

**Aplicação “Jogo Walk Alone”**

## **Relatório de Desenvolvimento de Software**

ISMT – Engenharia de Software 2024-25

João Melro - 12987 - [12987@ismt.pt](mailto:12987@ismt.pt)

## Índice

<b>Controlo de Versões</b>	<b>2</b>
<b>1. Introdução</b>	<b>3</b>
<b>2. Análise de Contexto</b>	<b>5</b>
<b>3. Requisitos</b>	<b>7</b>
<b>4. Diagramas de Casos de Uso</b>	<b>8</b>
<b>5. User Stories</b>	<b>10</b>
<b>6. Análise de Domínio / Modelo Entidade-Relação</b>	<b>12</b>
<b>7. Análise da Estrutura do Projeto</b>	<b>14</b>
<b>8. Apresentação do Projeto Final: "Screenshots" da UI</b>	<b>17</b>
<b>9. Conclusão Geral / Reflexão final</b>	<b>18</b>

## Controlo de Versões

Versão	Data	Autores	Comentários
0.1.0	2024-10-20	João Melro	Criação inicial do projeto com a configuração básica da Unreal Engine 5 e definição do conceito do jogo.
0.2.0	2024-10-25	João Melro	Desenvolvimento dos scripts essenciais, incluindo a movimentação do jogador e mecânicas básicas, como checkpoints e jump boost.
0.3.0	2024-11-01	João Melro	Implementação de objetos colecionáveis (moedas e gemas) e integração com o sistema de pontuação.
0.4.0	2024-11-25	João Melro	Adição de funcionalidades avançadas, como chaves para abrir portas e plataformas móveis.
1.0.0	2024-12-15	João Melro	Versão final com ajustes na jogabilidade, testes e criação da demo jogável para apresentação.

# 1. Introdução

Este projeto consiste na criação de um jogo 3D na terceira pessoa, do tipo aventura, intitulado "**Walk Alone**". O principal objetivo do jogo é proporcionar diversão e competitividade entre amigos, colegas e familiares, desafiando os jogadores a completar os níveis recolhendo o maior número de gemas e moedas no menor tempo possível.

Inicialmente, o conceito era criar um jogo no estilo "Street Fight" utilizando Unity 3D. No entanto, após explorar outras opções, foi decidido usar o **Unreal Engine 4**, devido à sua capacidade de criar gráficos mais apelativos, eficientes e dinâmicos. Este projeto resultou numa demo funcional que demonstra o potencial do jogo e abre caminho para o seu desenvolvimento futuro.

O relatório está organizado em várias secções, cada uma destinada a descrever diferentes etapas e componentes do desenvolvimento deste jogo:

- **Análise de Contexto:** Esta secção descreve o ambiente e a motivação para a criação do jogo, identificando as necessidades e objetivos do projeto.
- **Requisitos:** Especifica os requisitos funcionais e não-funcionais do jogo, detalhando as funcionalidades principais e os parâmetros de qualidade que o projeto deve cumprir.
- **Diagramas de Casos de Uso:** Apresenta um diagrama que ilustra as interações entre o jogador e o sistema, destacando as principais ações que podem ser realizadas no jogo.
- **User Stories:** Contém histórias de utilizador que descrevem, de forma simplificada, as necessidades e objetivos do jogador ao interagir com o jogo.
- **Análise de Domínio / Modelo Entidade-Relação:** Descreve a modelagem dos dados, apresentando um diagrama entidade-relação que ilustra os principais elementos do jogo e as suas relações.
- **Análise da Estrutura do Projeto:** Explica a organização do código e a estrutura do jogo, abordando como as diferentes partes (scripts, blueprints e assets) se interligam para proporcionar a jogabilidade desejada.
- **Apresentação do Projeto Final: Screenshots da UI:** Inclui capturas de ecrã da interface do jogo para demonstrar o design e exemplificar a experiência do jogador.
- **Conclusão Geral / Reflexão Final:** Finaliza o relatório com uma reflexão sobre os desafios e aprendizagens do desenvolvimento do jogo, além das considerações sobre possíveis melhorias futuras.

Este relatório procura documentar cada etapa do desenvolvimento de forma clara e detalhada, oferecendo uma visão abrangente dos processos, decisões e ferramentas utilizadas para construir um jogo funcional, apelativo e divertido.

## 2. Análise de Contexto

O jogo "**Walk Alone**" foi desenvolvido para atender à necessidade de criar uma experiência imersiva e desafiadora para os jogadores, combinando aventura e competição. Este projeto surge como uma oportunidade de explorar as possibilidades da criação de jogos 3D, aplicando conhecimentos adquiridos ao longo do curso técnico e utilizando ferramentas avançadas como o Unreal Engine 5.

### Caracterização do Projeto e do Seu Meio Envolvente

O "**Walk Alone**" é um jogo projetado para jogadores que apreciam desafios de habilidade, exploração e superação. O objetivo principal é completar níveis recolhendo gemas e moedas enquanto supera obstáculos como plataformas móveis, checkpoints, e puzzles envolvendo chaves e portas.

O jogo foi concebido para funcionar como uma demo inicial, mas com potencial de expansão para se tornar um produto completo. A escolha do Unreal Engine 5 permitiu criar um ambiente visualmente apelativo e uma jogabilidade fluida, essencial para atrair e envolver os jogadores. O foco na diversão e na competição entre amigos e familiares orientou as decisões de design e desenvolvimento.

### Motivação para Desenvolver o Jogo / Identificação das Necessidades

A motivação para desenvolver o "**Walk Alone**" surgiu do desejo de criar um jogo que fosse, ao mesmo tempo, divertido de jogar e desafiador de desenvolver. Antes de iniciar o projeto, o criador identificou uma oportunidade de combinar mecânicas clássicas de jogos de aventura com elementos modernos, como gráficos de alta qualidade e sistemas de pontuação competitivos.

Os principais objetivos do jogo incluem:

- **Desafio e Diversão:** Criar um ambiente onde os jogadores possam testar suas habilidades enquanto se divertem com mecânicas simples, mas envolventes.
- **Competitividade:** Incentivar a competição saudável entre os jogadores, com base na recolha de moedas e gemas e na conclusão dos níveis no menor tempo possível.
- **Exploração de Ferramentas:** Utilizar o Unreal Engine 5 para explorar a criação de jogos, combinando Blueprints e programação em C# para implementar as mecânicas principais.

O projeto também reflete a necessidade de desenvolver competências práticas em programação, design de jogos e animações, integrando todas essas áreas para criar uma experiência de jogo coesa.

Com o "**Walk Alone**", espera-se não apenas alcançar os objetivos técnicos e criativos do projeto, mas também abrir portas para um futuro promissor no desenvolvimento de jogos, mostrando o potencial do criador e a viabilidade de transformar a demo em um produto completo.

### 3. Requisitos

#### *Requisitos Funcionais*

Os requisitos funcionais descrevem as funcionalidades específicas que o jogo deve oferecer para atender às necessidades dos jogadores e alcançar os objetivos do projeto. Eles representam as principais ações que o jogador pode realizar no jogo, como interagir com objetos, superar desafios e completar níveis.

#	Descrição	Prioridade
RF0 1	O jogador deve poder interagir com checkpoints para salvar o progresso.	Alta
RF0 2	O jogador deve poder recolher moedas e gemas para aumentar a pontuação.	Alta
RF0 3	O jogador deve poder utilizar o jump boost para alcançar áreas difíceis.	Média
RF0 4	O jogador deve poder usar chaves para abrir portas e desbloquear novas áreas.	Alta
RF0 5	O jogador deve poder navegar em plataformas móveis para avançar no nível.	Alta
RF0 6	O jogo deve exibir o número de moedas e gemas recolhidas no final de cada nível.	Média
RF0 7	O jogo deve incluir um sistema de pontuação baseado no tempo e nos itens recolhidos.	Baixa

### *Requisitos Não-Funcionais*

Os requisitos não-funcionais especificam os atributos de qualidade do jogo, como desempenho, usabilidade e compatibilidade. Estes requisitos definem as condições em que o jogo deve operar, garantindo que ele seja confiável, eficiente e apelativo para os jogadores.

#	Descrição	Prioridade
RNF0 1	O jogo deve funcionar sem interrupções em sessões de até 2 horas.	Alta
RNF0 2	O jogo deve carregar cada nível em menos de 5 segundos.	Média
RNF0 3	A interface do jogo deve ser intuitiva e acessível para todos os jogadores.	Alta
RNF0 4	O jogo deve ser compatível com computadores Windows e MacOS.	Baixa
RNF0 5	O desempenho do jogo deve ser estável, mantendo pelo menos 60 FPS.	Alta
RNF0 6	O jogo deve suportar até 10 jogadores em simultâneo em partidas locais.	Média

## 4. Diagramas de Casos de Uso

Os diagramas de casos de uso são ferramentas fundamentais na análise de sistemas, pois permitem representar de forma clara e visual as interações entre o jogador e o jogo. Cada caso de uso é representado por uma "oval", que define uma ação ou um conjunto de ações que o jogador pode realizar no sistema. O ator principal, o **Jogador**, é representado no diagrama e conectado aos casos de uso relevantes.

### Descrição Geral dos Casos de Uso

No contexto do jogo "**Walk Alone**", os casos de uso foram organizados para refletir as ações e funcionalidades disponíveis para o jogador:

- **Interagir com Checkpoints:** Permite ao jogador ativar pontos de controle que salvam o progresso no jogo.
- **Recolher Moedas e Gemas:** O jogador pode apanhar moedas e gemas espalhadas pelo mapa, aumentando a pontuação.
- **Utilizar Jump Boost:** O jogador pode ativar um "power-up" que permite realizar saltos mais altos para alcançar áreas difíceis.
- **Usar Chave para Abrir Porta:** O jogador pode encontrar e usar chaves para desbloquear portas e acceder a novas áreas.
- **Navegar em Plataformas Móveis:** O jogador deve interagir com plataformas móveis para superar desafios e avançar no jogo.

### Estrutura do Diagrama de Casos de Uso

O diagrama de casos de uso para o jogo inclui os seguintes elementos:

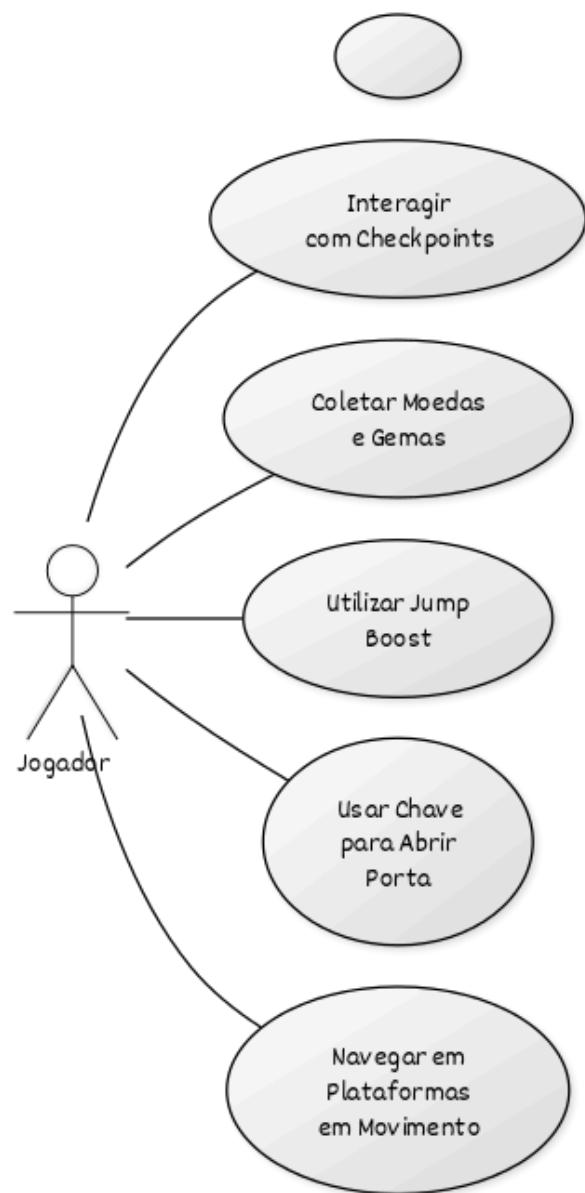
- **Autor:**
  - **Jogador:** Representado como o único autor, com acesso a todas as funcionalidades do jogo.
- **Casos de Uso:**
  - Cada caso de uso é representado por uma oval, com o nome da funcionalidade principal (ex.: "Recolher Moedas e Gemas").
  - Linhas de associação conectam o autor (Jogador) aos casos de uso, indicando as ações que pode realizar.

### Importância do Diagrama de Casos de Uso

O diagrama de casos de uso é uma ferramenta valiosa para o desenvolvimento do jogo, pois:

1. Facilita a comunicação entre os membros da equipa, garantindo que todos compreendem as funcionalidades principais.
2. Serve como documentação para validar os requisitos funcionais junto aos stakeholders, assegurando que o jogo atende às expectativas.
3. Fornece uma base sólida para a implementação e os testes, alinhando o desenvolvimento com os objetivos do projeto.

O diagrama criado é uma representação clara e visual das interações do jogador com o jogo, sendo uma referência importante para o desenvolvimento e a evolução do projeto.



CREATED WITH YUML

## 5. User Stories

As **User Stories** são uma técnica de modelagem de requisitos que descreve, de forma simplificada e na perspetiva do jogador, as funcionalidades esperadas do jogo. Elas ajudam a capturar as necessidades do utilizador final de forma clara e direta, permitindo que a equipa de desenvolvimento compreenda melhor as expectativas de quem irá jogar. Cada User Story é formulada no seguinte formato:

**"Como [Ator], pretendo [Caso de Uso], de modo a conseguir [Objetivo]"**

### User Stories para o Jogo "Walk Alone"

As User Stories foram desenvolvidas para refletir as ações e objetivos do jogador no jogo:

#### 1. Interagir com Checkpoints:

- "Como jogador, pretendo interagir com checkpoints, de modo a salvar o meu progresso no jogo e evitar repetir grandes secções após falhar."

#### 2. Recolher Moedas e Gemas:

- "Como jogador, desejo recolher moedas e gemas espalhadas pelo mapa, de modo a aumentar a minha pontuação e competir com outros jogadores."

#### 3. Utilizar Jump Boost:

- "Como jogador, pretendo utilizar o jump boost, de modo a alcançar áreas difíceis e explorar locais escondidos no mapa."

#### 4. Usar Chave para Abrir Porta:

- "Como jogador, quero encontrar e usar chaves, de modo a desbloquear portas que me permitam avançar no nível ou aceder a áreas secretas."

#### 5. Navegar em Plataformas Móveis:

- "Como jogador, desejo navegar em plataformas móveis, de modo a superar desafios e progredir no jogo."

#### 6. Sistema de Pontuação:

- "Como jogador, pretendo visualizar a minha pontuação baseada no tempo e nos itens recolhidos, de modo a avaliar o meu desempenho e competir com outros jogadores."

## 6. Análise de Domínio / Modelo Entidade-Relação

No contexto do jogo "Walk Alone", o **Modelo Entidade-Relação (ER)** foi desenvolvido para estruturar e organizar as informações necessárias para o funcionamento do jogo. O modelo ER identifica as entidades principais, os seus atributos e as relações entre elas, oferecendo uma visão clara e organizada do domínio de dados.

### Estrutura Geral do Modelo ER

O modelo ER foi projetado para atender aos requisitos funcionais do jogo, incluindo a gestão de objetos interativos, pontuações e progresso do jogador. Abaixo estão as principais entidades e suas características:

- **Entidades Principais:**

1. **Jogador:** Representa o utilizador que interage com o jogo.

- **Atributos:**

- id (chave primária)
    - nome
    - pontuacao\_total
    - nivel\_atual

2. **Objeto Interativo:** Representa elementos interativos no jogo, como moedas, gemas, checkpoints, plataformas móveis, chaves e portas.

- **Atributos:**

- id (chave primária)
    - tipo (moeda, gema, checkpoint, etc.)
    - estado (ativo/inativo)
    - posicao (coordenadas no mapa)

3. **Pontuação:** Regista a pontuação obtida pelo jogador em cada nível.

- **Atributos:**

- id (chave primária)
    - jogador\_id (chave estrangeira referindo-se a Jogador)
    - nivel (número do nível)
    - moedas\_recolhidas
    - gemas\_recolhidas
    - tempo

- **Relações:**
  1. **Jogador - Pontuação:** Um jogador pode ter várias pontuações associadas (relação de um para muitos), representando o desempenho em diferentes níveis.
  2. **Jogador - Objeto Interativo:** Um jogador pode interagir com múltiplos objetos no jogo, como recolher moedas ou ativar checkpoints (relação de muitos para muitos).

#### Descrição das Entidades e Relações

- **Jogador:** É a entidade central do jogo, representando o utilizador que interage com o sistema. Os atributos como pontuacao\_total e nivel\_atual ajudam a acompanhar o progresso do jogador.
- **Objeto Interativo:** Esta entidade engloba todos os elementos com os quais o jogador pode interagir. O atributo tipo permite identificar o propósito de cada objeto, enquanto o estado indica se o objeto já foi utilizado (ex.: moeda recolhida, checkpoint ativado).
- **Pontuação:** Regista os resultados de cada nível, permitindo que o jogador acompanhe o seu desempenho e compare com outros jogadores.

#### Ferramentas e Implementação

Para a criação do modelo ER, podem ser utilizadas ferramentas como o **LucidChart** ou o **Draw.io** para representar visualmente as entidades e suas relações. O diagrama final facilita o desenvolvimento do jogo, servindo como referência para a implementação do sistema e para a organização das classes e tabelas no Unreal Engine 4.

## 7. Análise da Estrutura do Projeto

O projeto do jogo "**Walk Alone**" foi desenvolvido com uma organização modular, garantindo uma separação clara de responsabilidades e facilitando a manutenção e a escalabilidade do sistema. A estrutura do projeto é composta por componentes principais que interagem para oferecer uma experiência de jogo fluida e eficiente.

#### Estrutura Geral do Projeto

O jogo foi dividido em três componentes principais: **Camada de Apresentação (Interface de Utilizador)**, **Camada de Lógica de Jogo** e **Camada de Dados**. Cada componente cumpre uma função específica, permitindo uma interação eficiente entre as partes e garantindo que cada uma lide exclusivamente com a sua responsabilidade.

#### Camada de Apresentação (Interface de Utilizador)

Esta camada oferece a interface visual do jogo e permite a interação do jogador com os elementos do jogo. A interface é composta por elementos como:

- **HUD (Heads-Up Display):** Mostra informações como pontuação, gemas recolhidas, moedas e tempo restante.
- **Menus:** Incluem o menu inicial, de pausa e de fim de nível.
- **Elementos Visuais Interativos:** Checkpoints, plataformas móveis e objetos colecionáveis.

A interface comunica diretamente com a camada de lógica do jogo para receber atualizações e enviar ações do jogador, como a ativação de checkpoints ou a recolha de moedas.

#### Camada de Lógica de Jogo

A lógica do jogo é responsável por gerir as regras e o comportamento dos elementos do jogo. Esta camada utiliza Blueprints e scripts em C# para implementar funcionalidades como:

- **Gestão de Objetos Interativos:** Controla o estado de moedas, gemas, chaves e portas.
- **Sistema de Pontuação:** Calcula e atualiza a pontuação com base nos itens recolhidos e no tempo gasto.
- **Gestão de Progressão:** Regista o progresso do jogador, incluindo checkpoints ativados e níveis concluídos.

A lógica do jogo atua como intermediária entre a interface de utilizador e os dados do jogo, assegurando que as ações do jogador são processadas corretamente e refletidas na interface.

### **Camada de Dados**

Esta camada é responsável por armazenar e gerir as informações do jogo. Inclui elementos como:

- **Estado do Jogo:** Guarda o progresso atual do jogador, incluindo pontuação, nível e objetos recolhidos.
- **Configurações do Jogo:** Armazena definições como controlos, volume de som e preferências de gráficos.
- **Persistência de Dados:** Permite salvar e carregar o progresso do jogador, utilizando ficheiros locais ou bases de dados simples.

### **Relação entre as Camadas**

#### 1. Interface e Lógica do Jogo:

- A interface comunica com a lógica do jogo para realizar ações, como recolher moedas ou ativar checkpoints. Por exemplo, quando o jogador recolhe uma moeda, a interface informa a lógica, que atualiza a pontuação.

#### 2. Lógica do Jogo e Dados:

- A lógica do jogo utiliza a camada de dados para armazenar e recuperar informações, como o progresso do jogador ou as configurações do jogo. Esta separação permite uma fácil manutenção e expansão futura.

### **Estrutura do Código**

O projeto foi organizado de forma modular, com pastas dedicadas a diferentes responsabilidades:

- **UI (Interface de Utilizador):** Contém os Blueprints e elementos visuais para menus e HUD.
- **Gameplay (Lógica de Jogo):** Inclui scripts e Blueprints responsáveis pelas regras e funcionalidades do jogo.
- **Data (Dados):** Armazena classes e ficheiros relacionados à persistência de dados, como progresso do jogador e configurações.

## 8. Apresentação do Projeto Final: "Screenshots" da UI

As capturas de ecrã apresentadas nesta secção foram retiradas diretamente do jogo "**Walk Alone**", desenvolvido como parte do projeto final. Estas imagens ilustram as funcionalidades principais, a jogabilidade e os elementos visuais mais relevantes, proporcionando uma visão clara do resultado final.

### Exemplos de Capturas de Ecrã

#### 1. Personagem Principal

Captura que destaca o personagem jogável, incluindo a implementação das ações principais, como movimentação, sprint, ativação do jump boost e interação com o ambiente.



*Figura 1: Personagem Principal*

#### 2. Mapa do Jogo

Demonstração de uma área do mapa, criada com recurso a **assets gratuitos da Unreal Engine 5**. A imagem mostra a organização espacial, as ilhas e os obstáculos que o jogador enfrenta.



*Figura 2: Mapa*

#### 3. Checkpoints

Uma imagem que apresenta o sistema de checkpoints, com mudança de cor e som de alerta a indicar a sua ativação.



Figura 3: Forma 3D do CheckPoint

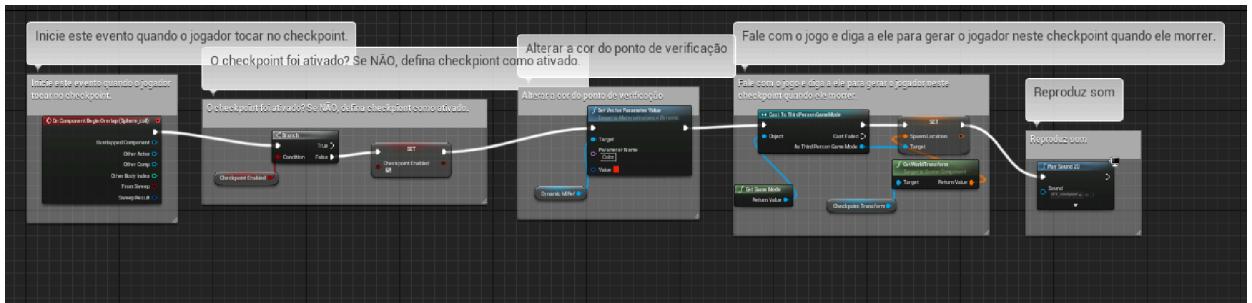


Figura 4: Event Graph CheckPoint

#### 4. Power-Up: Jump Boost

Captura do power-up que permite ao jogador realizar saltos mais altos durante um tempo limitado, representado visualmente por alterações no comportamento do personagem.



Figura 5: Forma 3D Jump Boost

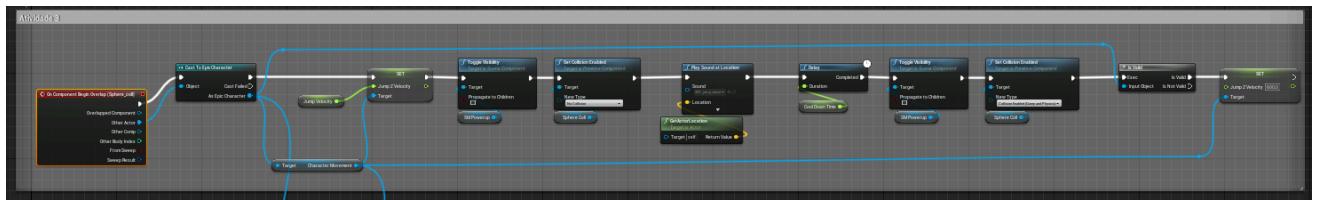


Figura 6: Event Graph Jump Boost

## 5. Objetos Colecionáveis: Coins e Gems

Imagen que destaca os coins e gems espalhados pelo mapa, que têm como objetivo aumentar a competitividade e o desafio do jogador.



Figura 6: Forma 3D Coin



Figura 7: Forma 3D Gem

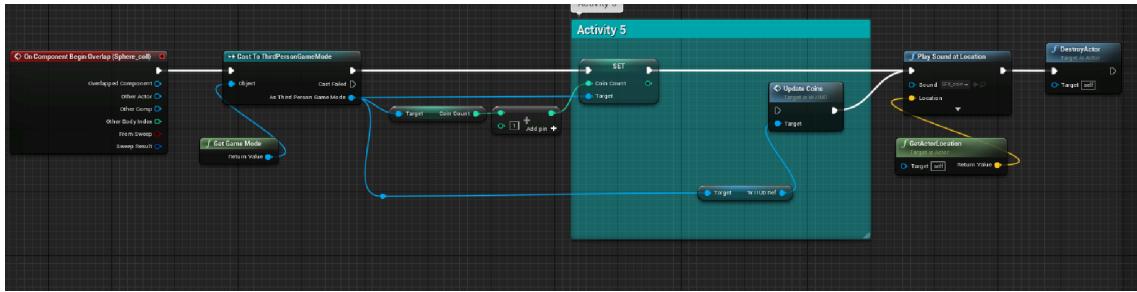


Figura 8: Event Graph Coin (Gem)

## 6. Portas e Chaves

Demonstração da interação do jogador com portas e chaves, incluindo mensagens de aviso e som de confirmação ao completar a ação correta.

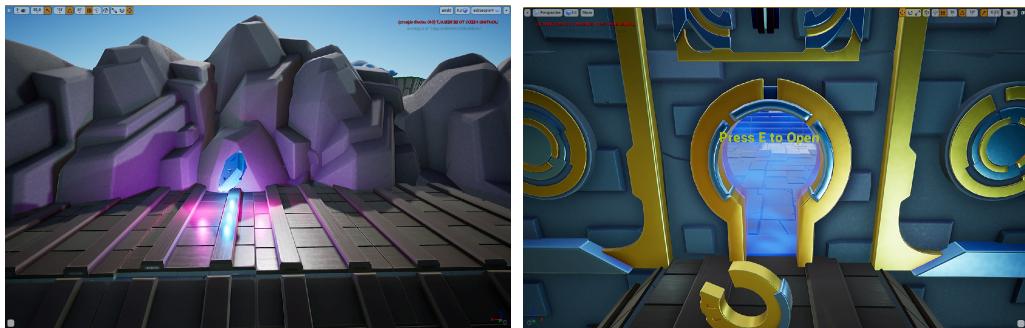


Figura 9: Forma 3D Chave e Door

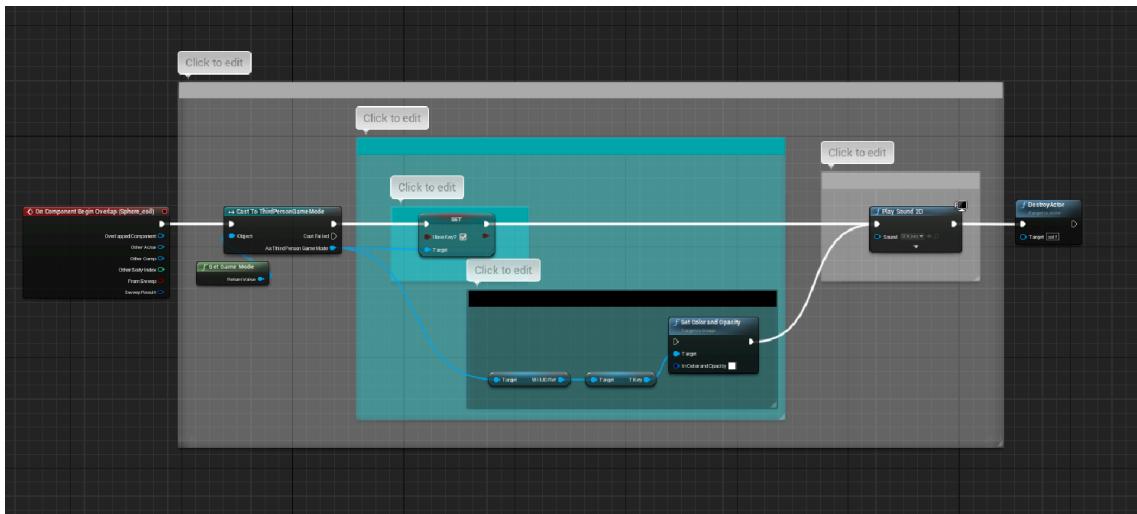


Figura 10: Event Graph Key

## 7. Plataformas em Movimento

Captura de plataformas móveis, destacando a utilização de sequências de movimento criadas com o **Level Sequence** para proporcionar desafios dinâmicos.

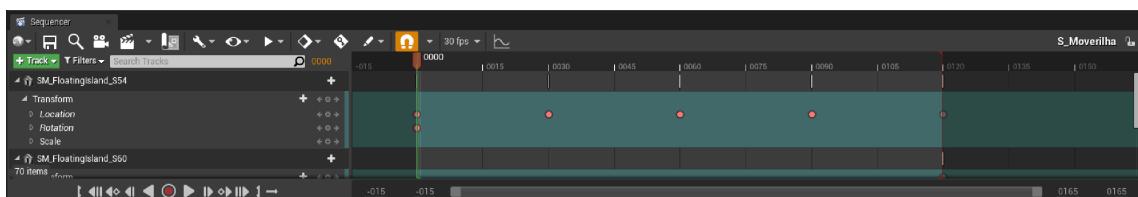


Figura 11: Level Sequence

## 9. Conclusão Geral / Reflexão final

O desenvolvimento do jogo "**Walk Alone**" foi uma experiência enriquecedora e desafiadora, permitindo-me aplicar, consolidar e expandir os conhecimentos adquiridos ao longo do curso de Gestão e Programação de Sistemas Informáticos. Este projeto representou uma oportunidade única de explorar ferramentas e técnicas de desenvolvimento de jogos, transformando uma ideia inicial numa demonstração funcional e visualmente apelativa.

Um dos aspectos mais marcantes foi a utilização do motor **Unreal Engine 5**, que me proporcionou um equilíbrio entre facilidade de uso e a possibilidade de criar ambientes gráficos de elevada qualidade. A escolha desta ferramenta revelou-se acertada, pois permitiu implementar funcionalidades complexas, como checkpoints, plataformas móveis e power-ups, com maior eficiência e clareza.

A programação de **scripts e blueprints** foi uma das áreas onde enfrentei maiores desafios. No entanto, esta também foi uma das componentes mais gratificantes, pois ajudou a aprofundar os meus conhecimentos técnicos e a compreender melhor as lógicas envolvidas na criação de ações e interações dentro de um jogo. O uso de Blueprints simplificou a criação de eventos visuais e comportamentos, enquanto a combinação com **C#** trouxe uma camada extra de controlo e personalização.

Outro elemento que considero um grande marco foi a criação de um mapa funcional e visualmente apelativo, aproveitando **assets gratuitos da Unreal Engine**. Esta abordagem não só economizou tempo como também permitiu concentrar-me no design e na jogabilidade. A implementação de elementos como gemas e moedas, que incentivam a competitividade, adicionou uma camada extra de desafio e diversão ao jogo.

A elaboração deste projeto ajudou-me a compreender a importância de equilibrar design, jogabilidade e performance. Desde a implementação de sistemas dinâmicos, como as plataformas móveis, até à criação de menus e interfaces simples, cada decisão técnica foi um passo em direção ao objetivo de proporcionar uma experiência de jogo cativante.

Por fim, este projeto não só reforçou a minha paixão pelo desenvolvimento de jogos como também me preparou para desafios futuros na área. Apesar das dificuldades encontradas, como aprender novas ferramentas e técnicas, consegui superá-las através de pesquisa, ajuda de colegas e recursos online. Este processo demonstrou que o desenvolvimento de jogos é uma área em constante evolução e cheia de oportunidades para inovar e aprender.

Em suma, o "**Walk Alone**" é mais do que um projeto académico: é um reflexo do meu crescimento enquanto programador e da minha dedicação em criar algo que não só me orgulhe, mas também oferece diversão e desafio a todos os que o experimentarem.