**Instituto Superior de Tecnologias Avançadas**

**Licenciatura em Engenharia Multimédia**

# '*NIGHT OF THE TERRORBORN*' - DESENVOLVIMENTO DE UM JOGO 3D DE TERROR, EM PRIMEIRA PESSOA

Projeto Académico, realizado por:

**David Miguel Fonseca da Costa**

**Lima Amorim**, n.º 21389

Coordenador – Professor Dr. Pedro Brandão

Orientadora – Professora Dra. Dulce Mourato

Lisboa, Ano Letivo 2022/23

**Instituto Superior de Tecnologias Avançadas**

**Licenciatura em Engenharia Multimédia**

# '*NIGHT OF THE TERRORBORN*' - DESENVOLVIMENTO DE UM JOGO 3D DE TERROR, EM PRIMEIRA PESSOA

Projeto Académico, realizado por:

**David Miguel Fonseca da Costa**

**Lima Amorim**, n.º 21389

Coordenador – Professor Dr. Pedro Brandão

Orientadora – Professora Dra. Dulce Mourato

Lisboa, Ano Letivo 2022/23

# DEDICATÓRIA

À minha avó.

# AGRADECIMENTOS

Aos meus pais, por todo o apoio incondicional e por me proporcionarem a possibilidade de progredir no meu percurso académico.

À minha irmã e sobrinhos por toda a companhia e palavras de incentivo.

À minha namorada por acreditar nas minhas capacidades e de me dar motivação e força todos os dias.

Aos meus amigos, Bruno, Sofia, Rúben, Claúdia e Duarte pelo total apoio e por me questionarem: "*Então e o projeto final*?"

A todos os restantes familiares, amigos, professores e colegas de curso que me apoiaram nesta etapa.

À minha orientadora, Professora Doutora Dulce Mourato pela sua orientação, disponibilidade e amabilidade ao longo deste ano letivo. Sem dúvida, um elemento crucial para o desenvolvimento do presente projeto.

Ao Instituto Superior de Tecnologias Avançadas (ISTEC), que me permitiu adquirir múltiplos conhecimentos, durante todo o meu processo académico e por toda a experiência obtida.

# RESUMO

Nos últimos anos, o número de jogadores ou '*gamers*', como são globalmente designados, verificou um acréscimo muito significativo, tornando a Indústria dos Videojogos um dos setores mais lucrativos. A Europa é a terceira região, em que o '*gaming*' constitui um enorme valor económico, logo a seguir ao continente asiático e americano. Portugal não é exceção, verificando também estas tendências, comportando-se à semelhança de outros países europeus. Este comportamento veio a acentuar-se com a pandemia COVID-19 e os sucessivos confinamentos, permitindo o acesso facilitado ao mundo dos jogos.

O *'**Night of the Terrorborn**'* consiste num videojogo de temática paranormal, com dimensão a três dimensões – 3D, na perspetiva de primeira pessoa. A conceção e desenvolvimento deste projeto teve como principal inspiração, o sucesso de videojogos *indie* de terror, como '*Slenderman'*, *'Phasmophobia'* e *'Lunch Lady'*.

O jogo centra-se, no ambiente do Campus Académico do Lumiar, interior, do ISTEC Lisboa e na exploração do mesmo, onde, ao longo do tempo, vão sendo experienciadas sensações de horror e ocorrências sobrenaturais. Este projeto baseou-se na construção de uma narrativa, na escolha do *design*, dos mecanismos, entre outros.

Os programas utilizados para o desenvolvimento do jogo foram a plataforma de criação e operação de jogos, UNITY.

No capítulo final averiguou se as questões de investigação foram respondidas e se foi cumprindo os objetivos estabelecidos no início do projeto, também são referidos aspetos a melhorar, no futuro.

PALAVRAS-CHAVE: Videojogo, Primeira-Pessoa, Terror, *Unity*, Linguagem #C

# ABSTRACT

In recent years, the number of players or ‘gamers’, as they are globally designated, deserved a very significant increase, making the Video Game Industry one of the most profitable sectors. Europe is the third region in which gaming is of enormous economic value, right after the Asian and American continents. Portugal is no exception, also verifying these trends, behaving like other European countries. This behaviour was motivated by the COVID-19 pandemic and successive confinements, allowing easier access to the world of games.

The video game ***‘Night of the Terrorborn’*** consists of a paranormal theme, with a three-dimensional – 3D, first-person perspective. The creation and development of this project were inspired by the success of indie horror video games, such as ‘*Slenderman*’, ‘*Phasmophobia*’ and ‘*Lunch Lady’*. The game focuses on the environment of the Academic Campus of Lumiar, the interior, of ISTEC Lisboa, and on its exploration, where, over time, sensations of horror and supernatural occurrences are experienced.

This project is based on the construction of a narrative, on the choice of design, and mechanisms, among others. The programs used for the development of the game were the game creation and operation platform, UNITY.

The final chapter examined whether the research questions were answered and whether the objectives set at the beginning of the project were being met, aspects to be improved in the future are also mentioned.



KEYWORDS: *Videogame, First-Person, Terror, Unity*, #C *Language*

# ABREVIATURAS E SIGLAS

AI – *Artificial Intelligence*

3D – 3 Dimensões

C# – C *Sharp*

*Co-op – Co-operative*

*DIY – Do it Yourself*

PC – *Personal Computer*

# ÍNDICE

INTRODUÇÃO ..... 8
REVISÃO DA LITERATURA ..... 9
MATERIAIS E MÉTODOS DE DESENVOLVIMENTO ..... 18
RESULTADOS E DISCUSSÃO DE RESULTADOS ..... 20
CONCLUSÃO ..... 44
REFERÊNCIAS BIBLIOGRÁFICAS ..... 45
APÊNDICES E ANEXOS ..... 48

# ÍNDICE DE FIGURAS

Figura 1 – Phasmophobia ..... 15
Figura 2 – Slenderman ..... 16
Figura 3 – *Chased by Darkness* ..... 16
Figura 4 – Lunch Lady ..... 17
Figura 5 –Script do jogo 'NIGHT OF THE TERRORBORN' ..... 20
Figura 6 – Faixas etárias dos entrevistados ..... 22
Figura 7 – Prevalência de género ..... 22
Figura 8 – Identificação do dispositivo de jogo ..... 23
Figura 9 – Frequência de tempo de jogo ..... 23
Figura 10 – Tipologias de género de jogos ..... 24
Figura 11 – Apetência pelo género Terror ..... 24
Figura 12 – Especificidade do tipo de jogo Indie ..... 25
Figura 13 – Características de um bom jogo de terror ..... 25
Figura 14 – Modo de jogar ..... 26
Figura 15 – Se utiliza habitualmente um repositório de jogos ..... 26
Figura 16 - Planta do jogo ..... 27
Figura 17 - Planta das salas ..... 27
Figura 18 - Sala de Aula ..... 28
Figura 19 - Casa de banho ..... 28
Figura 20 - Player ..... 29
Figura 21 - Porta ..... 31
Figura 22 - Luz ambiente ..... 33
Figura 23 . Lanterna ..... 33
Figura 24 - Pistola ..... 35
Figura 25 - Inimigo ..... 37
Figura 26 - Chave ..... 40
Figura 27 - Menu Inicial ..... 41
Figura 28 - Vitória ..... 41
Figura 29 - Derrota ..... 42
Figura 30 - Cenário de Vitória e respetivos menus ..... 43
Figura 31 - Cenário de Derrota e respetivos menus ..... 43

# ÍNDICE DE TABELAS

Tabela 1 – Cronograma ..... 19
Tabela 2 - Categorias apuradas para a realização do jogo ..... 20

# INTRODUÇÃO

O principal objetivo deste Projeto Académico consiste na **conceção e desenvolvimento de um jogo paranormal, da categoria de terror, a três dimensões – 3D, em primeira pessoa**. Este jogo é exclusivo para utilizadores que possuam computadores, com Sistema Operativo *Windows*.

Atualmente, a indústria dos jogos é um dos sectores com mais receita no mercado internacional e representa uma das atividades de lazer mais "procuradas" pela sociedade atual (Newzoo, 2019). No ano de 2021, com pandemia de COVID-19 e os sucessivos confinamentos, verificou-se o aumento exponencial nos lucros de venda de jogos e no número de *gamers*, termo recorrente para os jogadores. Estima-se que cerca de 3.5 mil milhões de pessoas jogam videojogos recorrentemente e que a tendências preveem que aumente (JPN, 2023).

Considerando que o aumento da popularidade do uso de computadores e o crescimento desta indústria, potencia o consumo de jogos, o **objetivo principal** deste projeto consiste na conceção e desenvolvimento de um jogo de terror paranormal e de entretenimento para um específico público-alvo, demonstrando competências na programação e modelagem 3D.

Os **objetivos específicos** do projeto compreendem a:

- Adoção de técnicas de estimulação visual e auditiva para a imersão do utilizador;
- Pesquisa e produção de situações que «testam os limites», utilizando técnicas apropriadas;
- Reprodução de cenários de «sobrevivência», obrigando os jogadores a ultrapassar as dificuldades do jogo;
- Enumeração das características do *software* e as suas capacidades para criação e desenvolvimento deste jogo.

# REVISÃO DA LITERATURA

Entende-se por ‘jogo’, a atividade em que um ou mais jogadores disputam, de acordo com regras e objetivos previamente estabelecidos. O conceito de jogo surge do termo latim ‘*jocus*’ que significa brincadeira, divertimento. Os jogos são atividades estruturadas, praticadas para fins recreativos, muitas das vezes, estimulando mentalmente e fisicamente o indivíduo. Sendo um termo tão abrangente, não existe uma definição precisa da palavra ‘jogo’ (Silva & Soares, 2023)

O conceito de jogos é mundialmente conhecido é uma realidade no nosso quotidiano. Com efeito, foi registada a presença de jogos e competições desde 2600 a.C., sendo um elemento universal em todas as culturas humanas (Sackson, 1991).

Os jogos podem ter uma variedade de formas, desde jogos de tabuleiro, a jogos de computador ou eletrónicos. Assim sendo, segundo existem oito tipo de jogos:

- Jogos de mesa;
- Jogos de papel e caneta;
- Jogos de cartas;
- Jogos de dados;
- Jogos de tabuleiro;
- Jogos musicais;
- Jogos interativos;
- Jogos eletrónicos.

Os jogos eletrónicos constituem os jogos praticados na internet ou em aparelhos como consolas, computadores, telemóveis, entre outros. Podem ser praticados individualmente (*single-player*) ou com vários jogadores (*multiplayer*). Para que o jogo comece, é necessário que o dispositivo utilizado receba instruções, tendo o jogador de comandar as ações desejadas, por meio de controles (Cruz, Lima & Nascimento, 2020).

Com a expansão da internet e a maioria da população “*online*”, os jogos eletrónicos, especialmente os de computador, verificaram um crescimento abrupto.

Como já anteriormente mencionado, os videojogos podem pertencer a cinco campos diferentes – de *browser*, de consola, de *tablet,* de computador e de *smartphones*. Ao contrário dos jogos “tradicionais”, o videojogo consiste na interação de um ou mais jogadores com os periféricos do aparelho escolhido – teclado, *joysticks*, entre outros, com a imagem visualizada no ecrã/monitor. Os sistemas eletrónicos que permitem os videojogos são denominados como

plataformas e podem ser desde computadores, como *arcades*, telemóveis e consolas (Cruz, Lima & Nascimento, 2020).

O avanço das tecnologias têm permitido que as interações entre o jogador e o jogo sejam cada vez mais intensas. Através de luzes, sons, vibrações e até óculos de realidade virtual, o jogador encontra-se imerso no mundo dos jogos (Barroso, 2020).

Segundo Guerra (2014), em trinta anos, os videojogos tornaram-se uma parte crucial da nossa cultura e a indústria dos videojogos tornou-se multibilionária.

As primeiras experimentações com videojogos, rondam a década de 40, no século XX, quando os físicos Thomas Goldsmith e Estle Ray Mann ligaram um tubo de raios catódicos a um osciloscópio. Os feixes de luz simulavam mísseis e o equipamento foi batizado de Dispositivo para Diversão de Tubo de Raios Catódicos.

Anos mais tarde, Charles Adama elaborou um programa que consistia numa bola saltitona, para o computador Whirlwind. E o britânico Christopher Strachey programou um simulador do jogo de damas para o *Pilot ACE*. O primeiro jogo gráfico para computador foi criado em 1952 e consistia no jogo do galo. Embora estas experiências pareçam insignificantes, moldaram o caminho para a criação e evolução dos videojogos.

Foi com William Higinbotham concebeu o '*Tennis for Two*' para um osciloscópio para entreter os visitantes do seu laboratório. Já na década de 60, o IMT ou Instituto de Tecnologias de Massachusetts desenvolveu o '*Spacewar*!' um jogo que consistia em duas naves espaciais que atiravam torpedos uma à outra. Este jogo foi reconhecido como o primeiro videogame (Techmundo, 2009).

Os primeiros jogos eletrónicos circulavam entre estudantes, em círculos universitários, em que o público era apenas *game-developers.* Assim sendo, na década de 70, as máquinas *arcade* e o uso de computadores pessoais, permitiram a elaboração do *Computer Space*, parecido com o '*Spacewar*!' mas com a capacidade de ser visualizado numa televisão. O jogo foi, posteriormente, comprado por Bushnell e Dabney, que lançaram a ATARI. A ATARI é uma empresa que se tornou pioneira em jogos de *arcade*, jogos em consolas e em computadores. Foi uma das principais responsáveis pela popularização dos videogames, nomeadamente do '*Pong'*, '*Asteroids'*, '*Missile Command'*. (Batista & Batista 2007)

Figura 1 – Consola de Videojogos Atari 2600 (The Strong – National Museum of play, s.d.)

Ainda na mesma década, surge a primeira consola doméstica – do qual se deu o nome de *Magnavox Odyssey*, da qual não teve grande impacto ou adesão. Em 1979, a empresa ATARI lança o jogo com mais sucesso, '*Asteroids'*, sendo possível que o sistema registar as iniciais dos jogadores com maior pontuação. Também no mesmo ano, criou o primeiro jogo em primeira pessoa e tridimensional, um simulador de guerra, o '*Battlezone'* (Techmundo, 2009).

Nos anos 80, no Japão, surgiu o tão conhecido jogo do '*Pacman'*, que ao contrário da tendência para jogos de guerra e simuladores, apenas seria um jogo que o objetivo principal seria comer. Este jogo mais simples, mas extremamente viciante, inspirou o russo Alex Pajitnov a elaborar o '*Tetris'*, cujo objetivo principal é encaixar peças. Foi neste período, que a empresa japonesa *Nintendo* produziu a consola *Famicom* e mais tarde a *Nintendo Entertainment System*. Foi com estes precedentes que Shigeru Miyamoto, que concebeu o jogo do '*Super Mario Bros'* e '*The Legend of Zelda'* (Techmundo, 2009).

Os anos 90, permitiram a transição de videojogos de pixéis para gráficos 3D, pela popularidade das consolas, de jogos como '*Mortal Combat'* e '*Night Trap'*, e do maior sucesso no Japão (no campo dos jogos), a *Playstation* da *Sony*. A difusão e vulgarização do uso de computadores e o acesso à internet tornou a experiência dos jogos ainda mais imersiva e a atividade mais lúdica dos dias de hoje. A evolução tecnológica e a ascensão do mercado de *Gaming* tornou possível os jogos de realidade virtual, com reconhecimento de voz e movimento, sendo a interação entre o jogador e o jogo o mais intenso desde sempre, com melhores gráficos, capacidade computacional, mais memória e novos equipamentos adicionais (Ingram, 2021).

O conceito de *gaming* surge recentemente e refere-se à atividade de jogar videojogos, tanto em consolas, *tablets*, computadores, telemóveis, ou qualquer outro dispositivo móvel. Esta atividade pode ser praticada tanto online como offline, com apenas um jogador, ou vários, e até em modo profissional (Silva, 2022)

Nas últimas décadas, a evolução tecnológica permitiu que os jogos "passassem de uma simples modulação de sinal de vídeo, até se poder criar um mundo tridimensional credível em tempo real" (Costa, 2015). Esta mudança de paradigma foi promovida pela facilidade que os computadores oferecem para o desenvolvimento de jogos e a sua rápida evolução. O desenvolvimento de videojogos está correlacionada com o desenvolvimento das tecnologias nos computadores.

As **questões de investigação** que serviram de ponto de partida para esta reflexão, foram:

1) Porque é que este género de jogo é apelativo para um específico público-alvo?
2) Como pode ser estimulado novas formas de interação entre o jogador e o jogo?
3) Que tipo de tecnologia e efeitos especiais devem ser utilizados para a maior imersividade e estimulação dos sentidos?

A resposta à primeira questão centra-se nas emoções e no estado que o jogo estimula ao jogador. Neste tipo de jogo, emoções como a adrenalina e o medo, são equiparados a "andar de montanha-russa". Considera-se que esta categoria de jogos seja apelativa a um determinado grupo de indivíduos, e não a totalidade do público.

A segunda e a terceira questão dá ênfase à evolução das novas tecnologias e das novas formas de imersão do jogador no jogo – por exemplo óculos de realidade virtual e outras experiências sensoriais. Especialmente os jogos de terror, devem ser procurar, através de sons, *flashs*, *jumpscares*/sustos, provocar uma reação emotiva ao jogador.

**Géneros de Videojogos**

O tipo de videojogo, ou género é um dos parâmetros que mais contribui no processo de decisão de compra. Embora não haja concordância na definição dessas categorias, M. Wolf (2000) estabeleceu alguns géneros, tendo em consideração alguns fatores como: as características dominantes da experiência, as metas e objetivos dos jogos, a personagem do jogador, os controlos, entre outros, que podem ser combinados com as categorias tipicamente de filmes: terror, aventura/ação, ficção científica.

Algumas categorias são:

- **Abstrato** – jogos com gráficos não representativos, que normalmente envolvem um objetivo que não é orientado para nenhuma narrativa (exemplo: '*Tetris'*, '*Q*bert'*);
- **Adaptação** – jogos baseados em atividades desportivas, jogos de tabuleiro, jogos de carta ou baseados em filmes, bandas desenhadas, séries (exemplo: jogos adaptados de desportos, '*Virtua Tennis'*, '*Fifa'*; jogos adaptados de séries: *'The Simpsons Game'*; jogos adaptados de bandas desenhadas: '*Spiderman'*; '*Teenage Mutant Ninja Turtles'*;
- **Aventura** – jogos cujo o "mundo" é constituído por quartos ou telas múltiplas e conectadas entre si; geralmente os jogos de aventura possuem objetivos necessários para alcançar as diferentes etapas – recolher artefactos, cumprir missões, destrancar portas; as personagens são capazes de carregar objetos, como armas, lanternas, entre outros; os jogos desta categoria distinguem-se dos designados jogos de **Pista de Obstáculos**, que são limitados a determinadas ações – correr, saltar, desviar e atirar (exemplo: '*Donkey Kong'*) e dos **Filmes Interativos**, jogos em que os jogadores não podem explorar de livre vontade o ambiente envolvente e a narrativa segue opções pré-definidas e uma progressão linear (exemplo: '*Dragon's Lair'*);
- **Vida Artificial** – jogos que envolvem a manutenção de criaturas digitais; esta categoria, não deve ser confundida com jogos de **Simulação** (exemplo de jogos de Vida Artificial: '*Pou'*; jogos de Simulação: '*The Sims'*;
- **Estratégia** – jogos que enfatizam o uso de estratégia e não de ações ou reflexos rápidos (exemplo: Xadrez e Damas – aliados ao Jogos de Tabuleiro).

**Videojogos *Indie***

Os videojogos *Indie* têm ganho bastante dimensão na Indústria do '*Gaming*'. De forma geral, é possível distinguir os jogos *indie* dos considerados *mainstreams*, pela sua experiência mais autêntica, superando as limitações técnicas e financeiras inerentes à elaboração de um jogo.

Os jogos indie são caracterizados pelos seguintes fatores (Silva, 2022):

a) São concebidos por pequenas equipas ou até por um único indivíduo;
b) Orçamentos reduzidos;

c) Espírito de DIY ou *Do it Yourself*, espírito de ‘faça você mesmo’;
d) Liberdade criativa, não sofrendo de pressões e desejos externos, como as grandes empresas;

Assim sendo, os jogos *indie* distinguem-se dos demais pela produção independente, com recursos próprios e sem interferências criativas.

**Videojogos de Terror**

A grande diferença dos videojogos de terror e os jogos de outras categorias baseia-se na reação e nas emoções do jogador, enquanto o joga. (Luz, 2010). Os principais aspetos, dos videojogos da categoria de terror, são:

- Atmosfera ou ambiente envolvente;
- Efeitos sonoros e visuais;
- *Jumpscares* ou sustos.

Para a produção deste jogo, os seguintes jogos serviram inspiração:

***1) Phasmophobia***

O ‘*Phasmophobia’* é considerado um jogo de *co-op* (cooperação) online de terror e sobrevivência para quatro pessoas, em perspetiva de primeira pessoa, desenvolvido e publicado pela *Kinetic Games*.

O jogo consiste num grupo de quatro pessoas, no papel de caçadores de fantasmas, contratados para lidar com diferentes entidades, em diferentes habitações, com objetivo de descobrir que tipo de fantasma assombram, através do seu comportamento.

*Figura 1 – Phasmophobia (Guideon's Gaming Reviews, 2020.)*

Este jogo, elaborado em *Unity*, serve de inspiração, pois é considerado um dos melhores jogos *Indie* do momento e oferecendo uma experiência emocionante ao jogador, com várias interações e com um ambiente aterrorizante.

### *2) Slenderman*

É um jogo *single-player*, em perspetiva de primeira pessoa de terror e sobrevivência desenvolvido e publicado pela *Parsec Productions*. Este jogo consiste num único objetivo, encontrar oito páginas numa floresta à noite, tendo só uma lanterna para guiar e sendo perseguido por uma criatura, o '*Slenderman'*. Este jogo, sendo feito em *Unity*, alcançou grande popularidade face ao seu simples objetivo e mecanismo de jogo, tendo também servido para inspiração.

*Figura 2 – Slenderman (WikiHow,2022.)*

### *3) Chased by Darkness*

É um jogo *co-op online* de terror e sobrevivência, para quatro pessoas, em perspectiva de primeira pessoa, desenvolvido e publicado pela *South Blue Moon.* O objetivo deste jogo consiste em explorar o mapa, enfrentar monstros e bani-los. Este jogo serviu de base no seu mecanismo de armas e luta e também no seu objetivo.

Figura 3 – *Chased by Darkness (Game Fabrique, 2020.)*

### *4) Lunch Lady*

É um jogo *co-op online* de terror e sobrevivência para quatro pessoas em perspetiva de primeira pessoa desenvolvido e publicado pela *Manic Vice.* Este jogo consiste em sobreviver à noite na escola e roubar as dez páginas com as perguntas para o teste enquanto é perseguido pelo funcionário. Este jogo serviu de inspiração para fazer um mapa de escola e na sua criação um mapa representando o *campus* da Faculdade.

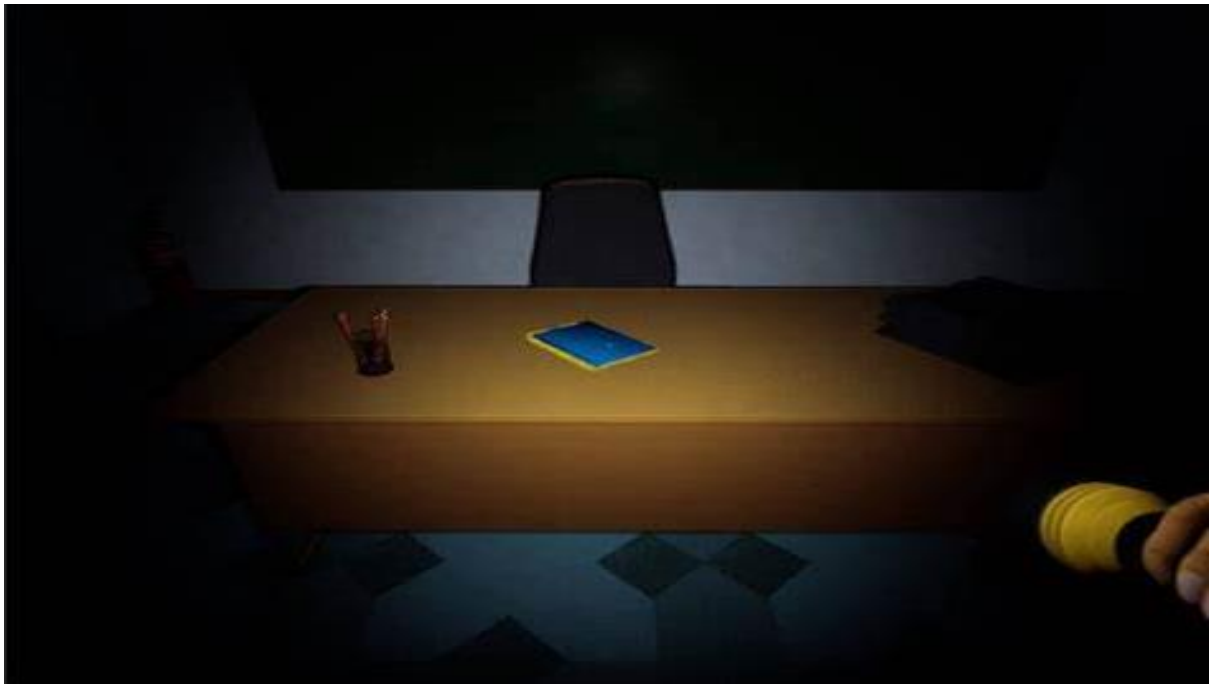

*Figura 4 – Lunch Lady (Steam, 2021)*

# MATERIAIS E MÉTODOS DE DESENVOLVIMENTO

O objeto de estudo deste projeto consistiu na realização de um jogo paranormal, do género de terror, com dimensão 3D, em primeira pessoa, destinando-se a apreciadores desta forma de entretenimento, exclusivo para computador com sistema operativo Windows.

Para elaborar o projeto teve-se, primeiramente, estabelecer o tipo de investigação que o projeto se vai basear e de selecionar as opções mais adequadas ao mesmo. Para saber qual a melhor abordagem e o método científico mais apropriado deve ser considerado o objetivo do trabalho, os procedimentos escolhidos, a finalidade, entre outros.

Quanto à finalidade ou natureza, o tipo de investigação foi a aplicada. A investigação aplicada permite adquirir novos conhecimentos de um determinado fenómeno.

Quanto à abordagem, para este projeto foi optado uma abordagem qualitativa, baseando-se no carácter subjetivo, ou seja a experiência como utilizador. Quanto aos objetivos estipulados são de cariz descritivo, aprofundando melhor os temas de 'Terror' e 'Paranormal'.

O procedimento é etnográfico porque baseia-se na recolha de dados dos comportamentos, caracteristicas e costumes de uma comunidade, neste caso da comunidade '*gamer'*. Além da consulta de fontes documentais e científicas, optou-se pela recolha de dados, através de um inquérito *online*, confidencial na plataforma *Google Forms*, a vinte indivíduos, sobre a temática dos videojogos, em concreto da categoria de terror.

Após seleção do tipo de investigação a seguir, foi estipulado a calendarização de todas as atividades a serem desenvolvidas. O cronograma abaixo ilustra as atividades que foram executadas e o seu período de duração. Estas atividades consistiam em:

- Estipulação dos objetivos, questões de investigação e pesquisa;
- Leitura e Revisão da Literatura;
- Programação do jogo; Resposta às questões de investigação;
- Notas finais e entrega.

*Tabela 1 – Cronograma (Amorim, 2023, p.1)*

| **ATIVIDADES DESENVOLVIDAS** | **2021** | | | **2022** | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Out | Nov | Dez | Jan | Fev | Mar | Abr | Mai | Jun |
| Estipulação dos objetivos, questões de investigação e pesquisa. | | | | | | | | | |
| Leitura e Revisão da Literatura. | | | | | | | | | |
| Programação do jogo; Resposta às questões de investigação. | | | | | | | | | |
| Notas Finais e Entrega. | | | | | | | | | |

Na tabela 2 surgem as questões que permitiram identificar os perfis de jogadores e a sua apetência sobre este tipo de jogos, de modo a entender quais eram os pontos mais importantes. Foram entrevistadas 20 pessoas.

*Tabela 2 - Categorias apuradas para a realização do jogo (Amorim 2023, p.1)*

| Categorias | Questões |
|---|---|
| Perfil de jogadoras/es de jogos de terror | Questão 1: 'Qual a tua idade?'<br>Questão 2: 'Indique o seu género' |
| Interação e frequência de uso das TIC e jogo | Questão 3: 'Costumas jogar no computador?'<br>Questão 4: 'Com que frequência costumas jogar durante a semana?'<br>Questão 5: 'Que tipos de jogos costumas escolher?'<br>Questão 9: ' Fazes stream enquanto jogas?'<br>Questão 10: ' Porque escolheste esta plataforma? (Se aplicável)' |
| Razões de escolha da tipologia/género de jogo | Questão 6: 'Gostas de jogos de terror/paranormal?'<br>Questão 7: 'Gostas de jogos Indie?'<br>Questão 8: ' O que achas mais importante num "bom" jogo de terror/paranormal?' |

No sentido de se perceber com o jogo se vai desenrolar no geral foi efetuado um script do jogo que descreve, nos seus pontos essenciais, cada etapa (Figura 5).

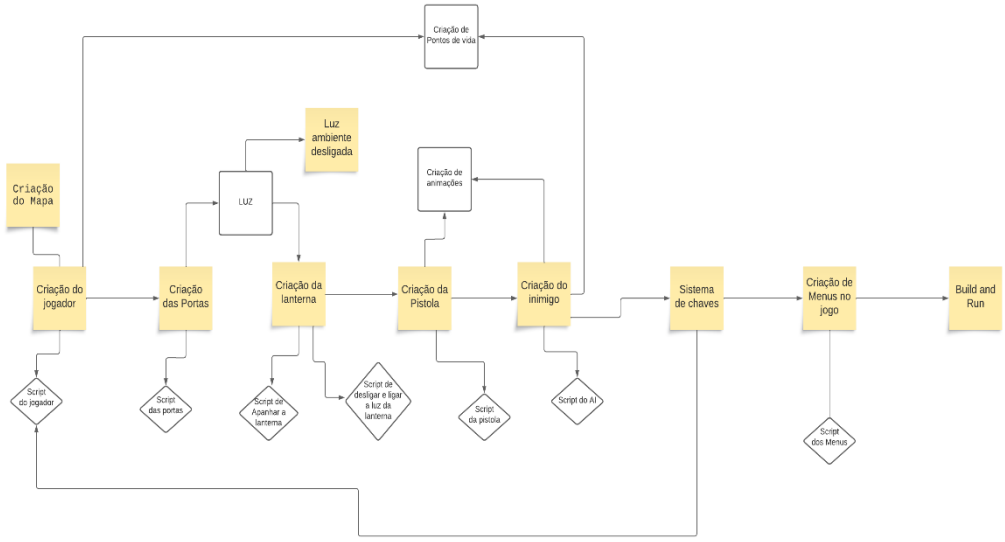


*Figura 5 – Script do jogo 'NIGHT OF THE TERRORBORN' (Amorim, 2023, p.2).*

.

**Storyboard/Sinopse**

1 Abril de 2023 – Um barco russo, sedeado no rio Tejo e vigiado de perto pela Marinha Portuguesa tem um problema no lançamento de mísseis e é lançado, por engano, um míssil em direção à cidade de Lisboa. 12 h, Alameda das Linhas de Torres, 179, ISTEC é atingido pelo míssil. Ninguém está no local, mas todas as sirenes da cidade de Lisboa ficam em alerta. Detritos de edifícios e um grande buraco no solo. Começam a surgir, monstros, espíritos a subir para a luz. Uma mulher encontra-se na entrada, mas procura alguém que supostamente estava lá a trabalhar. Fica sem hipótese de sair, porque a porta do Edifício A do campus encontra-se trancada. Ela vai ter de lutar contra monstros e descobrir como se pode salvar, usando uma arma para enfrentar os inimigos. Ela tinha medo de tudo mas faz tudo para se salvar. O jogo acaba quando finalmente encontra as dez chaves para abrir a porta.

No próximo capítulo serão detalhados ao pormenor as escolhas efetuadas, mediante os resultados obtidos e as diferentes etapas prosseguidas nas criação de cada ecrã do jogo. Em termos éticos e da privacidade não foram colocados quaisquer informações que possam prejudicar alguém ou alguma instituição. O jogo foi identificado devidamente e com ele a atribuição da faixa etária mínima para o jogar.

# RESULTADOS E DISCUSSÃO DE RESULTADOS

O presente capítulo ilustra todo o processo para a elaboração do jogo de terror '***The Night of Terrorborn'*** e o inquérito elaborado, no início da investigação, serviu para identificar quais os pontos cruciais a focar no desenvolvimento e design, assim como nas interfaces e na sua relação com o utilizador.

O inquérito consistiu em dez perguntas sobre o tema dos jogos, em particular os jogos de terror (Tabela 2). As respostas obtidas foram as seguintes:

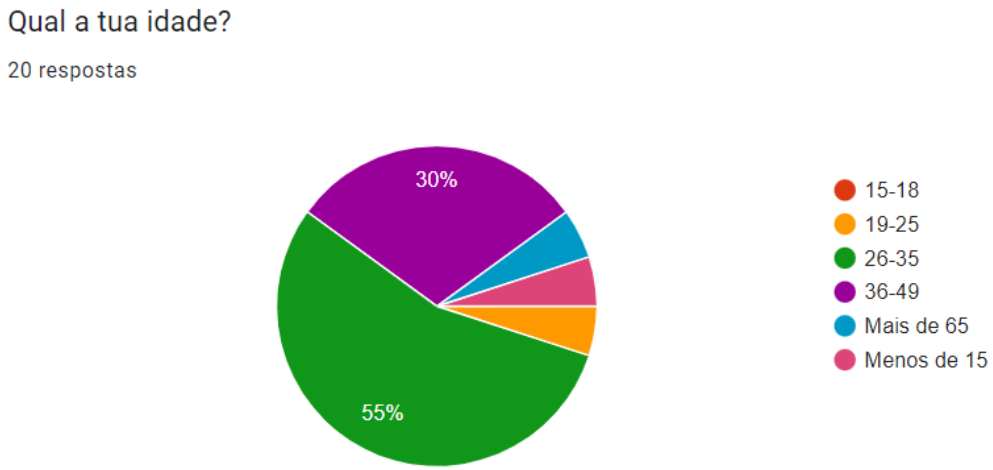


*Figura 6 – Faixas etárias dos entrevistados (Amorim, 2023, p.2).*

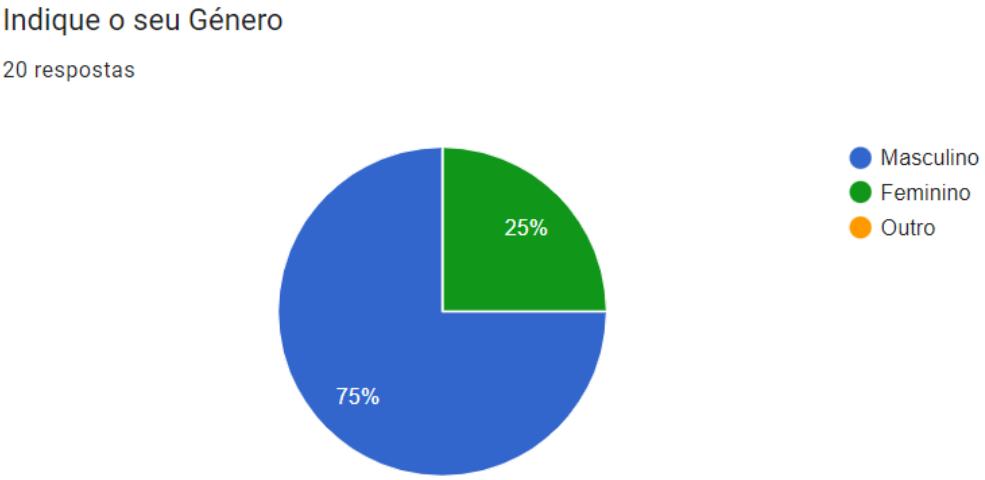


*Figura 7 – Prevalência de género (Amorim, 2023, p.3),*

*Figura 8 – Identificação do dispositivo de jogo (Amorim, 2023, p.3).*

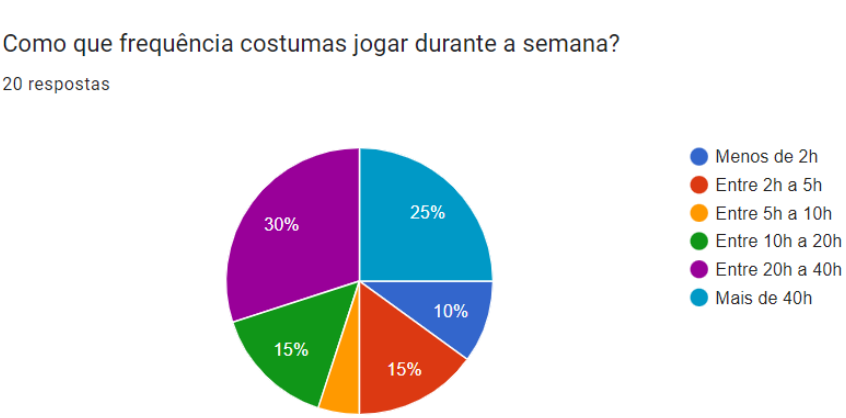


*Figura 9 – Frequência de tempo de jogo (Amorim, 2023, p.4).*

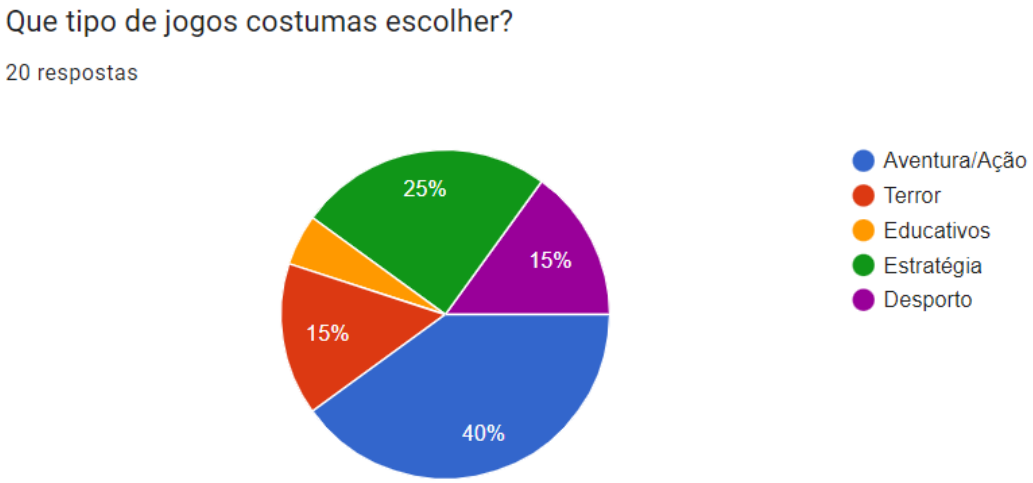


*Figura 10 – Tipologias de género de jogos (Amorim, 2023, p.4).*

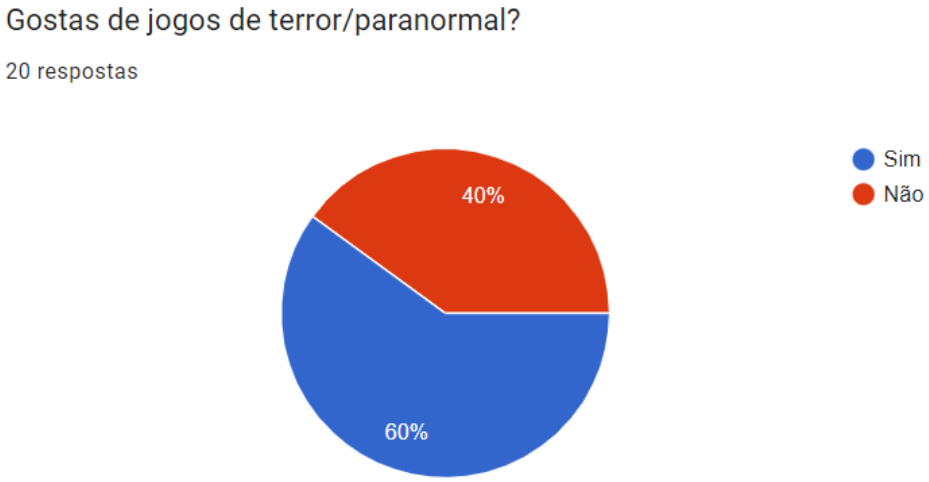


*Figura 11 – Apetência pelo género Terror (Amorim, 2023, p.5).*

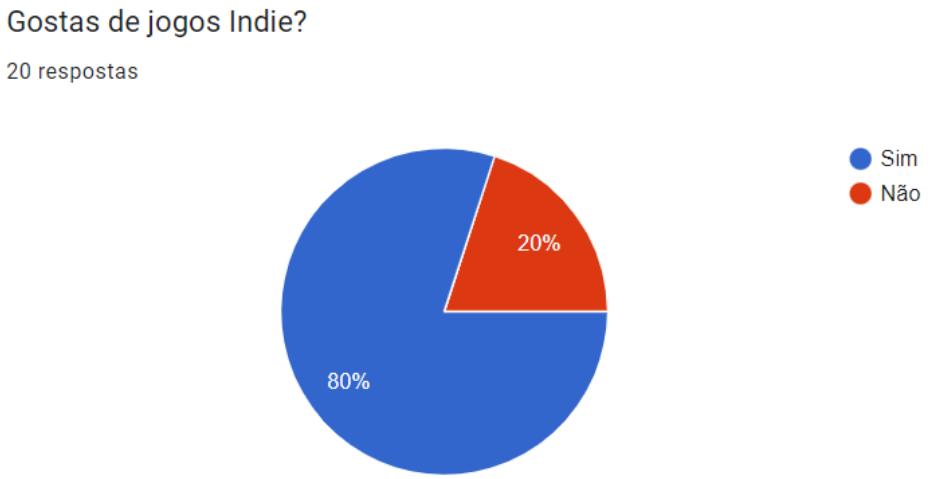


*Figura 12 – Especificidade do tipo de jogo Indie (Amorim, 2023, p.5).*

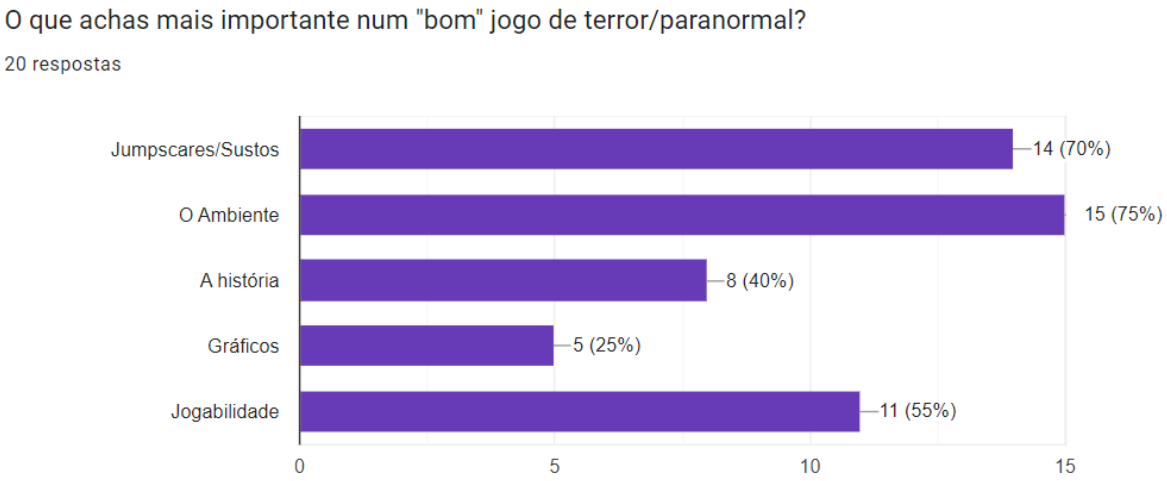


*Figura 13 – Características de um bom jogo de terror (Amorim, 2023, p.6).*

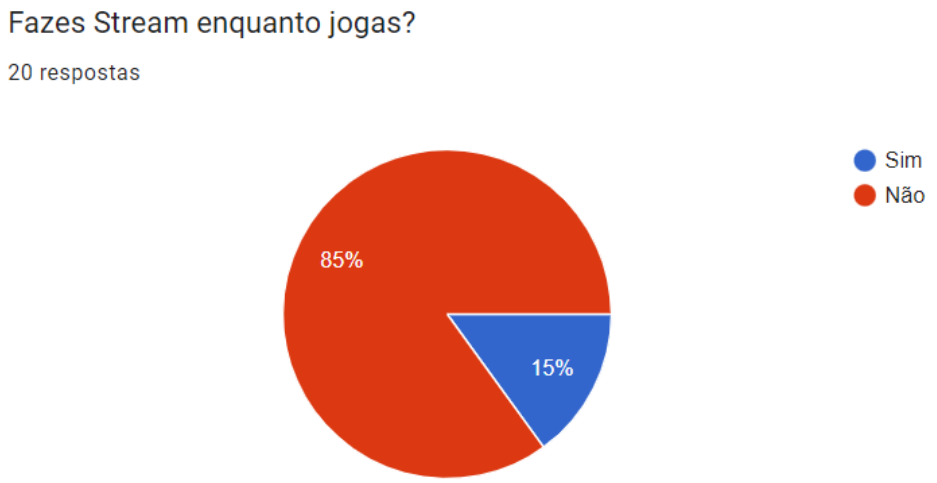


*Figura 14 – Modo de jogar (Amorim, 2023, p.6).*

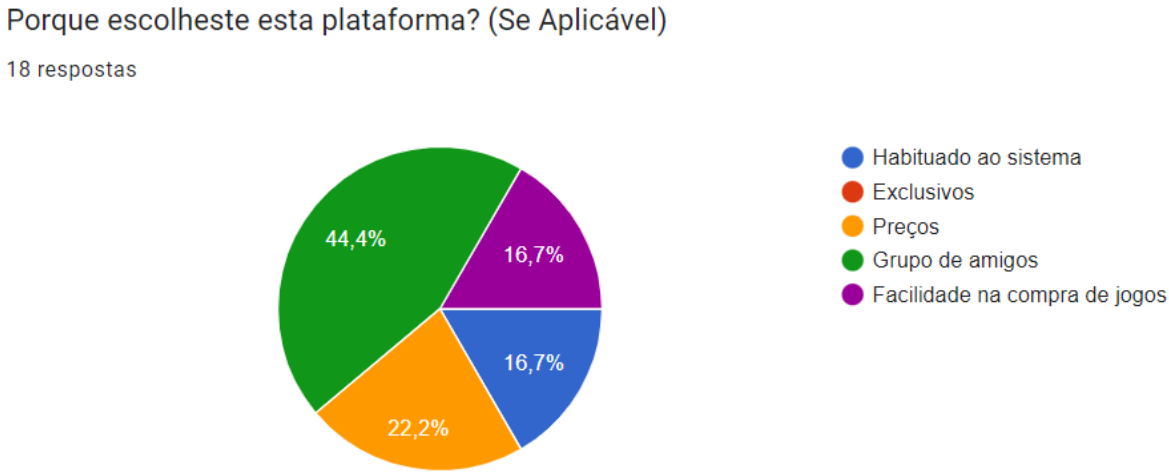


*Figura 15 – Se utiliza habitualmente um repositório de jogos (Amorim, 2023, p.7).*

Estas perguntas revelaram ser de extrema importância para a elaboração do jogo. Teve-se em conta as opiniões dos inquiridos para aperfeiçoar a narrativa, o *design* e outros fatores da jogabilidade. Na questão do Stream (Streaming é uma forma de distribuição digital que permite partilhar conteúdo multimédia pela internet (Gogoni, 2023)), foi evidente a esmagadora maioria que o adopta nas suas práticas de jogo.

**Passos da Elaboração do jogo.**

Primeiro passo consistiu em realizar o mapa do jogo. Foi escolhido um mapa, com base nos elementos arquitetónicos do Edifício A do campus do Lumiar do ISTEC, com algumas mudanças, de modo a tornar o mapa mais acessível as necessidades do jogo.

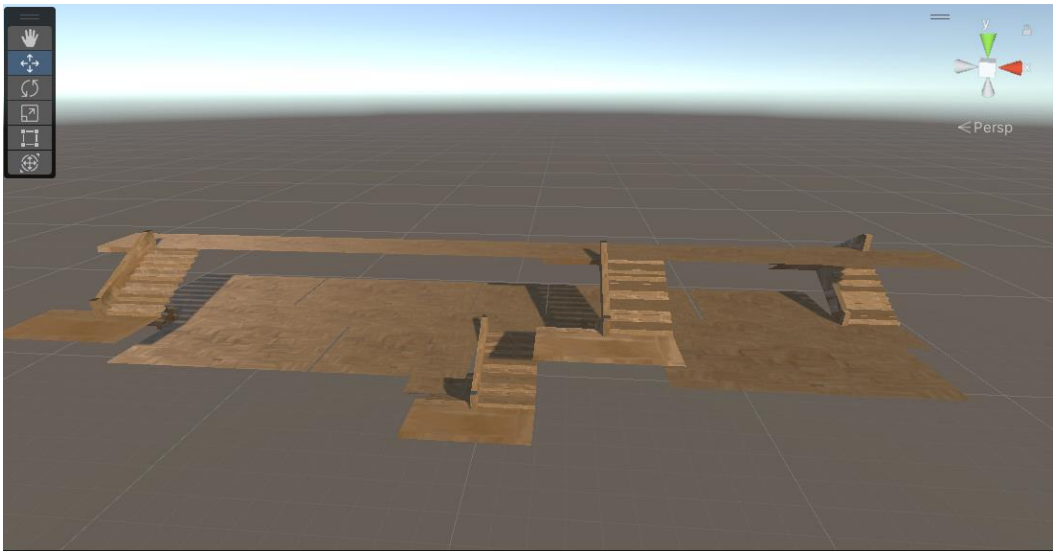


*Figura 16 - Planta do jogo (Amorim, 2023, p.7)*

Ao ter com fonte inspiração o Edifício, apresenta semelhanças nas salas e nas casas de banho.

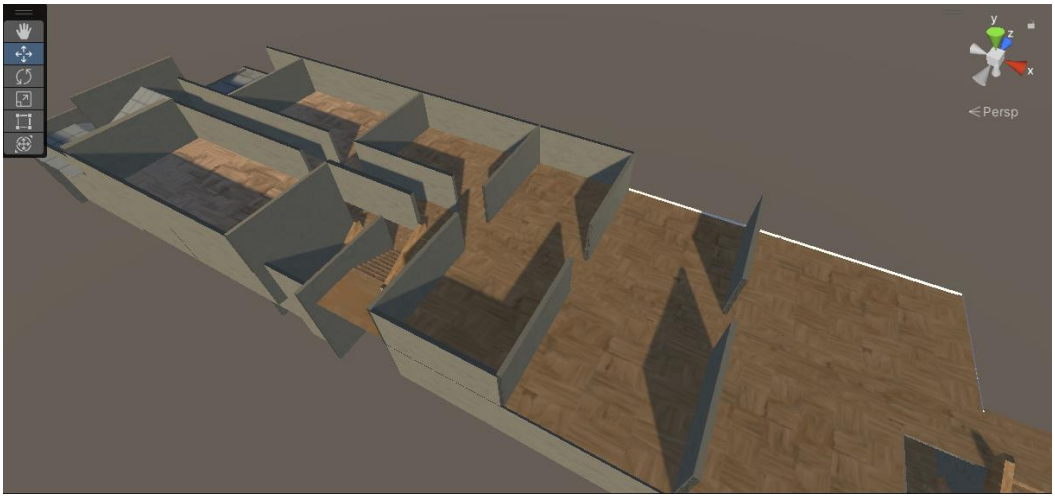


*Figura 17 - Planta das salas (Amorim, 2023, p.8)*

Após a elaboração da planta global, as salas de aula foram mobiladas com objetos antiquados, dando um aspecto mais sombrio.

*Figura 18 - Sala de Aula (Amorim, 2023, p.8)*

Também as texturas neste ambiente permitiram a representação de um ambiente sinistro.

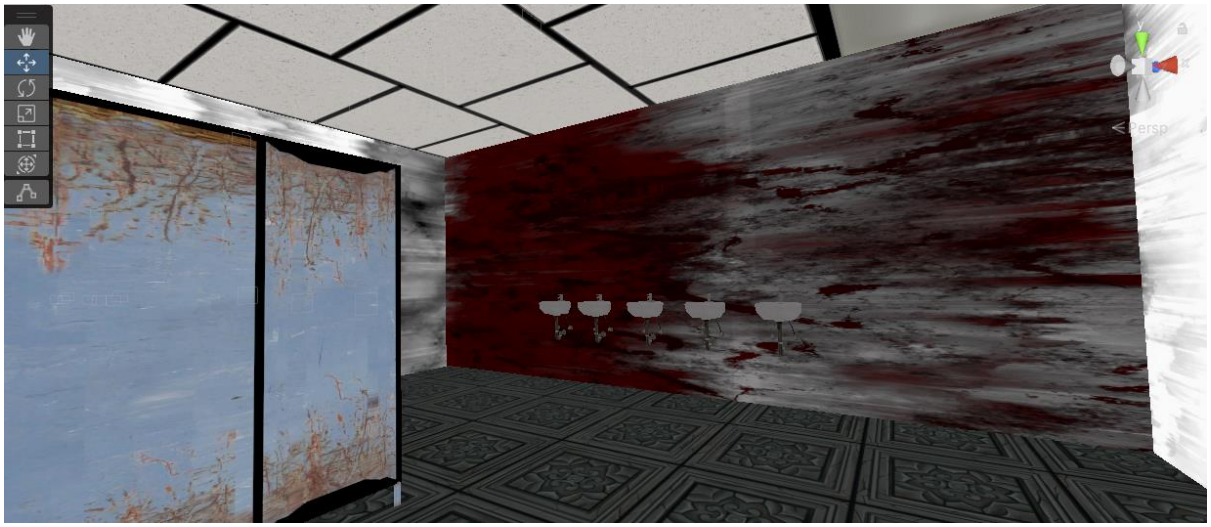

*Figura 19 - Casa de banho (Amorim, 2023, p.9)*

Em seguida foi executado a estrutura da personagem em primeira pessoa.

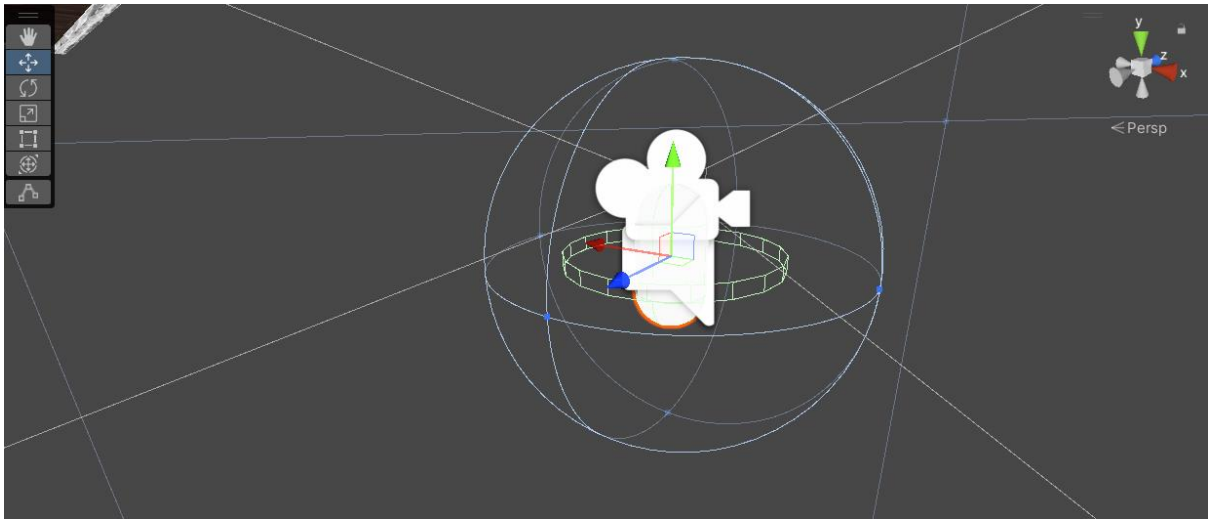


*Figura 20 - Player (Amorim, 2023, p.9)*

***Script*:**

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

[RequireComponent(typeof(CharacterController))]

public class SC_FPSController : MonoBehaviour
{
    public float walkingSpeed = 7.5f;
    public float runningSpeed = 11.5f;
    public float jumpSpeed = 8.0f;
    public float gravity = 20.0f;
    public Camera playerCamera;
    public float lookSpeed = 2.0f;
    public float lookXLimit = 45.0f;
    public int collectedKeys;
    public int require;
    CharacterController characterController;
    Vector3 moveDirection = Vector3.zero;
    float rotationX = 0;

    [HideInInspector]
    public bool canMove = true;

    void Start()
    {
        characterController = GetComponent<CharacterController>();


        Cursor.lockState = CursorLockMode.Locked;
        Cursor.visible = false;
        InGameManager.instance.KeyUpdate(collectedKeys, require);
```

```
    }

    void Update()
    {

        Vector3 forward = transform.TransformDirection(Vector3.forward);
        Vector3 right = transform.TransformDirection(Vector3.right);

        bool isRunning = Input.GetKey(KeyCode.LeftShift);
        float curSpeedX = canMove ? (isRunning ? runningSpeed : walkingSpeed) *
Input.GetAxis("Vertical") : 0;
        float curSpeedY = canMove ? (isRunning ? runningSpeed : walkingSpeed) *
Input.GetAxis("Horizontal") : 0;
        float movementDirectionY = moveDirection.y;
        moveDirection = (forward * curSpeedX) + (right * curSpeedY);

        if (Input.GetButton("Jump") && canMove && characterController.isGrounded)
        {
            moveDirection.y = jumpSpeed;
        }
        else
        {
            moveDirection.y = movementDirectionY;
        }


        if (!characterController.isGrounded)
        {
            moveDirection.y -= gravity * Time.deltaTime;
        }


        characterController.Move(moveDirection * Time.deltaTime);


        if (canMove)
        {
            rotationX += -Input.GetAxis("Mouse Y") * lookSpeed;
            rotationX = Mathf.Clamp(rotationX, -lookXLimit, lookXLimit);
            playerCamera.transform.localRotation = Quaternion.Euler(rotationX, 0, 0);
            transform.rotation *= Quaternion.Euler(0, Input.GetAxis("Mouse X") *
lookSpeed, 0);
        }
    }
    private void OnCollisionEnter(Collision collision)
    {
        if (collision.gameObject.CompareTag("Key"))
        {
            Destroy(collision.gameObject);
            collectedKeys++;
            InGameManager.instance.KeyUpdate(collectedKeys, require);


            if (collectedKeys == require)
            {

                InGameManager.instance.YouWon();
            }
        }
    }
}
```

Em cima é apresentado o *script* que possibilitou a criação do jogador, os seus movimentos, a sua velocidade, gravidade e visão de primeira pessoa.

O seguinte elemento foi a animação das portas.

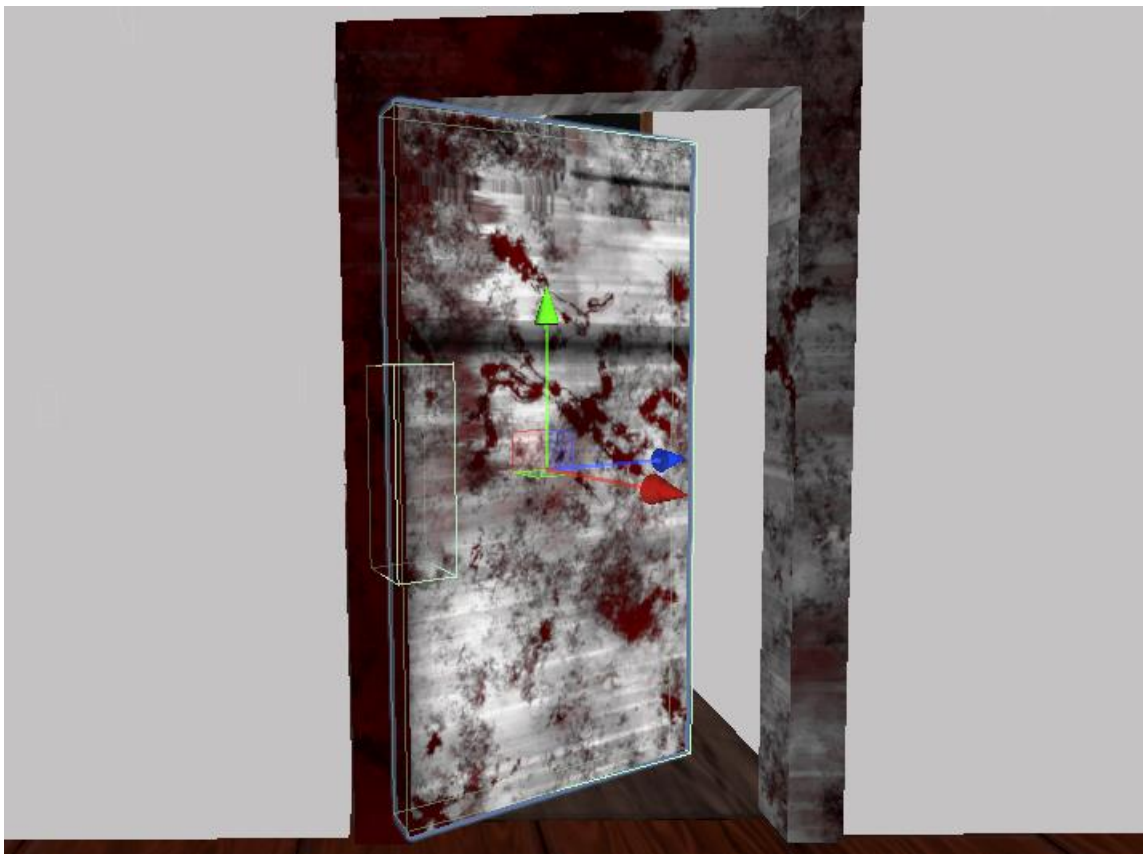

*Figura 21 - Porta (Amorim, 2023, p.10)*

***Script*:**

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

public class OpenDOOR : MonoBehaviour {

    public float TheDistance;
    public GameObject ActionDisplay;
    public GameObject ActionText;
    public GameObject TheDoor;
    public AudioSource CreakSound;

       void Update()
       {
              TheDistance = NewBehaviourScript.DistanceFromTarget;
       }
```

```
	void OnMouseOver()
	{
		if (TheDistance <= 2)
		{
			ActionDisplay.SetActive(true);
			ActionText.SetActive(true);
		}
		if (Input.GetButtonDown("Action"))
		{
			if (TheDistance <= 2)
			{
				this.GetComponent<BoxCollider>().enabled = false;
				ActionDisplay.SetActive(false);
				ActionText.SetActive(false);
				TheDoor.GetComponent<Animation>().Play("OpenDoor");
				CreakSound.Play();
			}
		}
	}

	void OnMouseExit()
	{
		ActionDisplay.SetActive(false);
		ActionText.SetActive(false);
	}
}
```

Em cima está apresentado o *script* que permite a abertura das portas das salas no mapa, onde foi adicionado a distancia necessária para interação do jogador com a porta, a animação de abertura, o som e o texto.

Tendo sido tambem preciso adicionar essa interação com um novo *script* ao jogador.

***Script*:**

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

public class NewBehaviourScript : MonoBehaviour {

	public static float DistanceFromTarget;
	public float ToTarget;


	void Update() {
		RaycastHit Hit;
		if (Physics.Raycast(transform.position,
transform.TransformDirection(Vector3.forward), out Hit)) {
			ToTarget = Hit.distance;
			DistanceFromTarget = ToTarget;
		}
	}
}
```

A luz ambiente foi retirada.

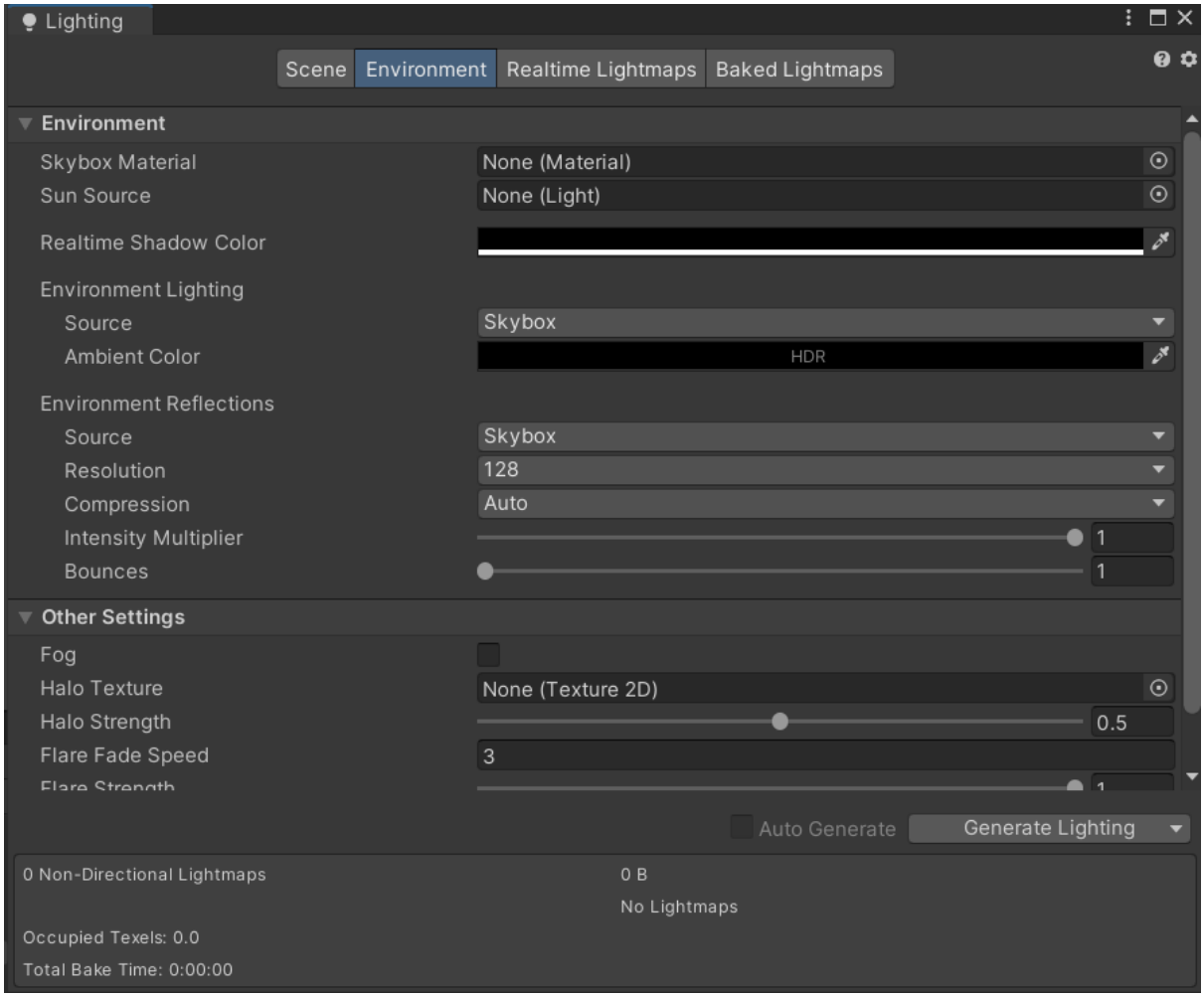


*Figura 22 - Luz ambiente (Amorim, 2023, p.10)*

Em seguida foi adicionada uma lanterna utilizável ao jogador, com mecanismo de ligar e desligar a luz.

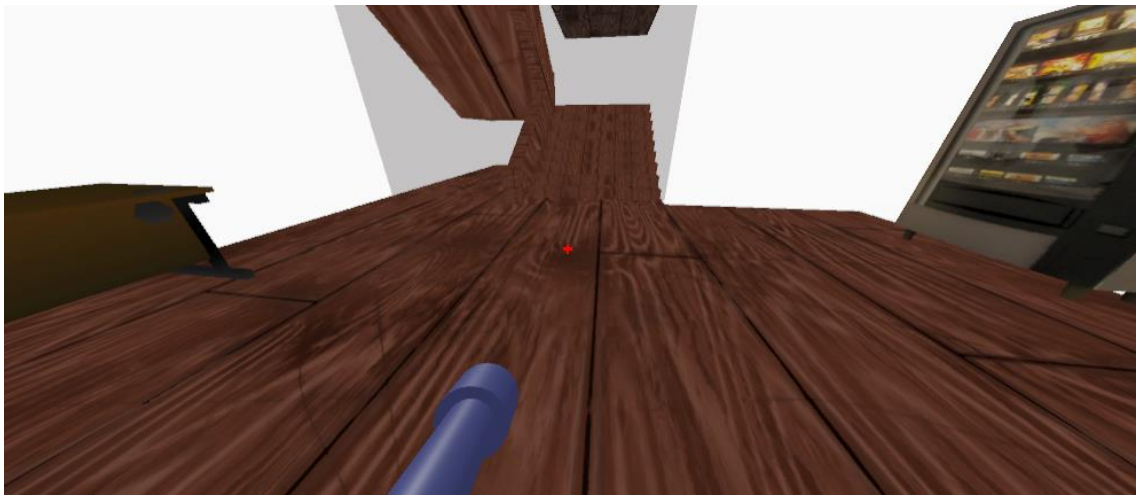

*Figura 23 . Lanterna (Amorim, 2023, p.11)*

***Script*1:**

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

public class PickupFlashlight : MonoBehaviour
{
    public GameObject inttext, flashlight_table, flashlight_hand;
    public AudioSource pickup;
    public bool interactable;
    public GameObject TheJumpTrigger;

    void OnTriggerStay(Collider other)
    {
        if (other.CompareTag("MainCamera"))
        {
            inttext.SetActive(true);
            interactable = true;
            TheJumpTrigger.SetActive(true);
        }
    }
    void OnTriggerExit(Collider other)
    {
        if (other.CompareTag("MainCamera"))
        {
            inttext.SetActive(false);
            interactable = false;
        }
    }
    void Update()
    {
        if (interactable == true)
        {
            if (Input.GetKeyDown(KeyCode.E))
            {
                inttext.SetActive(false);
                interactable = false;
                //pickup.Play();
                flashlight_hand.SetActive(true);
                flashlight_table.SetActive(false);
            }
        }
    }
}
```

***Script*2:**

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

public class flashlight : MonoBehaviour
{
    public GameObject light;
    public bool toggle;
    public AudioSource source;
    public AudioClip[] clips;
    void Start()
    {
        if (toggle == false)
        {
```

```
            light.SetActive(false);
        }
        if (toggle == true)
        {
            light.SetActive(true);
        }
    }

    void Update()
    {
        if (Input.GetKeyDown(KeyCode.F))
        {
            toggle = !toggle;
            if (toggle == false)
            {
                light.SetActive(false);

            }
            if (toggle == true)
            {
                light.SetActive(true);

            }
        }
    }
}
```

O próximo elemento adicionado foi a arma do jogo. A cruz vermelha representa a mira da arma, auxiliando o jogador na pontaria aos inimigos.

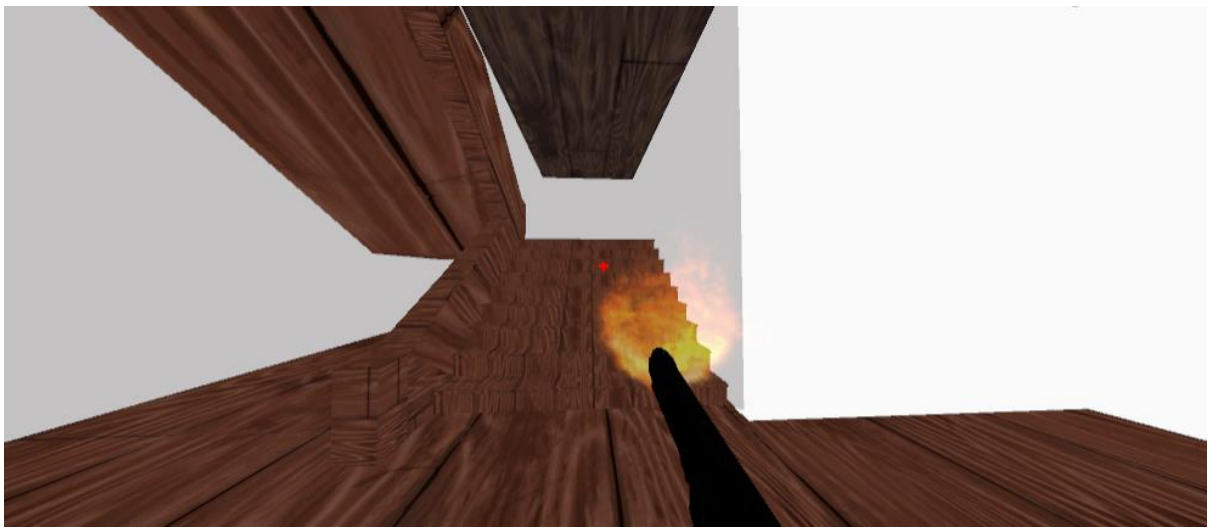

*Figura 24 - Pistola (Amorim, 2023, p.11)*

***Script*:**

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

public class FirePistol : MonoBehaviour
{
    public GameObject TheGun;
    public GameObject MuzzleFlash;
    public AudioSource GunFire;
    public bool IsFiring = false;
    public float TargetDistance;
    public int DamageAmount = 5;
    public int hitDemage = 5;
    void Update()
    {
        if (Input.GetButtonDown("Fire1"))
        {
            if (IsFiring == false)
            {
                StartCoroutine(FiringPistol());
            }
        }

    }

    IEnumerator FiringPistol()
    {
        RaycastHit Shot;
        IsFiring = true;
        if (Physics.Raycast(transform.position,
transform.TransformDirection(Vector3.forward), out Shot))
        {
            TargetDistance = Shot.distance;
            Shot.transform.SendMessage("DamageZombie", DamageAmount,
SendMessageOptions.DontRequireReceiver);
        }
        if (Shot.transform.gameObject.GetComponent<ZombieDeath>())
        {
            Shot.transform.gameObject.GetComponent<ZombieDeath>().Health(hitDemage);
        }
        TheGun.GetComponent<Animation>().Play("PistolShot");
        MuzzleFlash.SetActive(true);
        MuzzleFlash.GetComponent<Animation>().Play("MuzzleAnim");
        GunFire.Play();
        yield return new WaitForSeconds(0.5f);
        IsFiring = false;
    }
}
```

O *script* acima permite adicionar  simulação do disparo com a sua animação, som, dano e o tempo entre cada disparo.

Logo após foi adicionado o inimigo do mapa.

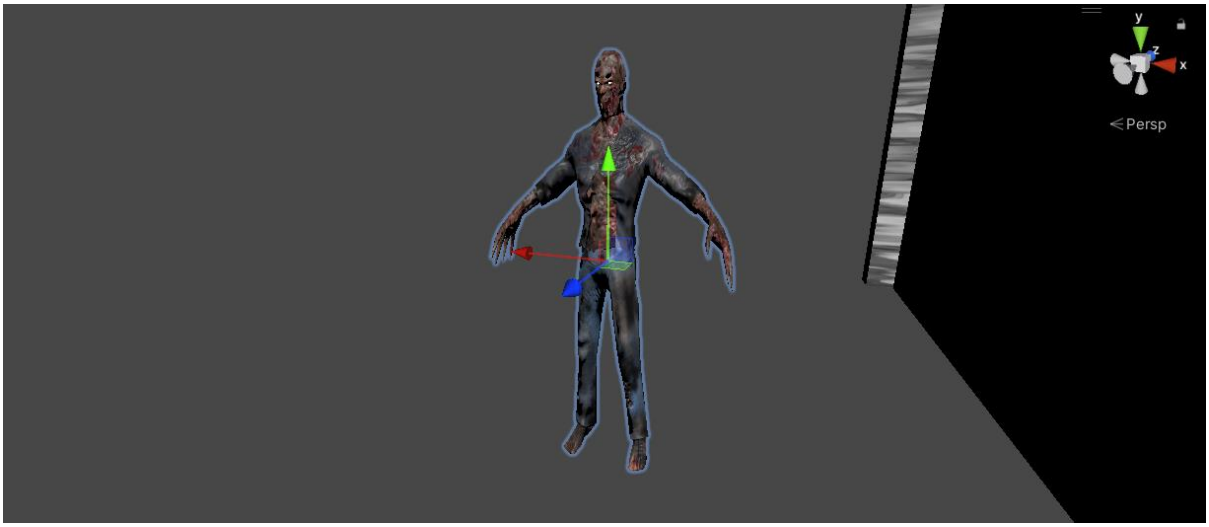


*Figura 25 - Inimigo (Amorim, 2023, p.12)*

### *Script*1:

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

public class ZombieAI : MonoBehaviour
{

    public GameObject thePlayer;
    public GameObject theEnemy;
    public float enemySpeed = 0.01f;
    public bool attackTrigger = false;
    public bool isAttacking = false;
    public Animator anim;
    public float chaseDistance;
    public float attackDistance;
    [SerializeField] int demagePlayer = 2;
    void Update()
    {
        transform.LookAt(thePlayer.transform);

        if (Vector2.Distance(transform.position, thePlayer.transform.position) <=
chaseDistance)
        {
            enemySpeed = 0.01f;
            anim.SetBool("Walk", true);
            transform.position = Vector3.MoveTowards(transform.position,
thePlayer.transform.position, enemySpeed);
        }

        if (Vector2.Distance(transform.position, thePlayer.transform.position) <=
attackDistance)
        {
            enemySpeed = 0;
            anim.SetBool("Walk", false);
            anim.SetBool("Attack", true);
            StartCoroutine(InflictDamage());
        }

    }
```

```
    private void OnCollisionEnter(Collision collision)
    {
        if (collision.gameObject.GetComponent<GlobalHealth>())
        {
            collision.gameObject.GetComponent<GlobalHealth>().Health(demagePlayer);
        }
    }



    IEnumerator InflictDamage()
    {
        isAttacking = true;
        yield return new WaitForSeconds(1.1f);

        yield return new WaitForSeconds(0.2f);
        isAttacking = false;
    }

}
```

***Script*2:**

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

public class ZombieDeath : MonoBehaviour
{

    public int EnemyHealth = 20;
    public GameObject TheEnemy;
    public int StatusCheck;
    public Animator anim;
    void DamageZombie(int DamageAmount)
    {
        EnemyHealth -= DamageAmount;
    }



    public void Health(int demage)
    {
        EnemyHealth -= demage;
        if (EnemyHealth <= 0)
        {
            Destroy(this.gameObject);
        }
    }
```

O primeiro *script* formou os movimentos e animações do inimigo. No segundo *script* foi adicionado a barra de vida do inimigo. Depois de ter sido posto a barra de vida do inimigo, foi posto a barra de vida no jogador.

***Script*:**

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

public class GlobalHealth : MonoBehaviour
{

    public int currentHealth = 20;
    public int internalHealth;

    public void Health(int demnage)
    {
        currentHealth -= demnage;
        if (currentHealth <= 0)
        {
            Debug.Log("Player is Dead");

            InGameManager.instance.YouDie();
        }
    }
}
```

Outro elemento crucial para a narrativa do jogo foi a criação do sistema de chaves.

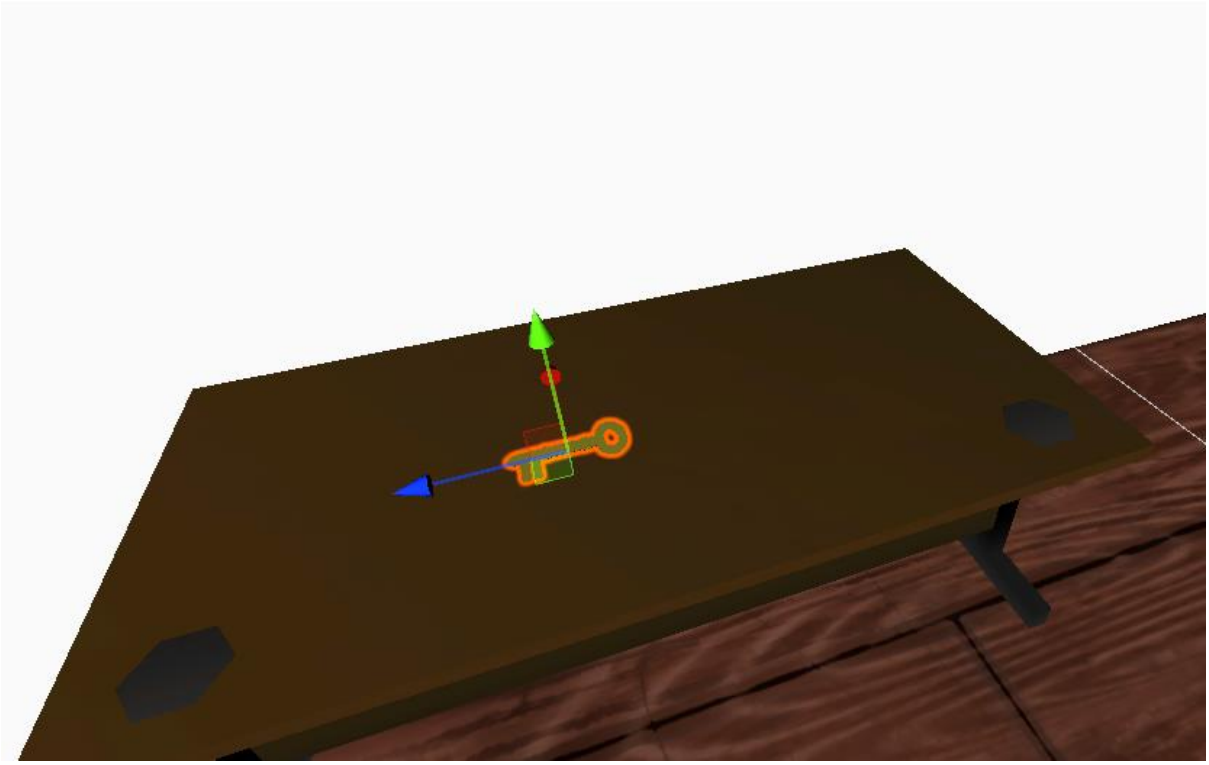

*Figura 26 - Chave (Amorim, 2023, p.12)*

Em que foi necessário adicionar um *script* ao *player* com colisão e incremento para adicionar as chaves até ao objetivo especificado.

***Script*:**

```
private void OnCollisionEnter(Collision collision)
    {
        if (collision.gameObject.CompareTag("Key"))
        {
            Destroy(collision.gameObject);
            collectedKeys++;
            InGameManager.instance.KeyUpdate(collectedKeys, require);


            if (collectedKeys == require)
            {

                InGameManager.instance.YouWon();
            }
```

Para finalizar o jogo foi criado e adicionado o menu de início e outros dois, consoante os dois possíveis cenários.

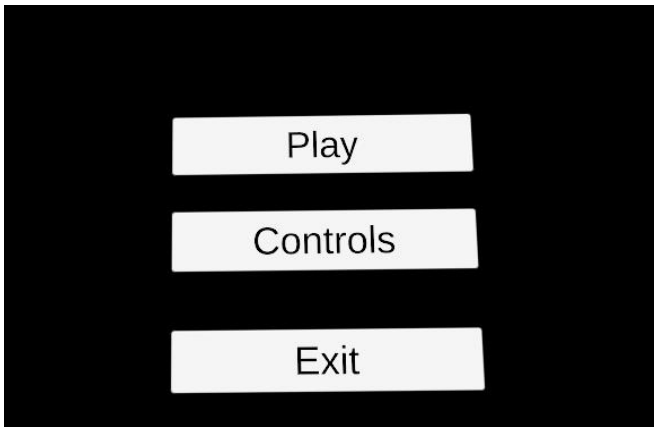


*Figura 27 - Menu Inicial (Amorim, 2023, p.13)*

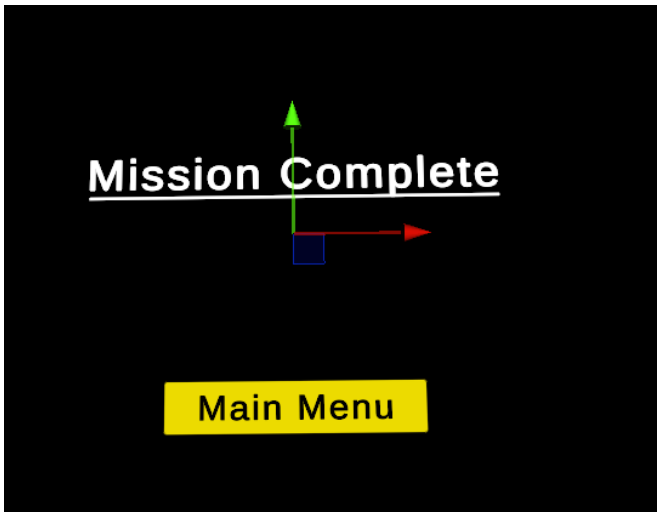


*Figura 28 - Vitória (Amorim, 2023, p.13)*

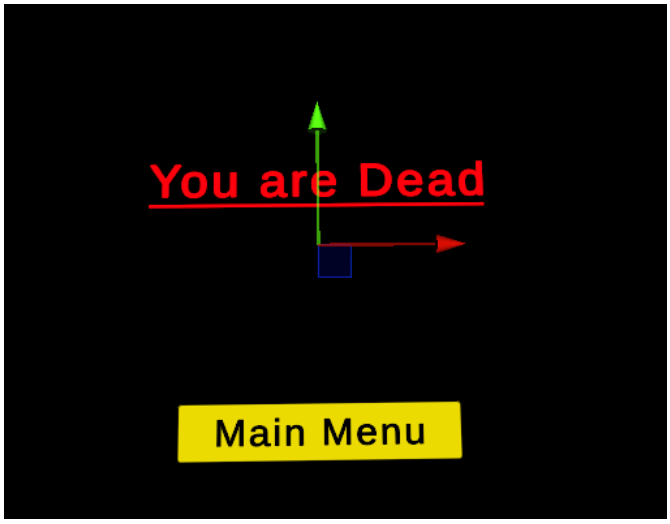


*Figura 29 - Derrota (Amorim, 2023, p.14)*

***Script*:**

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;
using TMPro;
public class InGameManager : MonoBehaviour
{
    public static InGameManager instance;

    public TextMeshProUGUI keys;
    public GameObject deadPanel;
    public GameObject wonPanel;
    public GameObject pausePanel;
    private void Awake()
    {
        instance = this;
    }
    private void Update()
    {
        if(Input.GetKeyDown(KeyCode.Escape))
        {
            if(pausePanel.activeSelf)
            {
                Continue();
            }
            else
            {
                pausePanel.SetActive(true);
                Time.timeScale = 0;
            }

        }
    }
    public void Continue()
    {
        pausePanel.SetActive(false);
        Time.timeScale = 1;
```

```
    }
    public void MainMenu()
    {
        SceneManager.LoadScene("GameScene");
    }
    public void KeyUpdate(int got, int require)
    {
        keys.text = got.ToString() + "/" + require;
    }
    public void YouDie()
    {
        deadPanel.SetActive(true);
    }
    public void YouWon()
    {
        wonPanel.SetActive(true);
    }
}
```

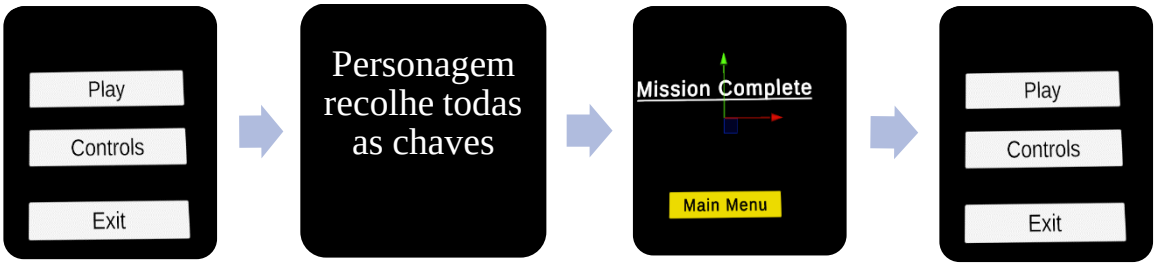


*Figura 30 - Cenário de Vitória e respetivos menus (Amorim, 2023, p. 14).*

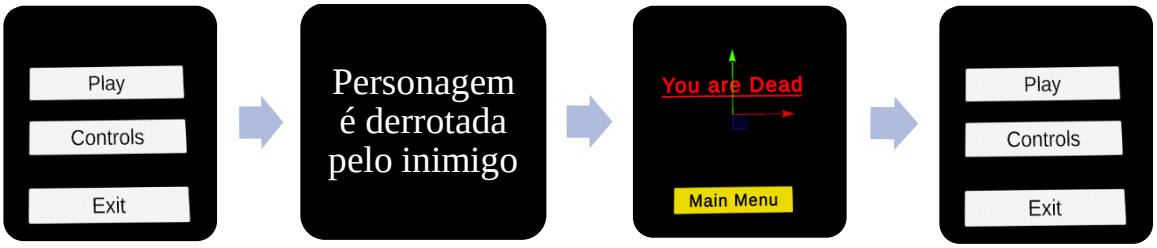


*Figura 31 - Cenário de Derrota e respetivos menus (Amorim, 2023, p.15).*

# CONCLUSÃO

O Projeto Académico teve como principal objetivo a conceção e desenvolvimento de um jogo paranormal, da categoria de terror, a três dimensões, com perspetiva de primeira pessoa. No início do projeto foram delineadas questões de investigação que foram respondidas com sucesso ao longo do trabalho e objetivos específicos que foram cumpridos.

O trabalho segue uma metodologia de investigação, do tipo aplicada, já que foram aprofundados conhecimentos de fenómenos já conhecidos. Além de fontes documentais e científicas, foi elaborado um inquérito online, a vinte indivíduos, sobre a temática dos jogos de computador e jogos de computador. As questões do inquérito permitiram conhecer um pequeno universo de indivíduos, os seus gostos e opiniões acerca do tema em estudo.

O jogo elaborado denominou-se como '***The Night of Terrorborn'***, cuja narrativa centra-se na exploração do Edifício A do *Campus* do Lumiar do ISTEC, com auxílio de uma lanterna e arma, evitando ser assassinado pelos monstros. Através de sons e de um ambiente obscuro, o jogo permite a imersividade do jogador. Este jogo foi elaborado no software *Unity,* na linguagem #C, tal como variados jogos *Indie*.

Os jogos *Indie*, jogos em ascensão nos dias de hoje, são executados com pequenos orçamentos e por uma ou pequenas equipas. O jogo '***The Night of Terrorborn***' foi elaborado no âmbito académico, sem qualquer orçamento e executado por apenas uma pessoa.

Apesar dos objetivos terem sido cumpridos são propostas medidas para a melhoria, na possibilidade de trabalhos futuros, tais como:

- Melhorar detalhes de texturas do mapa;
- Melhorar o AI do inimigo;
- Adicionar *jumpscares*, para maior imersividade;
- Melhoria dos menus de acesso.

# REFERÊNCIAS BIBLIOGRÁFICAS


*A história dos video games: do osciloscópio aos gráficos 3D*. (n.d.). Www.tecmundo.com.br. https://www.tecmundo.com.br/xbox-360/3236-a-historia-dos-video-games-do-osciloscopio-aos-graficos-3d.htm

Batista, Mônica; Quintão, Patrícia; Lima, Sérgio; Campos, Luciana; Batista, T. (2007). UM ESTUDO SOBRE A HISTÓRIA DOS JOGOS ELETRÔNICOS. *Revista Eletrônica Da Faculdade Metodista Granbery*, *3*. https://d1wqtxts1xzle7.cloudfront.net/60207130/Artigo_-_Um_estudo_sobre20190805-80467-mtavau-libre.pdf?1565004547=&response-content-disposition=inline%3B+filename%3DUM_ESTUDO_SOBRE_A_HISTORIA_DOS_JOGOS_ELE.pdf&Expires=1688505717&Signature=KQj5c1ZD4Cql3A8e

Barroso, H. (2020). *Videojogos e Imersão: O caso dos Jogos Sérios. Universidade da Beira Interior*

Cruz, G. R. da, Lima, M. R. de, & Nascimento, S. S. do. (2020). Jogos Eletrônicos na Formação de Professores: Uma Revisão Sistemática no Portal de Periódicos da Capes. Teoria E Prática Da Educação, 23(2), 117-141. https://doi.org/10.4025/tpe.v23i2.52971

Costa, H. (2015). *A Evolução Gráfica nos Jogos de Computador. As idades do desenho. Universidade de Lisboa. Faculdade de Belas–Artes*

*Da primeira à última geração: a evolução dos jogos eletrônicos*. (2021, June 21). Blog Ingram. https://blog.ingrammicro.com.br/gaming/evolucao-dos-jogos-eletrnicos/

Gil, A. C. (2023, April 14). *Indústria dos videojogos deve alcançar os 4 mil milhões de utilizadores em 2024 - JPN*. JPN - JornalismoPortoNet. https://www.jpn.up.pt/2023/04/14/industria-dos-videojogos-deve-alcancar-os-4-mil-milhoes-de-utilizadores-em-2024/

Gamefabrique (2020). *Chased by Darkness*. https://gamefabrique.com/games/chased-by-darkness/#&gid=1&pid=pc/chased-by-darkness-15.png

Guideon´s Gaming Reviews (2020). *Phasmophobia*. https://gideonsgaming.com/phasmophobia-early-access-review/

Guerra, A. R. (2014). Videojogos: a indústria milionária que está a nascer em Portugal. https://www.dinheirovivo.pt/invalidos/videojogos-a-industria-milionaria-que-esta-anascer-em-portugal

*Introducing Newzoo's Gamer Segmentation™: The Key to Understanding, Quantifying, and Reaching the New Era of Game Enthusiasts | Newzoo*. (2018). Newzoo. https://newzoo.com/news/introducing-newzoos-gamer-segmentation-the-key-to-understanding-quantifying-and-reaching-game-enthusiasts-across-the-world/

LUZ, Alan Richard da. Vídeo Games: História, Linguagem e Expressão Gráfica. São Paulo: Blucher, 2010.

Nwezoo. (2019). *Latest market estimates: Key numbers.* https://platform.newzoo.com/key-numbers/.

Paula, Bruno Henrique de & Hildebrand, Hermes Renato. 2014. "Potencial Expressivo nos Jogos Indie e Gamearte: Análise dos jogos Flower e Journey". In Atas do III Encontro Anual da AIM, editado por Paulo Cunha e Sérgio Dias Branco, 298-309. Coimbra: AIM. ISBN 978-989- 98215-1-4

Ronaldo Gogoni. (2023, March 20). *O que é streaming? – Tecnoblog*. Tecnoblog. https://tecnoblog.net/responde/o-que-e-streaming/#:~:text=Streaming%20%C3%A9%20a%20tecnologia%20de%20transmiss%C3%A3o%20de%20dados%20pela%20internet

shamirpt. (n.d.). *O fenómeno Gaming, afinal o que é? - Shamir Optical Portugal*. Www.shamir.pt. Retrieved July 1, 2023, from https://www.shamir.pt/blog/item/6241-fenomeno-gaming

Sackson, S. (1991). *The Book of Classic Board Games. Palo Alto: Klutz*.

Silva, R. (2022). *i- Técnico*. https://www.i-tecnico.pt/gaming-nao-e-apenas-jogar-um-jogo-de-computador-entao-o-que-e-entenda/

Silva, C.S. da., & Soares, M. H. F. B.. (2023). Estudos bibliografia sobre conceito de jogo, cultura lúdica e abordagem de pesquisa em um periódico científico de Ensino de Química. Ciência & Educação (bauru), 29, e23003. https://doi.org/10.1590/1516-731320230003

Wikiwow. (2022). *SlenderMan*. https://iwantitblack.com/product/slender-man-android-app/

# JOGOS DE TERROR

No âmbito da unidade curricular Projeto Académico, da licenciatura de Engenheira Multimédia, foi elaborado o presente questionário para averiguar algumas questões pertinentes para o desenvolvimento de um jogo de terror.

* Indica uma pergunta obrigatória

1. Qual a tua idade? *

*Marcar apenas uma oval.*

- ◯ Menos de 15
- ◯ 15-18
- ◯ 19-25
- ◯ 26-35
- ◯ 36-49
- ◯ Mais de 65

2. Indique o seu Género

*Marcar apenas uma oval.*

- ◯ Masculino
- ◯ Feminino
- ◯ Outro

3. Costumas jogar no computador? *

*Marcar apenas uma oval.*

- ◯ Sim
- ◯ Não

4. Como que frequência costumas jogar durante a semana? *

*Marcar apenas uma oval.*

- ◯ Menos de 2h
- ◯ Entre 2h a 5h
- ◯ Entre 5h a 10h
- ◯ Entre 10h a 20h
- ◯ Entre 20h a 40h
- ◯ Mais de 40h

5. Que tipo de jogos costumas escolher? *

*Marcar apenas uma oval.*

- ◯ Aventura/Ação
- ◯ Terror
- ◯ Educativos
- ◯ Estratégia
- ◯ Desporto
- ◯ Outra: _____

6. Gostas de jogos de terror/paranormal? *

*Marcar apenas uma oval.*

- ◯ Sim
- ◯ Não

7. Gostas de jogos Indie?

*Marcar apenas uma oval.*

- ◯ Sim
- ◯ Não

8. O que achas mais importante num "bom" jogo de terror/paranormal? *

*Marcar tudo o que for aplicável.*

- [ ] Jumpscares/Sustos
- [ ] O Ambiente
- [ ] A história
- [ ] Gráficos
- [ ] Jogabilidade

9. Fazes Stream enquanto jogas? *

*Marcar apenas uma oval.*

- ◯ Sim
- ◯ Não

10. Porque escolheste esta plataforma? (Se Aplicável)

*Marcar apenas uma oval.*

- ◯ Habituado ao sistema
- ◯ Exclusivos
- ◯ Preços
- ◯ Grupo de amigos
- ◯ Facilidade na compra de jogos

Este conteúdo não foi criado nem aprovado pela Google.

Google Formulários

*Scripts*:

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

[RequireComponent(typeof(CharacterController))]

public class SC_FPSController : MonoBehaviour
{
    public float walkingSpeed = 7.5f;
    public float runningSpeed = 11.5f;
    public float jumpSpeed = 8.0f;
    public float gravity = 20.0f;
    public Camera playerCamera;
    public float lookSpeed = 2.0f;
    public float lookXLimit = 45.0f;
    public int collectedKeys;
    public int require;
    CharacterController characterController;
    Vector3 moveDirection = Vector3.zero;
    float rotationX = 0;

    [HideInInspector]
    public bool canMove = true;

    void Start()
    {
        characterController = GetComponent<CharacterController>();


        Cursor.lockState = CursorLockMode.Locked;
        Cursor.visible = false;
        InGameManager.instance.KeyUpdate(collectedKeys, require);
    }

    void Update()
    {

        Vector3 forward = transform.TransformDirection(Vector3.forward);
        Vector3 right = transform.TransformDirection(Vector3.right);

        bool isRunning = Input.GetKey(KeyCode.LeftShift);
        float curSpeedX = canMove ? (isRunning ? runningSpeed : walkingSpeed) *
Input.GetAxis("Vertical") : 0;
        float curSpeedY = canMove ? (isRunning ? runningSpeed : walkingSpeed) *
Input.GetAxis("Horizontal") : 0;
        float movementDirectionY = moveDirection.y;
        moveDirection = (forward * curSpeedX) + (right * curSpeedY);

        if (Input.GetButton("Jump") && canMove && characterController.isGrounded)
        {
            moveDirection.y = jumpSpeed;
        }
        else
        {
            moveDirection.y = movementDirectionY;
        }
```

```
        if (!characterController.isGrounded)
        {
            moveDirection.y -= gravity * Time.deltaTime;
        }


        characterController.Move(moveDirection * Time.deltaTime);


        if (canMove)
        {
            rotationX += -Input.GetAxis("Mouse Y") * lookSpeed;
            rotationX = Mathf.Clamp(rotationX, -lookXLimit, lookXLimit);
            playerCamera.transform.localRotation = Quaternion.Euler(rotationX, 0, 0);
            transform.rotation *= Quaternion.Euler(0, Input.GetAxis("Mouse X") *
lookSpeed, 0);
        }
    }
    private void OnCollisionEnter(Collision collision)
    {
        if (collision.gameObject.CompareTag("Key"))
        {
            Destroy(collision.gameObject);
            collectedKeys++;
            InGameManager.instance.KeyUpdate(collectedKeys, require);


            if (collectedKeys == require)
            {

                InGameManager.instance.YouWon();
            }
        }
    }
}


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

public class OpenDOOR : MonoBehaviour {

    public float TheDistance;
    public GameObject ActionDisplay;
    public GameObject ActionText;
    public GameObject TheDoor;
    public AudioSource CreakSound;

       void Update()
       {
              TheDistance = NewBehaviourScript.DistanceFromTarget;
       }

       void OnMouseOver()
       {
              if (TheDistance <= 2)
              {
                     ActionDisplay.SetActive(true);
```

```
                        ActionText.SetActive(true);
                }
                if (Input.GetButtonDown("Action"))
                {
                        if (TheDistance <= 2)
                        {
                                this.GetComponent<BoxCollider>().enabled = false;
                                ActionDisplay.SetActive(false);
                                ActionText.SetActive(false);
                                TheDoor.GetComponent<Animation>().Play("OpenDoor");
                                CreakSound.Play();
                        }
                }
        }

        void OnMouseExit()
        {
                ActionDisplay.SetActive(false);
                ActionText.SetActive(false);
        }
}



using System.Collections;

using System.Collections.Generic;
using UnityEngine;

public class NewBehaviourScript : MonoBehaviour {

        public static float DistanceFromTarget;
        public float ToTarget;


        void Update() {
                RaycastHit Hit;
                if (Physics.Raycast(transform.position,
transform.TransformDirection(Vector3.forward), out Hit)) {
                        ToTarget = Hit.distance;
                        DistanceFromTarget = ToTarget;
                }
        }
}

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

public class PickupFlashlight : MonoBehaviour
{
    public GameObject inttext, flashlight_table, flashlight_hand;
    public AudioSource pickup;
    public bool interactable;
    public GameObject TheJumpTrigger;

    void OnTriggerStay(Collider other)
    {
        if (other.CompareTag("MainCamera"))
        {
            inttext.SetActive(true);
```

```
            interactable = true;
            TheJumpTrigger.SetActive(true);
        }
    }
    void OnTriggerExit(Collider other)
    {
        if (other.CompareTag("MainCamera"))
        {
            inttext.SetActive(false);
            interactable = false;
        }
    }
    void Update()
    {
        if (interactable == true)
        {
            if (Input.GetKeyDown(KeyCode.E))
            {
                inttext.SetActive(false);
                interactable = false;
                //pickup.Play();
                flashlight_hand.SetActive(true);
                flashlight_table.SetActive(false);
            }
        }
    }
}


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

public class flashlight : MonoBehaviour
{
    public GameObject light;
    public bool toggle;
    public AudioSource source;
    public AudioClip[] clips;
    void Start()
    {
        if (toggle == false)
        {
            light.SetActive(false);
        }
        if (toggle == true)
        {
            light.SetActive(true);
        }
    }

    void Update()
    {
        if (Input.GetKeyDown(KeyCode.F))
        {
            toggle = !toggle;
            if (toggle == false)
            {
                light.SetActive(false);

            }
            if (toggle == true)
```

```
            {
                light.SetActive(true);

            }
        }
    }
}


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

public class FirePistol : MonoBehaviour
{
    public GameObject TheGun;
    public GameObject MuzzleFlash;
    public AudioSource GunFire;
    public bool IsFiring = false;
    public float TargetDistance;
    public int DamageAmount = 5;
    public int hitDemage = 5;
    void Update()
    {
        if (Input.GetButtonDown("Fire1"))
        {
            if (IsFiring == false)
            {
                StartCoroutine(FiringPistol());
            }
        }

    }

    IEnumerator FiringPistol()
    {
        RaycastHit Shot;
        IsFiring = true;
        if (Physics.Raycast(transform.position,
transform.TransformDirection(Vector3.forward), out Shot))
        {
            TargetDistance = Shot.distance;
            Shot.transform.SendMessage("DamageZombie", DamageAmount,
SendMessageOptions.DontRequireReceiver);
        }
        if (Shot.transform.gameObject.GetComponent<ZombieDeath>())
        {
            Shot.transform.gameObject.GetComponent<ZombieDeath>().Health(hitDemage);
        }
        TheGun.GetComponent<Animation>().Play("PistolShot");
        MuzzleFlash.SetActive(true);
        MuzzleFlash.GetComponent<Animation>().Play("MuzzleAnim");
        GunFire.Play();
        yield return new WaitForSeconds(0.5f);
        IsFiring = false;
    }
}



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

```
using UnityEngine;

public class ZombieAI : MonoBehaviour
{

    public GameObject thePlayer;
    public GameObject theEnemy;
    public float enemySpeed = 0.01f;
    public bool attackTrigger = false;
    public bool isAttacking = false;
    public Animator anim;
    public float chaseDistance;
    public float attackDistance;
    [SerializeField] int demagePlayer = 2;
    void Update()
    {
        transform.LookAt(thePlayer.transform);

        if (Vector2.Distance(transform.position, thePlayer.transform.position) <=
chaseDistance)
        {
            enemySpeed = 0.01f;
            anim.SetBool("Walk", true);
            transform.position = Vector3.MoveTowards(transform.position,
thePlayer.transform.position, enemySpeed);
        }

        if (Vector2.Distance(transform.position, thePlayer.transform.position) <=
attackDistance)
        {
            enemySpeed = 0;
            anim.SetBool("Walk", false);
            anim.SetBool("Attack", true);
            StartCoroutine(InflictDamage());
        }

    }

    private void OnCollisionEnter(Collision collision)
    {
        if (collision.gameObject.GetComponent<GlobalHealth>())
        {
            collision.gameObject.GetComponent<GlobalHealth>().Health(demagePlayer);
        }
    }



    IEnumerator InflictDamage()
    {
        isAttacking = true;
        yield return new WaitForSeconds(1.1f);

        yield return new WaitForSeconds(0.2f);
        isAttacking = false;
    }

}
```

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

public class ZombieDeath : MonoBehaviour
{

    public int EnemyHealth = 20;
    public GameObject TheEnemy;
    public int StatusCheck;
    public Animator anim;
    void DamageZombie(int DamageAmount)
    {
        EnemyHealth -= DamageAmount;
    }



    public void Health(int demage)
    {
        EnemyHealth -= demage;
        if (EnemyHealth <= 0)
        {
            Destroy(this.gameObject);
        }
    }



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

public class GlobalHealth : MonoBehaviour
{

    public int currentHealth = 20;
    public int internalHealth;

    public void Health(int demnage)
    {
        currentHealth -= demnage;
        if (currentHealth <= 0)
        {
            Debug.Log("Player is Dead");

            InGameManager.instance.YouDie();
        }
    }
}

using System.Collections;

using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;
using TMPro;
public class InGameManager : MonoBehaviour
{
    public static InGameManager instance;
```

```
    public TextMeshProUGUI keys;
    public GameObject deadPanel;
    public GameObject wonPanel;
    public GameObject pausePanel;
    private void Awake()
    {
        instance = this;
    }
    private void Update()
    {
        if(Input.GetKeyDown(KeyCode.Escape))
        {
            if(pausePanel.activeSelf)
            {
                Continue();
            }
            else
            {
                pausePanel.SetActive(true);
                Time.timeScale = 0;
            }

        }
    }
    public void Continue()
    {
        pausePanel.SetActive(false);
        Time.timeScale = 1;
    }
    public void MainMenu()
    {
        SceneManager.LoadScene("GameScene");
    }
    public void KeyUpdate(int got, int require)
    {
        keys.text = got.ToString() + "/" + require;
    }
    public void YouDie()
    {
        deadPanel.SetActive(true);
    }
    public void YouWon()
    {
        wonPanel.SetActive(true);
    }
}
```