Foi criada uma tabela na base de dados chamada `launcher_versions`, onde ficam registadas as versões disponíveis do launcher, juntamente com o link de download. Assim, o programa consegue saber qual é a versão mais recente e comparar com a versão instalada localmente.

2.23.3 – Verificação automática ao iniciar

Sempre que o launcher abre, ele primeiro verifica se existe uma nova versão disponível na base de dados. Se existir, faz o download da atualização antes de continuar para o jogo. Desta forma, o utilizador está sempre a usar a versão mais recente.

2.23.4 – Substituição dos ficheiros em uso

Um dos maiores desafios desta fase foi atualizar um ficheiro que já está a ser executado pelo Windows. Como o sistema operativo não permite substituir diretamente um `.exe` aberto, foi criada uma solução que renomeia os ficheiros antigos e coloca os novos no lugar certo. Depois disso, o launcher reinicia automaticamente com a nova versão. Esta abordagem evitou o uso de ficheiros auxiliares externos e tornou o processo mais limpo e mais estável.

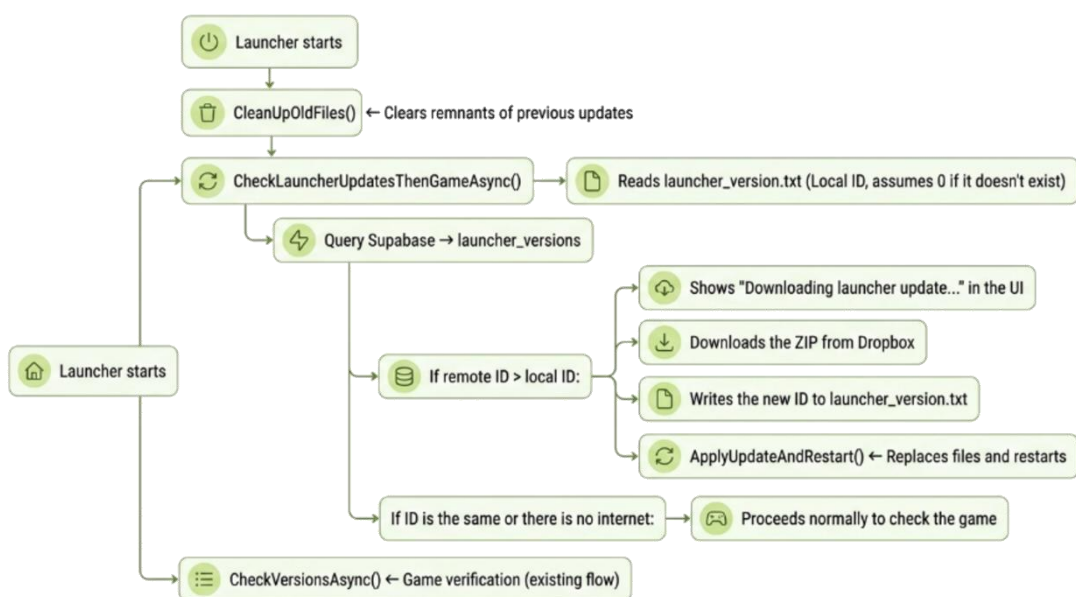
2.23.5 – Registo de Diagnóstico

Para facilitar a deteção de erros, foi criado um ficheiro de registo interno com os passos executados durante a atualização. Assim, caso algo falhe, é mais fácil perceber onde aconteceu o problema.

2.23.6 – Permissões de Administrador

Como o launcher é instalado numa pasta protegida do Windows, foi necessário pedir permissões de administrador sempre que ele é executado. Isto garante que o programa consegue atualizar ficheiros sem bloqueios de acesso.

Figura 40 - Estrutura de Funcionamento do Launcher



3 – MANUAL DO UTILIZADOR

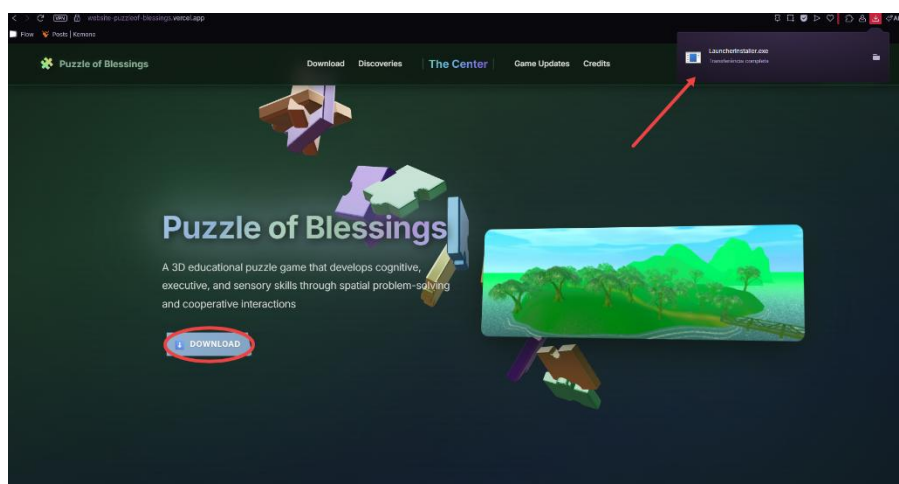
3.1 – Instalação do Jogo

A instalação do Puzzle of Blessings realiza-se em duas etapas: o *download* do executável de instalação do *Launcher* a partir do *website* oficial, seguido da instalação do próprio jogo através do *Launcher*.

3.1.1 – Etapa 1

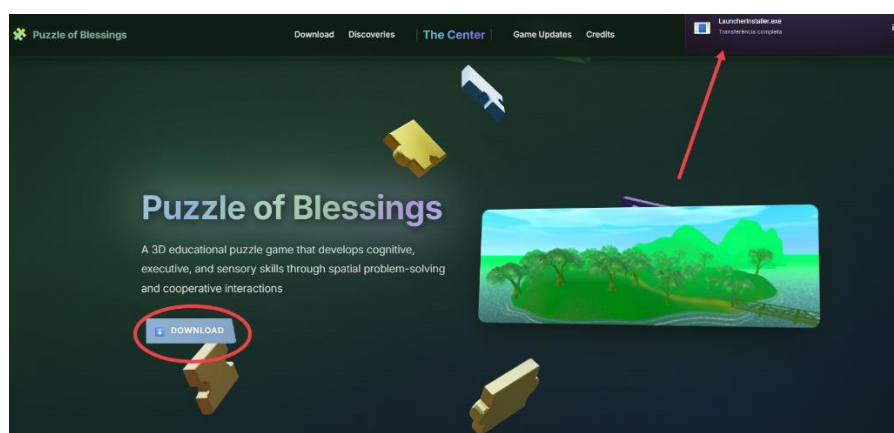
Aceda ao website oficial do jogo em website-puzzleof-blessings.vercel.app e clique no botão **DOWNLOAD** para descarregar o ficheiro LauncherInstaller.exe.

Figura 41 - Download no Site



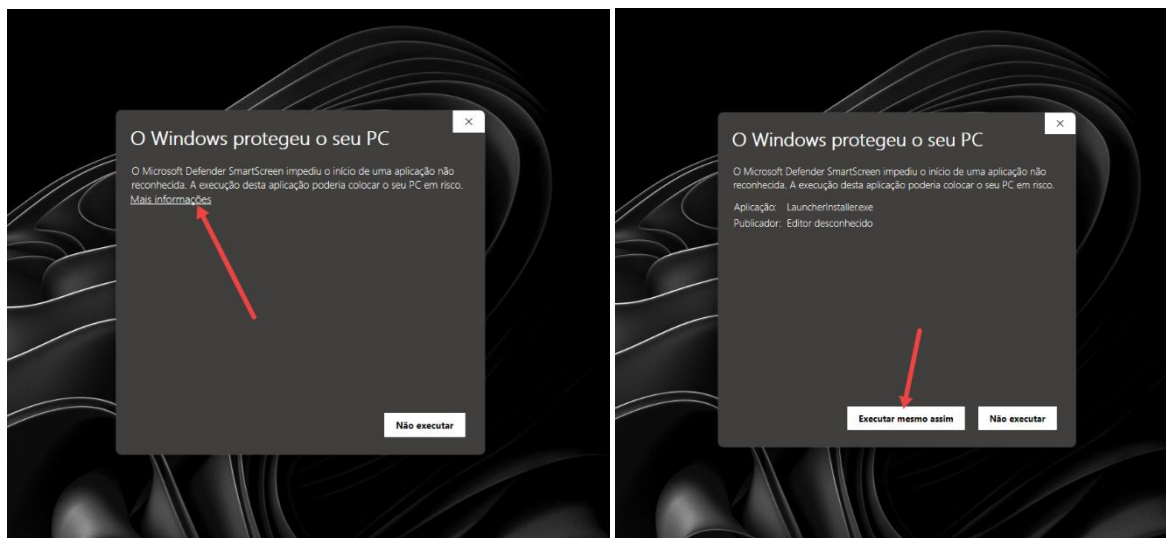
Após concluído o descarregamento, abra o ficheiro LauncherInstaller.exe.

Figura 42 - Abrir Executável



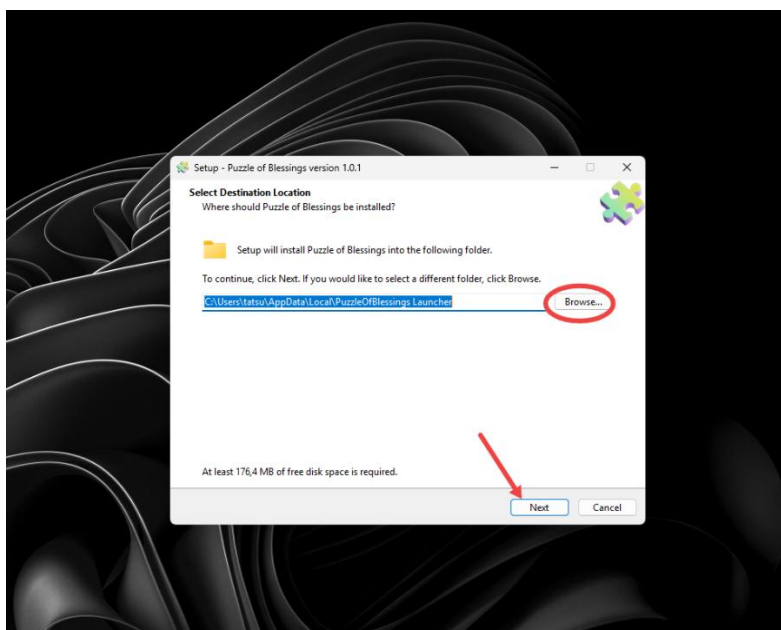
É provável que o *Windows Defender SmartScreen* apresente um aviso de segurança, uma vez que o instalador não possui uma assinatura digital certificada. Clique em "**Mais informações**" na janela de aviso do *SmartScreen*.

Figura 43 - Smartscreen Windows



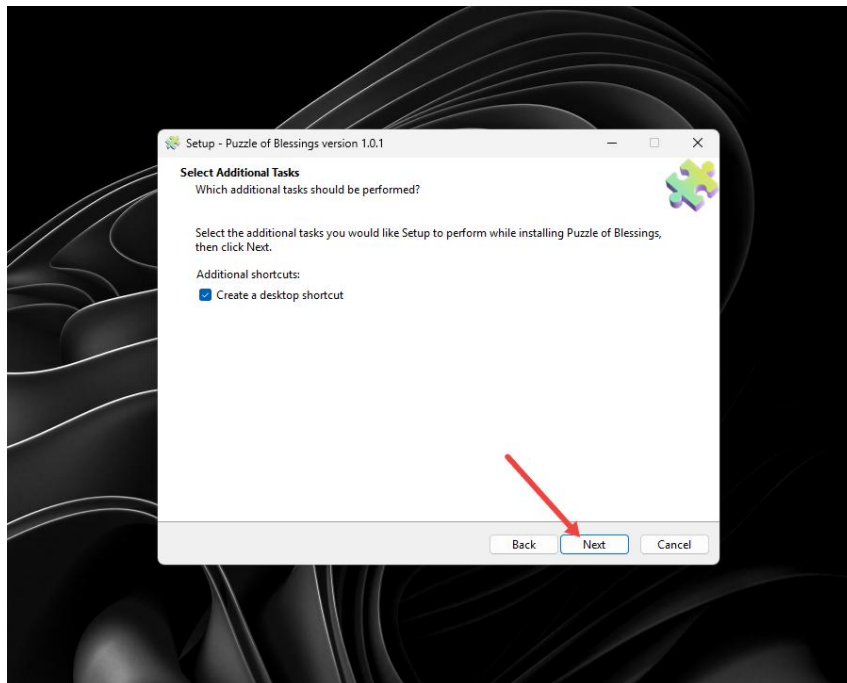
Clique em "**Executar mesmo assim**" para continuar a instalação. O assistente de instalação orienta o utilizador através de alguns passos simples. Selecione a pasta de destino para o Launcher (pode manter a localização predefinida) e clique em **Next**.

Figura 44 – Instalador



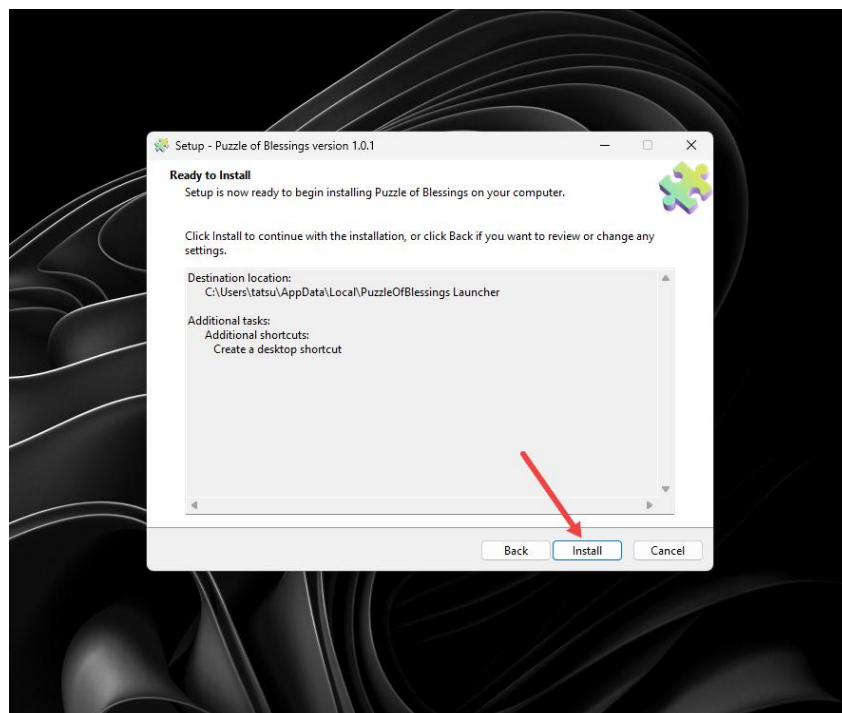
Confirme a criação de um atalho no Ambiente de Trabalho e clique em **Next**.

Figura 45 - Criar Executável no Ambiente de Trabalho



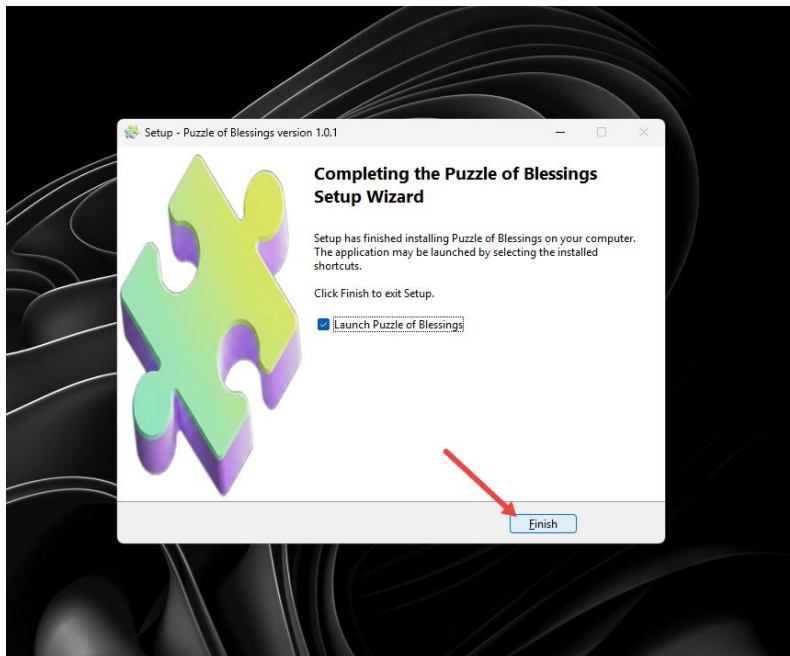
Reveja as opções selecionadas e clique em **Install** para iniciar a instalação

Figura 46 – Instalar



Quando a instalação estiver concluída, clique em **Finish**. O Launcher iniciará automaticamente.

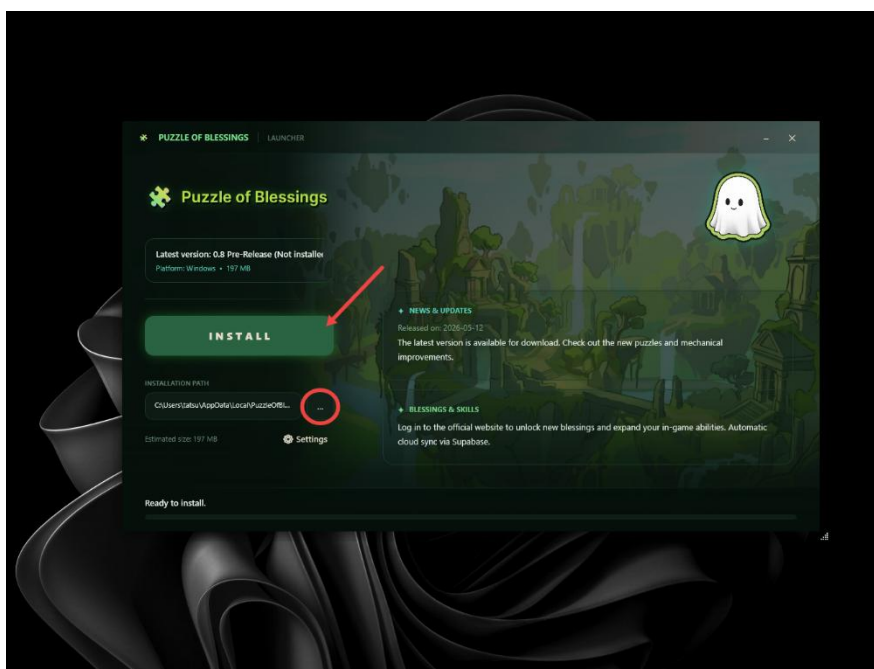
Figura 47 - Abrir Launcher



3.1.2 – Etapa 2

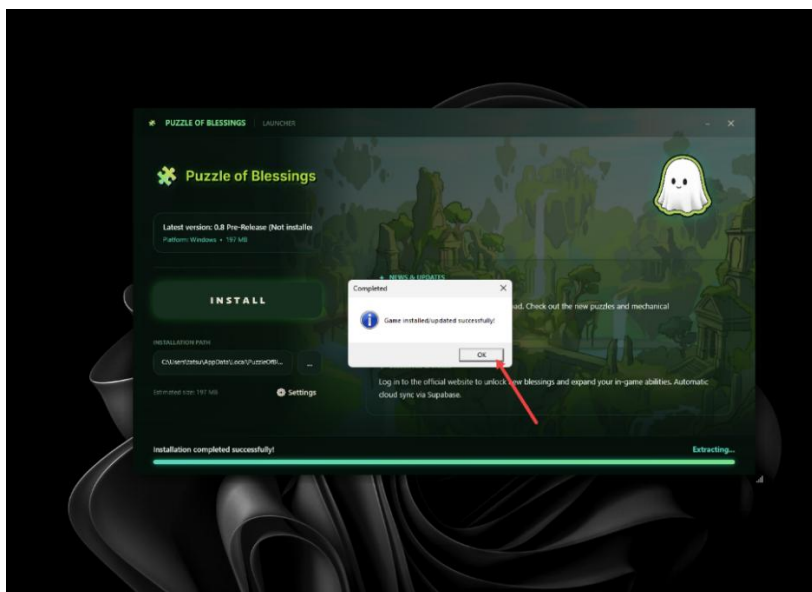
Após a abertura do Launcher, este verifica automaticamente se existe uma versão do jogo instalada. Se ainda não estiver instalado, será apresentado o botão **INSTALL**.

Figura 48 - Instalar Jogo



Quando a instalação estiver concluída, surge uma notificação de sucesso. Clique em **OK**, o botão **INSTALL** será substituído pelo botão **PLAY**.

Figura 49 – Confirmar

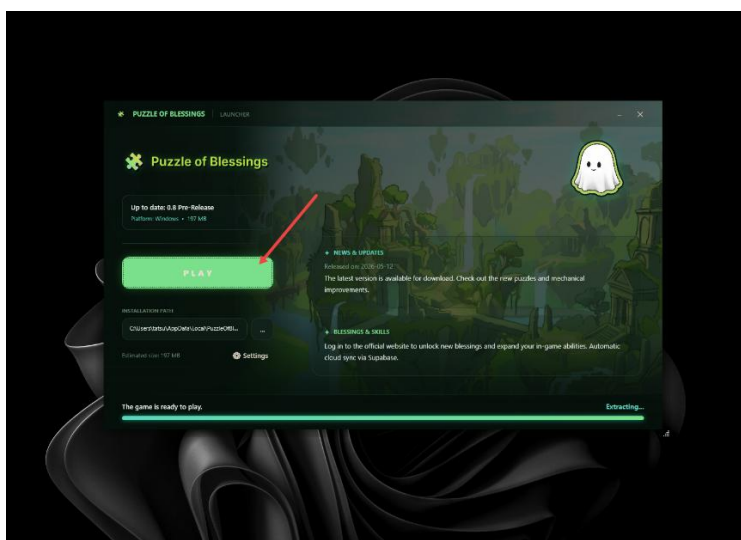


Nota: O *Launcher* verifica automaticamente atualizações do jogo e dele próprio sempre que é aberto. Se existir uma versão mais recente, esta é descarregada antes de permitir o acesso ao jogo, caso algum erro persista, desinstale o jogo na pasta do diretório, ou mesmo o próprio *Launcher*.

3.1.3 – Arranque do Jogo

Ao clicar em **PLAY** no Launcher, o jogo abrirá no menu principal. Para aceder ao conteúdo é necessário iniciar sessão com uma conta previamente criada no *website* oficial.

Figura 50 - Abrir Jogo



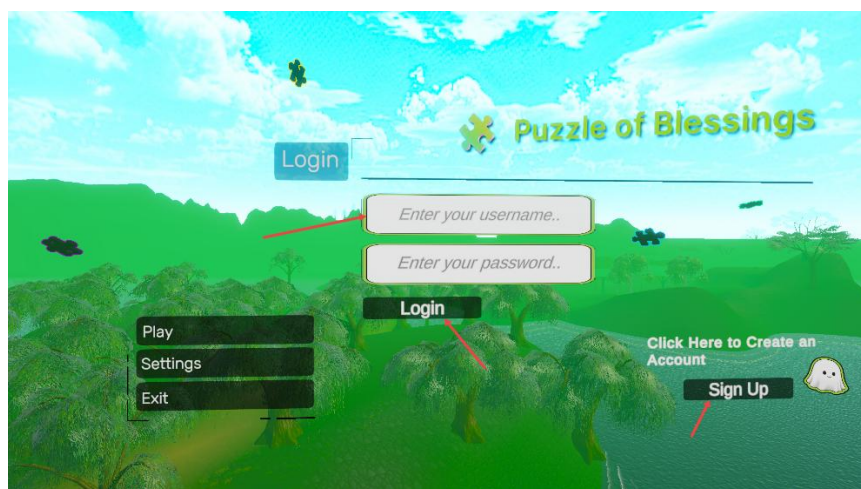
No menu principal, clique em **"Play"** para abrir o painel de login.

Figura 51 - Abrir Login



Introduza o nome de utilizador e a palavra-passe nos campos respetivos e clique em **Login**. As credenciais são verificadas em tempo real junto da base de dados na nuvem.

Figura 52 - Iniciar Sessão Ou Criar Conta



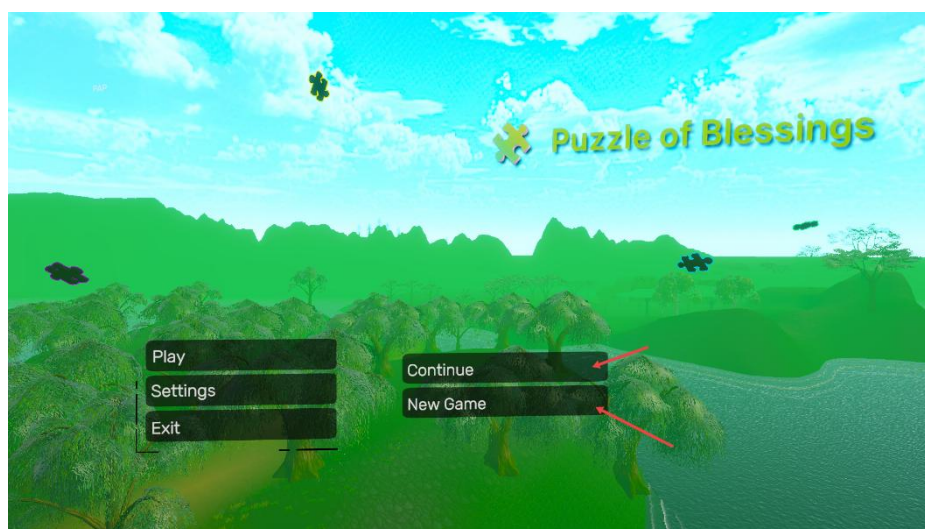
Nota: Caso ainda não possua uma conta, clique no botão **Sign Up** para ser redirecionado para o *website* de registo. O registo é realizado exclusivamente através do *website*, o jogo não permite criar contas diretamente.

Após autenticação bem-sucedida, surgem duas opções:

- **Continue** — retoma a sessão guardada anteriormente, teletransportando o jogador para a última posição registada.
- **New Game** — inicia uma nova campanha desde o início.

Atenção: esta ação apaga permanentemente o progresso anterior, incluindo Bênçãos desbloqueadas, moedas recolhidas e colecionáveis obtidos.

Figura 53 - Novo Jogo ou Continuar



3.2 – Habilidades / Blessings do Jogo

As Bênçãos são o elemento central da jogabilidade do Puzzle of Blessings. Sendo um total de sete cartas, cada carta representa uma habilidade especial desbloqueável ao longo da aventura. Uma vez obtidas, são sincronizadas na nuvem e ficam associadas permanentemente à conta do jogador.

Cada Bênção possui os seguintes atributos:

- Tecla de ativação — ao premir a tecla configurada, a carta na interface começa a tremer, indicando que o efeito está ativo.
- Duração — tempo em segundos durante o qual o efeito se mantém.
- Recarga (*Cooldown*) — período de bloqueio após a desativação; a carta aparece sombreada a cinzento e ligeiramente deslocada para baixo na interface.
- Cronómetro — contador visível na carta com os segundos restantes de recarga, apresentado com casas decimais para maior precisão.

Figura 54 - Cartas Disponíveis



3.2.1– Descrição Blessings

Spirit Vision — Permite ver o espectro espiritual e detetar objetos e elementos que anteriormente não eram visíveis.

Spirit Anchor — Permite ancorar objetos no ar através de poderes espirituais.

Duplication — Cria uma cópia temporária de um objeto físico que possa ser segurado. (Ativar o modo de construção, premir a tecla da carta, depois clicar com o botão direito para copiar e novamente com o botão direito para colar.)

Ephemeral Point — Gera uma plataforma mágica com duração de 8 segundos.

Pattern Lenses — Revela, durante 4 segundos, a sequência correta de puzzles.

Sequential Jump — Permite ao jogador realizar até um triplo salto no ar, caso acerte a sequência correta.

Admin NoClip — Permite voar e entrar num estado imaterial. Esta carta é exclusiva para administradores e é inacessível aos jogadores comuns.

3.3 – Objetos Seguráveis

3.3.1 – Dado Azul

Figura 55 - Dado Azul



Um cubo azul com 6 lados, disponível no tutorial para testar a mecânica de agarrar e girar, sem qualquer utilidade adicional.

3.3.2 – Chave Tutorial

Figura 56 - Chave Tutorial



Uma chave disponível em cima de uma mesa no tutorial, utilizada para testar a mecânica de agarrar e para abrir o portão que dá acesso ao próximo nível.

3.3.3 – Memory Fragment

Figura 57 - Fragmento de Memória



É um objeto apenas visível através da Spirit Vision. Quando é agarrado e combinado com outra memória, desencadeia um teleporte para outra região do jogo.

3.4 – Colecionáveis

3.4.1 – Cassete VHS

Figura 58 - Cassete VHS



Uma cassete VHS que funciona apenas como colecionável, localizada no primeiro nível do jogo e apresentada como conquista no site.

3.4.2 – Troféu de Escalada

Figura 59 - Troféu de Escalada



Um Troféu de Escalada atribuído aos jogadores que enfrentam os desafios com coragem, resiliência e um espírito inabalável. Localizado no Terceiro nível.

3.4.3 – Almofada dos Sonhadores

Figura 60 - Almofada dos Sonhadores

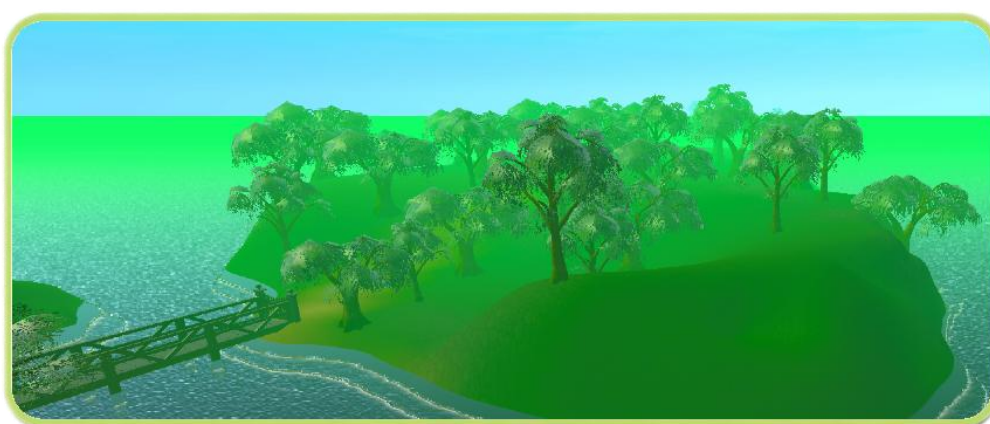


Um colecionável em forma de almofada, criado no coração da Dream's Island e preenchido com sonhos tranquilos e acolhedores. Localizado no nível 1, numa ilha com uma árvore enorme.

3.5 – Níveis do Jogo

3.5.1 – Tutorial

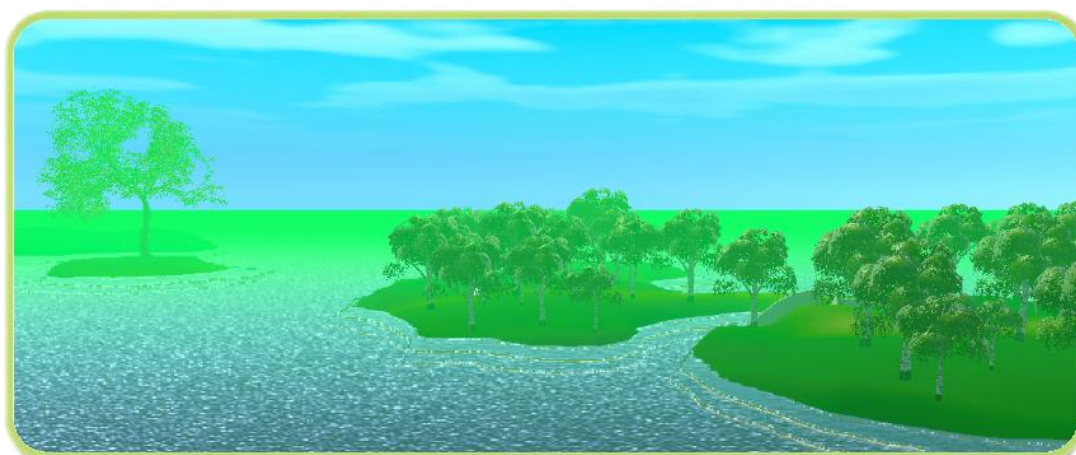
Figura 61 – Tutorial



Nível introdutório destinado à aprendizagem dos comandos básicos e das principais mecânicas do jogo.

3.5.2 – Nível 1 (Genesis Field)

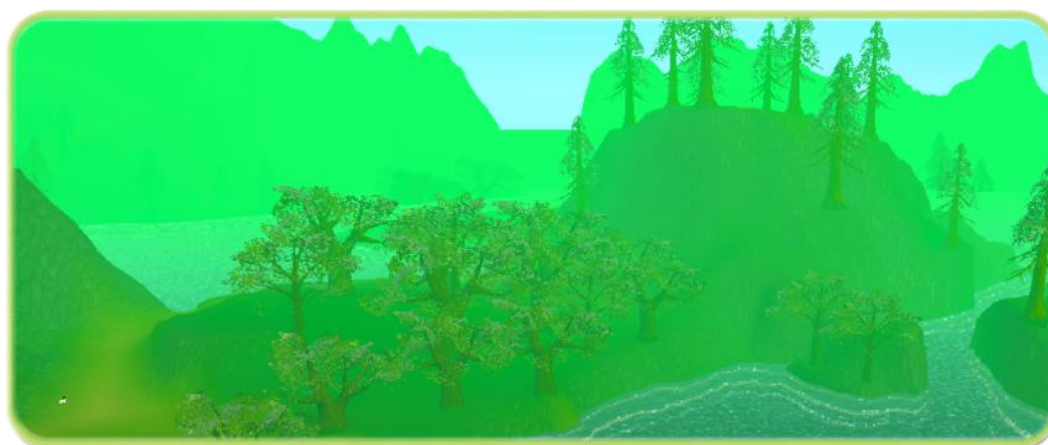
Figura 62 - Genesis Field



O início da jornada do jogador pelas vastas planícies iluminadas, onde começam as primeiras descobertas e desafios.

3.5.3 – Nível 2 (Deep Exploration)

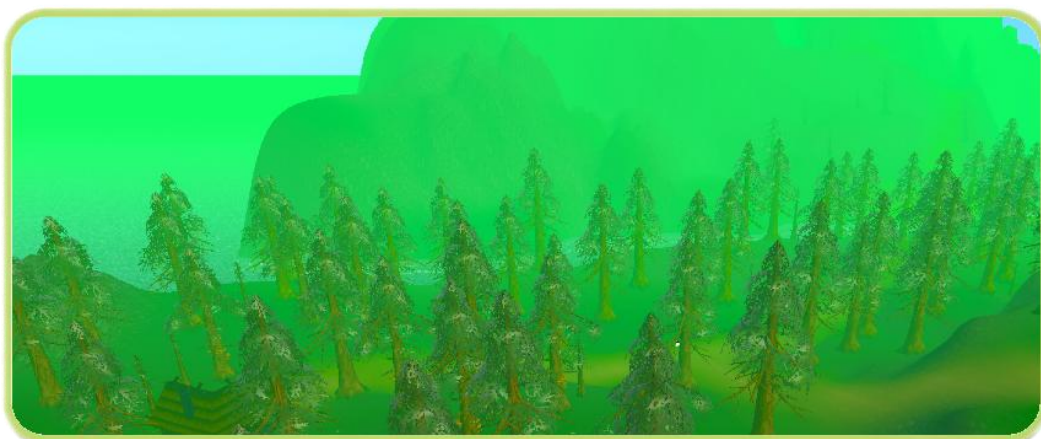
Figura 63 - Deep Exploration



Um nível focado na exploração de cadeias montanhosas e vales escondidos acima das nuvens, incentivando a descoberta e a progressão do jogador.

3.5.4 – Nível 3 (Spirit Realm)

Figura 64 - Spirit Realm



O desafio final, puzzle da floresta assombrada e, lugar onde o jogador deverá resolver puzzles envolvendo o rio para construir uma ponte para conseguir avançar.

4 – TRABALHO FUTURO

O término do desenvolvimento base do *Puzzle of Blessings* para a presente Prova de Aptidão Profissional não representa o fim do projeto. Tendo esta infraestrutura consolidada, o leque de possibilidades para a expansão do universo do jogo é vasto, justificando um planeamento detalhado para desenvolvimentos futuros. Uma das principais ideias no roteiro futuro do jogo é a aplicação de múltiplas línguas mas principalmente português.

Quebrando barreiras linguísticas e garantindo que o jogo atinja e seja plenamente compreendido por um público mais global e diversificado, elevando o nível de acessibilidade e de profissionalismo do produto final.

Modelação e integração de novas ilhas e biomas inexplorados, com desafios de complexidade superior. A intenção é desenhar *puzzles* de uso isolado de uma única habilidade, obrigando o jogador a compreender e a explorar sinergias entre as várias *Blessings* disponíveis. Paralelamente, o sistema de progressão do jogador tem uma enorme margem de expansão. O jogador poderia ganhar a capacidade de investir no aprimoramento individual de cada poder, desbloqueariam atributos secundários para poderes já existentes, recompensando a exploração rigorosa dos cenários.

O ecossistema em torno do jogo, nomeadamente a sua vertente *Web* e Social, é outra frente de forte aposta. Uma vez que o *Puzzle of Blessings* já dispõe de um canal de comunicação bidirecional com a base de dados relacional *Supabase*. Possivelmente a introdução de Tabelas de Classificação Globais (*Leaderboards*) para a conclusão de níveis ou a resolução de determinados *puzzles* em tempo recorde (*speedruns*) injetaria um fator competitivo saudável, prolongando substancialmente a longevidade (*replayability*) do jogo.

RESULTADOS E DISCUSSÃO

Ao longo do desenvolvimento do Puzzle of Blessings, os objetivos definidos no plano inicial foram alcançados, com as adaptações naturais decorrentes dos desafios técnicos encontrados durante o processo. O jogo encontra-se funcional em todos os seus elementos principais. O sistema de autenticação com a base de dados Supabase funciona de forma consistente e segura, incluindo proteção das palavras-passe por hashing, garantindo que cada jogador acede exclusivamente ao seu próprio progresso. O mapa principal, composto por ilhas, está concluído e jogável, com os níveis previstos implementados: o tutorial da chave e do portão, o puzzle das alavancas, o SimonSays, a armadilha na floresta e a construção da ponte, com as mecânicas de Blessings e a sequência de fim de jogo. As opções de personalização da experiência de jogo, incluindo o controlo de volume geral e de música através de sliders no menu de pausa estão implementadas, com os valores persistidos entre sessões através de *PlayerPrefs*. Os puzzles apresentam uma progressão de dificuldade coerente: do tutorial simples, que introduz a mecânica de agarrar objetos, até ao puzzle de memória e à sequência final que exige a utilização combinada de duas habilidades distintas. Esta progressão alinha-se com os objetivos pedagógicos do projeto e com os princípios de aprendizagem ativa que fundamentam o enquadramento teórico. O *Blessing Manager* revelou ser a decisão técnica mais sólida do projeto. A sua arquitetura modular, baseada na interface *IBlessingAbility* e na configuração via Inspetor do Unity, permitiu implementar progressivamente seis habilidades distintas sem necessidade de alterar o código central. A integração com o Supabase para persistência das habilidades desbloqueadas funcionou corretamente em todos os cenários testados, incluindo o *reset* completo no modo Novo Jogo. O desenvolvimento do *launcher* personalizado com auto-atualização representou um resultado técnico de maturidade assinalável no contexto de um projeto escolar. A solução implementada com verificação de versão via Supabase, *download* automático, substituição de ficheiros em execução e instalador gerado com *Inno Setup*, é equivalente, em princípio de funcionamento, ao que se encontra em produtos comerciais de distribuição de jogos.



CONSIDERAÇÕES FINAIS

O desenvolvimento do *Puzzle of Blessings* constituiu um percurso muito trabalhoso e exigente em colaboração com meu colega, aprendendo sobre novos recursos e aplicando os conhecimentos adquiridos ao longo dos três anos do Curso Profissional de Técnico Programador de Informática.

Em termos de aprendizagem, este projeto permitiu consolidar e aplicar competências em programação orientada a objetos em C#, arquitetura de software em Unity, comunicação com APIs REST, design de interfaces, e distribuição de software. A necessidade de articular o jogo com a base de dados e com o website, adicionalmente, competências de comunicação técnica e gestão de dependências entre sistemas distintos.

Em síntese, o *Puzzle of Blessings* é simultaneamente um produto de entretenimento, uma ferramenta pedagógica e a demonstração prática de que é possível, com os recursos disponíveis num contexto escolar, construir um sistema técnico completo, com propósito social definido.



BIBLIOGRAFIA

Doe, J. (n.d.). Retrieved from Sketchfab: <https://sketchfab.com/>

Kolb, D. A. (1984). *Experiential Learning: Experience as the Source of Learning and Development*. Englewood Cliffs, NJ: Prentice-Hall.

Minnesota, U. o. (2023). *Active Learning*. Retrieved from <https://cei.umn.edu/active-learning>

Piaget, J. (1969). *The Psychology of the Child*. New York: Basic Books.

Pixabay. (n.d.). Retrieved from Pixabay: <https://pixabay.com/>

Society, National Autistic. (2023). *Autism and learning*. Retrieved from <https://www.autism.org.uk>

Unity Technologies. (n.d.). Retrieved from <https://assetstore.unity.com>

W3Schools. (n.d.). Retrieved from <https://www.w3schools.com/>

Anexos

ANEXO 1 – AVALIAÇÕES

Anexo 1.1 – Primeira Avaliação Acompanhamento PAP

 REPÚBLICA PORTUGUESA EDUCAÇÃO, CIÊNCIA E INOVAÇÃO	AGRUPAMENTO DE ESCOLAS DR. GINESTAL MACHADO CURSO PROFISSIONAL DE TÉCNICO DE PROGRAMAÇÃO DE INFORMÁTICA Ano letivo 2025-2026	 Agrupamento de Escolas Dr. Ginestal Machado
---	---	---

**RELATÓRIO DE ACOMPANHAMENTO
DA PROVA DE APTIDÃO PROFISSIONAL**

1ª APRESENTAÇÃO à Comissão de Acompanhamento da PAP do(s) aluno(s): João Andrade

Tema do projeto: Logo digital 3D

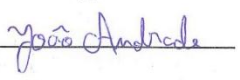
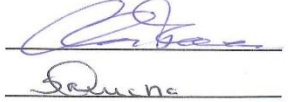
Fase de elaboração em que se encontra:

Iniciado	Sim	☒	Não	☐
Estruturado	Sim	☐	Não	☒
Concluído	Sim	☐	Não	☒

Apreciação Global: Apresentação Muito Boa Desenvolvida

Muito Boa.

Data: 14 de janeiro de 2026

O Aluno 	A Comissão 
---	---

1

Cofinanciado por:



Os Fundos Europeus mais próximos de si.

Anexo 1.2 – Segunda Avaliação Acompanhamento PAP

 REPÚBLICA PORTUGUESA EDUCAÇÃO, CIÊNCIA E INOVAÇÃO	AGRUPAMENTO DE ESCOLAS DR. GINESTAL MACHADO CURSO PROFISSIONAL DE TÉCNICO DE PROGRAMAÇÃO DE INFORMÁTICA Ano letivo 2025-2026	 Agrupamento de Escolas Dr. Ginestal Machado
---	--	---

RELATÓRIO DE ACOMPANHAMENTO DA PROVA DE APTIDÃO PROFISSIONAL

2ª APRESENTAÇÃO à Comissão de Acompanhamento da PAP do(s) aluno(s): João Pedro Lopes Pais de Andrade

Tema do projeto: Puzzle of Blessing: jogo digital educativo em 3D

Fase de elaboração em que se encontra:

Iniciado	Sim	☒	Não	☐
Estruturado	Sim	☒	Não	☐
Concluído	Sim	☐	Não	☐

Apreciação Global: Apresentar e trabalhar
Exatente

Data: maio de 2026

O Aluno

João Andrade

A Comissão

[Signature]
[Signature]

Cofinanciado por:



Anexo 1.3 – Terceira Avaliação Acompanhamento PAP

 REPÚBLICA PORTUGUESA EDUCAÇÃO, CIÊNCIA E INOVAÇÃO	AGRUPAMENTO DE ESCOLAS DR. GINESTAL MACHADO CURSO PROFISSIONAL DE TÉCNICO DE PROGRAMAÇÃO DE INFORMÁTICA Ano letivo 2025-2026	 Agrupamento de Escolas Dr. Ginestal Machado
---	--	---

RELATÓRIO DE ACOMPANHAMENTO DA PROVA DE APTIDÃO PROFISSIONAL

3ª APRESENTAÇÃO à Comissão de Acompanhamento da PAP do(s) aluno(s): João Pedro Lopes Pais de Andrade

Tema do projeto: Puzzle of Blessing: jogo digital educativo em 3D

Fase de elaboração em que se encontra:

Iniciado	Sim	☒	Não	☐
Estruturado	Sim	☒	Não	☐
Concluído	Sim	☒	Não	☐

Apreciação Global: Trabalho Excelente Apresentado
Muito Bom

Data: 6 de maio de 2026

O Aluno

João Andrade

A Comissão

[Signature]
[Signature]

Cofinanciado por:



Anexo 1.4 – Autoavaliação

 REPÚBLICA PORTUGUESA EDUCAÇÃO, CIÊNCIA E INOVAÇÃO	AGRUPAMENTO DE ESCOLAS DR. GINESTAL MACHADO Prova de Aptidão Profissional Triénio 2023/2026	 Agrupamento de Escolas Dr. Ginestal Machado
--	--	---

Curso Profissional de Técnico de Programador de Informática

Autoavaliação

Nome do Aluno: João Pedro Lopes Pais de Andrade

Ano/Turma: 12º

Nº: 13

Tema: P

Nota Final do Projeto: 20 (0 a 20 Valores)

Santarém, 22 de maio de 2026

O aluno

João Andrade

Cofinanciado por:



Os Fundos Europeus mais próximos de si.

