Relatório Técnico

**Space Survivor**

Realizado por: Francisco Duarte

Índice

* Introdução
* Descrição do Jogo
* Estrutura Técnica
* Implementação das Funcionalidades
* Desafios e Soluções
* Testes e Validação
* Conclusão
* Referências

Introdução

Este relatório descreve o desenvolvimento de um jogo criado em Unity, aborda as etapas principais do projeto, as mecânicas implementadas e as soluções técnicas adotadas. O objetivo do projeto foi criar um jogo funcional e interativo que combina movimentação do jogador, combate contra inimigos, sistemas de HP e controle da câmara.

O projeto procura explorar aspetos fundamentais do design de jogos, como a integração de animações, física, e inteligência artificial. Além disso, a implementação de uma interface amigável para o jogador, como barras de HP e mudanças visuais durante eventos críticos (por exemplo, a morte do jogador), foi essencial para melhorar a experiência de jogo.

A metodologia utilizada envolveu o uso de scripts em C#, realizados no Visual Studio, para implementar a lógica de jogo, interação entre objetos e controle do estado do jogo. Cada componente foi testado de forma iterativa, garantindo que as mecânicas funcionassem de maneira consistente e fluida.

Este relatório também destaca as possibilidades futuras de expansão do projeto.

Descrição do Jogo

O projeto consiste num jogo de ação em terceira pessoa com elementos de tiro e sobrevivência num cenário de ficção científica a bordo de uma nave espacial. O jogador controla um personagem que deve explorar este ambiente hostil enquanto enfrenta monstros espaciais, inimigos perigosos que também possuem a capacidade de disparar projéteis. A experiência é desafiadora, exigindo habilidade no movimento e precisão no disparo para sobreviver.

O jogador pode se mover livremente pelo cenário utilizando um sistema de controle intuitivo que combina o teclado e o mouse. A câmara oferece uma perspetiva dinâmica em terceira pessoa, permite que o jogador tenha uma visão clara do ambiente e dos inimigos. Em caso de morte do jogador, a visão alterna para uma perspetiva superior, o que proporciona uma maior imersão.

O objetivo principal do jogo é sobreviver aos ataques dos monstros enquanto explora a nave. O jogador deve derrotar os inimigos disparando balas e evitar danos ao se movimentar estrategicamente pelo cenário. O personagem também tem a possibilidade de recuperar HP ao eliminar inimigos.

Além disso, o ambiente da nave espacial apresenta corredores estreitos que aumentam a tensão e exigem que o jogador esteja sempre atento aos perigos.

Implementação das Funcionalidades

A implementação do projeto foi realizada em Unity, e foi utilizada a linguagem C# nos 5 ***scripts*** criados.

* **Movimento do Jogador**

O sistema de movimento do jogador foi implementado utilizando o componente **CharacterController**. O jogador pode se mover nas direções definidas pelo teclado (WASD) com base na orientação da câmara. A gravidade foi simulada para garantir que o personagem se mantenha no solo. Para melhorar a experiência do jogador, foram adicionadas animações de andar sincronizadas com o movimento.

* **Controlo da Câmara**

A câmara principal segue o jogador dinamicamente, oferecendo uma perspetiva clara durante o jogo. Foi implementado um sistema para alternar para uma câmara superior ***OverheadCamera*** quando o jogador morre, o que altera a sensação visual e dá maior destaque à animação de morte.

* **Sistema de Tiro**

O jogador pode disparar balas pressionando o botão esquerdo do mouse. A implementação utiliza ***prefabs*** para os projéteis, que são instanciados no ponto de ***spawn*** da arma e lançados com uma força definida. Um sistema de som foi adicionado para simular o som do disparo.

* **Inimigos e Inteligência Artificial**

Os inimigos foram implementados utilizando ***NavMesh*** para patrulharem áreas definidas e atacar o jogador ao entrar nos seus campos de visão. Os inimigos disparam balas e perseguem o jogador caso este esteja próximo, mas não se aproximam demasiado, mantendo uma distância mínima.

* **Sistema de HP**

Tanto o jogador quanto os inimigos possuem barras de HP, o HP do jogador diminui ao ser atingido por balas dos inimigos e pode ser recuperada ao eliminá-los. A barra de HP do jogador é exibida no canto superior esquerdo do ecrã, enquanto a dos inimigos aparece por cima das suas cabeças.

* **Deteção de Colisões**

As interações no jogo, como tiros a acertar os inimigos ou o jogador, foram implementadas utilizando sistemas de colisão (***OnTriggerEnter***). A ***tag*** de cada objeto (***Player***, ***Enemy***, ***Bullet***) foi utilizada para diferenciar os comportamentos durante as colisões.

* **Animações**

As animações do jogador, como caminhar e morrer, foram integradas com transições definidas por parâmetros no ***Animator***. No caso de morte, foi adicionado um ***trigger*** para reproduzir a animação correspondente antes de mudar para o menu principal.

* **Sistema de Game Over**

Quando o HP do jogador chega a zero, a animação de morte é acionada, os controles do jogador são desativados, e a câmara alterna para a câmara superior. Após a animação, uma transição automática leva o jogador de volta ao menu principal.

* **Design do Ambiente**

O ambiente do jogo foi projetado para criar uma sensação de tensão. O uso de corredores estreitos e ***point lights*** contribui para o clima de perigo constante, enquanto os monstros espaciais tornam o avanço pela nave uma tarefa desafiadora.

Testes e Validação

* **Testes de Jogabilidade**
* **Movimentação do Jogador:**  
  O sistema de movimentação foi testado extensivamente para garantir que o jogador pudesse se deslocar de forma suave e responsiva em todas as direções. Foram verificadas colisões com obstáculos, fluidez do movimento em relação ao cenário, e interações com plataformas.
* **Mecânica de Tiro:**  
  Foram realizados testes para assegurar que o disparo de balas ocorresse no momento certo ao pressionar o botão de disparo, que a direção dos projéteis estivesse correta em relação à arma e que o alcance e destruição das balas funcionassem conforme esperado.
* **Câmara:**   
  A transição da câmara para a perspetiva superior após a morte do jogador foi testada para garantir que fosse acionada no momento correto e que os controles da câmara fossem desativados.
* **Testes de Combate**
* **Interação com Inimigos:**  
  O comportamento dos inimigos foi testado para verificar se eles atacam o jogador corretamente e se as balas disparadas por eles colidem adequadamente com o jogador.
* **Sistema de HP:**  
  O sistema de redução e regeneração de HP foi testado ao longo de várias sessões para garantir que:
  + O dano causado por balas inimigas fosse registado corretamente.
  + O jogador pudesse recuperar HP ao eliminar inimigos.
  + O jogo fosse encerrado quando os HP do jogador chegassem a zero.
* **Animação de Morte:**  
  Verificou-se se a animação de morte era acionada de maneira fluida e sincronizada com a transição de câmara e o retorno ao menu principal ocorre-se após a animação estar concluída.
* **Testes de Interface (UI)**
* **Barra de HP**  
  Testes foram realizados para assegurar que a barra do jogador reagisse dinamicamente às alterações nos HP, diminuindo ou aumentando proporcionalmente.
* **Menu Principal:**  
  Foi testada a navegação pelas ***scenes*** e o carregamento correto do jogo a partir do menu principal.
* **Testes de Desempenho**
* **Otimização de Objetos:**  
  Testou-se o uso eficiente de recursos, como a destruição de balas e inimigos quando fora de cena, para evitar sobrecarga na memória.
* **Frame Rate:**  
  Garantiu-se que o jogo “rodasse” sem quedas significativas de desempenho em ambientes com múltiplos inimigos e projéteis.

Os resultados destes testes ajudaram a identificar ajustes necessários, como equilíbrio de dificuldade e melhorias no sistema de controle.

**Conclusão**

O projeto desenvolvido é um jogo de ação em terceira pessoa ambientado em um cenário de ficção científica, com mecânicas de tiro, movimentação estratégica e sobrevivência. Desde a conceção inicial até a implementação e testes finais, o jogo foi projetado para oferecer uma experiência desafiadora e imersiva, explorando conceitos de ***gameplay*** dinâmico e integração de diferentes sistemas, como controle de câmara, animações e combate com inimigos.

Durante o processo, foram enfrentados desafios técnicos, como a sincronização de animações (o uso de **any** **state**), a otimização da ***NavMesh*** e o sistema de colisões. No entanto, através de testes rigorosos e contínuos, foi possível alcançar um resultado funcional, com destaque para a fluidez dos controles e a interação com os inimigos.

O projeto cumpriu os objetivos propostos, demonstrando a viabilidade técnica e criatividade da ideia inicial. Há, contudo, espaço para futuras expansões, como o desenvolvimento de mais cenários, inimigos com comportamentos mais complexos, maior diversidade de armas, outros sobreviventes, entre outros.

Referências

Foram utilizados diversos assets da unity store e do Mixamo:

* **Space SFX - 102218**
* Standard Galactic Alphabet Font
* UI Samples
* Diverse Space Skybox
* MicroBar - Animated Health Bar
* Stylized Guns Pack
* FPS Assault Rifle - Sci-Fi weapon
* MAC Sci-Fi Sniper Rifle
* 3D Scifi Kit Starter Kit
* Robot Hero : PBR HP Polyart
* <https://assetstore.unity.com>
* <https://www.mixamo.com/#/?page=1&type=Character>