Faculdade CESGRANRIO

Faculdade CESGRANRIO – FACESG – Projeto Integrador VI

Projeto de Jogos Digitais – 1º Semestre de 2023

GAME DESIGN DOCUMENT:

“SPACE DODGER”

Equipe de Desenvolver, Grupo A:

- Gabriel Portela Mello ID 191029
- Gabriel Rodrigues Santos ID 191011

Orientação

- Prof. Rogério
- Prof. André Cotelli

RIO DE JANEIRO 2023

SUMÁRIO

Sumário

LISTA DE FIGURAS

1 INTRODUÇÃO

Space Dodger é um jogo de ação e aventura espacial em que o jogador assume o papel de um piloto de nave espacial, competindo por recursos valiosos no espaço. Neste jogo, o jogador deve navegar através de um campo de asteroides, desviando de obstáculos aleatórios e evitando tiros de naves inimigas. Com controles de toque simples, o jogador deve utilizar suas habilidades para sobreviver o maior tempo possível e alcançar a pontuação mais alta. Space Dodger será lançado para dispositivos móveis com o sistema Android.

1.1 RESUMO DA HISTÓRIA

A história de Space Dodger se passa em um futuro distante, onde a humanidade descobriu que o espaço está cheio de recursos valiosos. Várias corporações competem por esses recursos e o jogador assume o papel de um piloto de nave espacial que trabalha para uma dessas corporações. O objetivo do jogador é navegar através de um campo de asteroides e evitar colidir com eles, enquanto enfrenta naves inimigas que tentam atacá-lo. À medida que o jogador avança no jogo, novos obstáculos e inimigos são introduzidos. O jogo combina uma jogabilidade intensa e viciante com uma história envolvente em um cenário de ficção científica futurista.

1.2 GAMEPLAY OVERVIEW

Space Dodger é um jogo de ação e aventura espacial onde o jogador assume o papel de um piloto de nave espacial em um ambiente caótico e desafiador. O objetivo do jogo é navegar através de um campo de asteroides e naves inimigas enquanto desvia e destrói os obstáculos no caminho, somando pontos para ficar no topo do ranking de jogadores.

O jogador controla a nave utilizando toques na tela do dispositivo móvel para fazer a nave subir, descer, atirar e desviar dos obstáculos e dos tiros inimigos. Os asteroides aparecem aleatoriamente no campo de jogo, aumentando a dificuldade do jogo à medida que o jogador progride. Além disso, as naves inimigas aparecem e atiram contra o jogador, o que adiciona uma nova dimensão à jogabilidade e aumenta ainda mais o desafio.

O jogador deve estar atento e utilizar suas habilidades para desviar e destruir os obstáculos e naves inimigas, somando pontos.

Com gráficos impressionantes e controles simples, o jogo é acessível a jogadores de todas as idades e níveis de habilidade. Space Dodger é uma homenagem aos jogos clássicos de nave espacial, como Space Invaders e Galaga, com um toque moderno e emocionante.

1.3 GÊNERO, SEMELHANÇAS E DIFERENÇAS

O gênero do jogo é ação/arcade, com elementos de ficção científica. O jogo tem semelhanças com jogos clássicos de nave espacial, como Galaga (**Error! Not a valid bookmark self-reference.**) e Space Invaders (**Figura 2**), com um toque moderno e emocionante. O objetivo principal é sobreviver o maior tempo possível enquanto se desvia de asteroides e naves inimigas que atiram no jogador.

A principal diferença é ser um jogo 3D. O jogo também será disponibilizado exclusivamente para dispositivos móveis com o sistema Android.

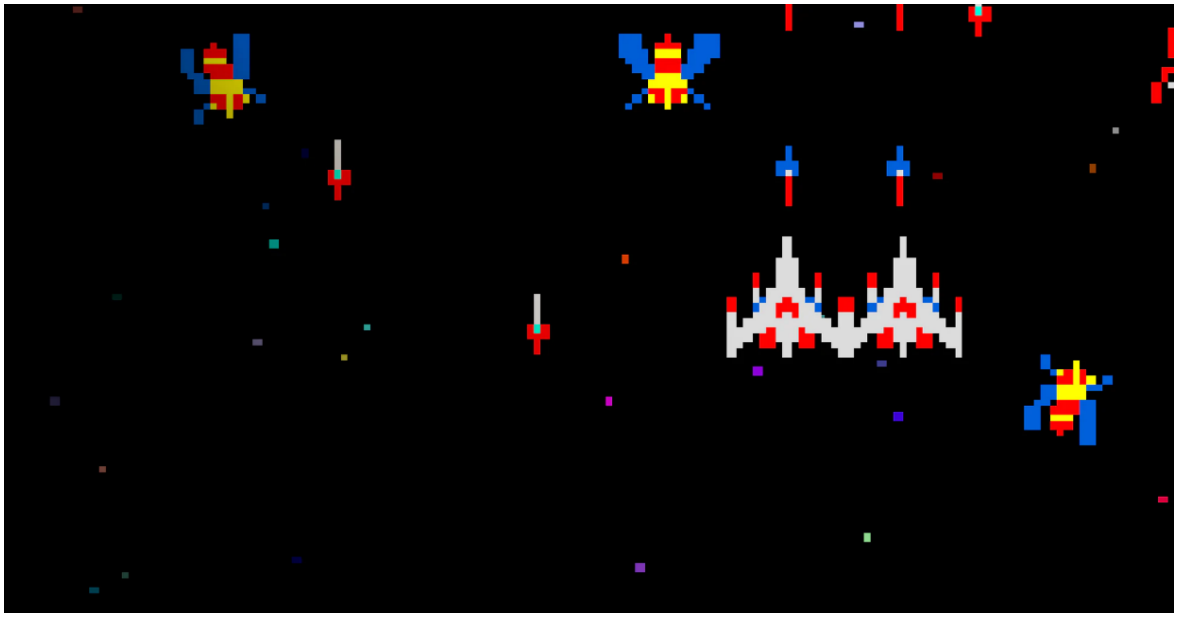

Figura 1 - Tela do jogo galaga referência (https://store.playstation.com/pt-br/concept/10006627)

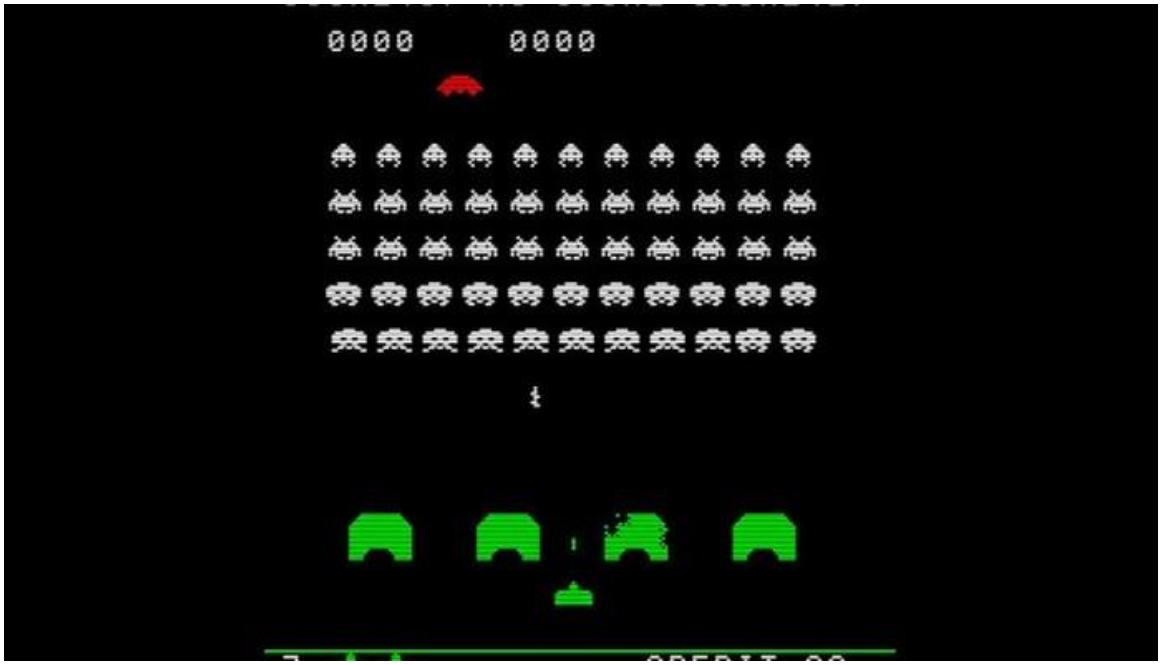

Figura 2 - Tela do jogo space invaders referência (https://www.techtudo.com.br/noticias/2016/03/space-invaders-veja-lista-com-curiosidades-e-polemicas-do-jogo.ghtml)

1.4 PÚBLICO ALVO

O público-alvo principal de Space Dodger são jogadores de jogos de ação e arcade, que apreciam jogos que apresentam uma jogabilidade intensa e desafiadora. O jogo é acessível a jogadores de todas as idades e níveis de habilidade, mas é mais adequado para jogadores que buscam um desafio.

O jogo também pode atrair jogadores que gostam de jogos de ficção científica futurista e que apreciam gráficos impressionantes e controles simples. A plataforma escolhida para o jogo, dispositivos móveis com o sistema Android, também pode atrair jogadores que preferem jogar em seus smartphones e tablets.

1.5 ATRATIVOS DO JOGO

O principal atrativo do jogo é o desafio de desviar de asteroides, naves inimigas e tiros inimigos, que aparecem aleatoriamente no campo de jogo, tornando cada partida única e imprevisível.

1.6 FLUXO DO JOGO

Ao iniciar o jogo, o jogador é levado para a tela inicial, onde pode escolher jogar ou acessar as opções de configuração.

O jogador deve controlar sua nave espacial com toques na tela para desviar dos asteroides e evitar os disparos dos inimigos.

O objetivo é sobreviver o maior tempo possível e somar pontos para ficar no topo do ranking de jogadores.

Se o jogador colidir com um asteroide, uma nave inimiga ou for atingido por um tiro inimigo, o jogo acaba e o jogador terá de tentar novamente.

O jogo apresenta gráficos impressionantes, controles simples, tornando a experiência de jogo envolvente e emocionante.

2 INTERFACE E INTERAÇÃO

Neste tópico, descrevemos como o jogador interage com o jogo através dos dispositivos de entrada e saída.

2.1 ENTRADAS

O jogo utilizará a tela do dispositivo móvel como entrada dos controles.

2.1.1 TOQUE NA TELA

Movimentos da Nave: O toque na tela move a nave para cima ao longo do eixo Y, permitindo que o jogador desvie dos obstáculos e inimigos.

Tiros: O toque no botão atirar, dispara tiros contra os inimigos que estão a frente da nave do jogador.

2.2 SAÍDAS

A saída do jogo é feita através de vídeo (tela), oferecendo uma experiência imersiva e emocionante para o jogador.

2.2.1 MENUS

O jogo contará com menus para diferentes fins. O menu principal permitirá que o jogador inicie um novo jogo, acesse as configurações e saia do jogo. Já o menu de pausa permitirá que o jogador retome o jogo ou saia para o menu principal.

Além disso, o jogo contará com menus de fim de jogo, onde o jogador poderá ver sua pontuação final e as opções de jogar novamente ou sair do jogo. Todos os menus serão projetados de forma clara e intuitiva, para garantir uma experiência de usuário agradável e fácil de usar.

3 MECÂNICA DO JOGO

Neste capítulo, abordaremos as principais mecânicas do jogo.

3.1 MECÂNICA BÁSICA

A mecânica básica do jogo é o controle da nave espacial através de toques na tela do dispositivo móvel. O jogador deve desviar dos obstáculos que aparecem no campo de jogo.Além disso, o jogador também deve enfrentar naves inimigas que atiram contra ele.

3.2 COMBATE

O jogador enfrentará naves inimigas que atiram contra ele. Para se defender, a nave do jogador também estará equipada com armas. Além disso, o jogador pode coletar bônus de pontos ao destruir as naves inimigas e asteroides.

3.3 PROGRESSÃO

O jogador precisará enfrentar obstáculos e inimigos. O objetivo final é acumular pontuação e ficar no topo do ranking de jogadores.

3.4 VIDA

O jogo será encerrado quando a nave colidir com um obstáculo, uma nave inimiga ou for atingida por um tiro inimigo.

3.5 CONDIÇÕES DE VITÓRIA

As condições de vitória em Space Dodger são baseadas na pontuação do jogador. O objetivo é somar o maior número de pontos possível ao desviar dos asteroides e destruir as naves inimigas. O jogo não tem um fim específico, permitindo que o jogador continue jogando e melhorando sua pontuação. No entanto, à medida que o jogador atinge novas pontuações, pode ser desbloqueado conteúdo adicional no jogo, como novas naves ou cenários. O ranking global de jogadores também é exibido, permitindo que os jogadores comparem suas pontuações com outros jogadores em todo o mundo.

4 CONCEPT ARTS E SPRITES

A seguir, uma apresentação dos personagens, naves, cenários e inimigos do jogo.

4.1 PERSONAGENS

4.1.1 NAVE ESPACIAL DO JOGADOR

Nome: Fury-9

Tipo: Nave espacial de combate

História do passado: A Fury-9 é uma das naves mais avançadas da frota espacial. Projetada para missões de alto risco, ela é pilotada por um grupo de elite de pilotos espaciais. A nave já enfrentou inúmeros desafios e se destacou em batalhas espaciais cruciais.

Habilidades:

- Armamento a laser: A Fury-9 possui um poderoso armamento a laser capaz de disparar projéteis de alta velocidade e precisão.
- Propulsores de alta potência: A nave está equipada com propulsores de alta potência, permitindo manobras verticais de subida e descida com agilidade.

Ilustração visual (Figura 3):

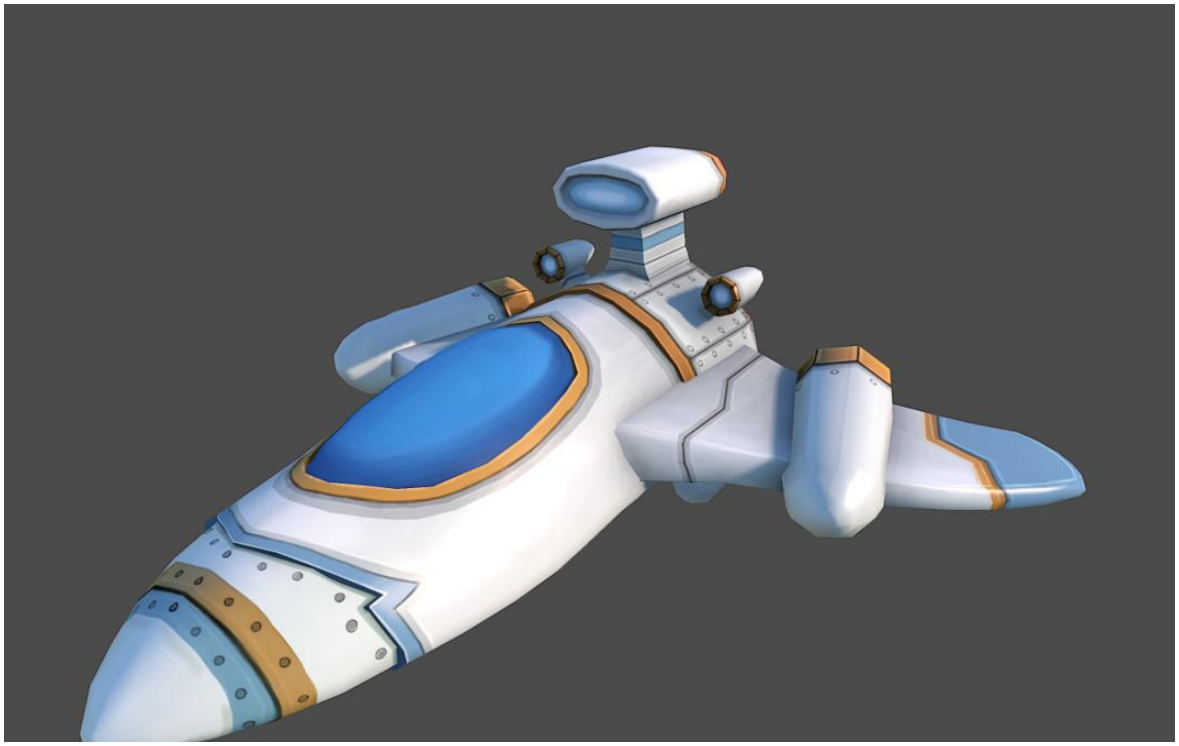

Figura 3 - Nave referência (https://assetstore.unity.com/packages/3d/galactic-heroes-cartoon-spaceship-70188)

- Ações: A nave do jogador pode executar as seguintes ações durante o jogo:
- Voar nas direções vertical de subida e descida.
- Disparar com suas armas a laser.
- Desviar de obstáculos.

4.1.2 INIMIGOS

O asteroide (**Figura 4**) e a nave espacial inimiga (**Error! Reference source not found.**).

Figura 4 - INIMIGO ASTEROIDE (https://assetstore.unity.com/packages/3d/props/breakable-asteroids-167825)

Asteroides: Rochas espaciais flutuantes representando um perigo constante para a nave de Alex Nova. Eles surgem de forma aleatória, exigindo habilidades de esquiva para evitar colisões.

Naves Inimigas: Caças espaciais hostis que atiram contra a nave de Alex Nova. Essas naves são rápidas, ágeis e têm diferentes armamentos. Cada tipo de nave inimiga possui comportamentos de ataque únicos.

Figura 5 - Inimigo principal do jogo - nave espacial referência (https://assetstore.unity.com/packages/3d/galactic-heroes-cartoon-spaceship-70188)

4.1.3 CENÁRIO

A imagem (**Figura 6**) de referência para o cenário:

Figura 6 - REFERÊNCIA DE CENÁRIO (https://br.freepik.com/vetores-gratis/fundo-de-galaxia-realista_14960493.htm#query=galaxia&position=3&from_view=keyword&track=sph)

O jogo se passa no espaço sideral, onde a nave espacial do jogador deve desviar dos asteroides e evitar os ataques inimigos. O cenário é um espaço aberto e infinito, com asteroides aparecendo aleatoriamente no campo de jogo. O jogador deve demonstrar habilidade e agilidade para navegar com sucesso nesse ambiente desafiador.

Nesse cenário, a jogabilidade se concentra em desviar dos asteroides e enfrentar as naves inimigas que atiram contra o jogador. A nave espacial deve se movimentar para cima ou para baixo para evitar colisões e também usar suas armas a laser para destruir as naves inimigas. A ação se desenrola em um espaço aberto, com um fundo estrelado e visualmente cativante.

4.1.4 CÂMERA:

A câmera do jogo (Error! Reference source not found.) é posicionada ao lado da nave espacial, proporcionando ao jogador uma visão lateral. Isso permite ao jogador ter uma perspectiva clara do ambiente ao lado da nave e facilita a visualização dos obstáculos e inimigos que se aproximam.

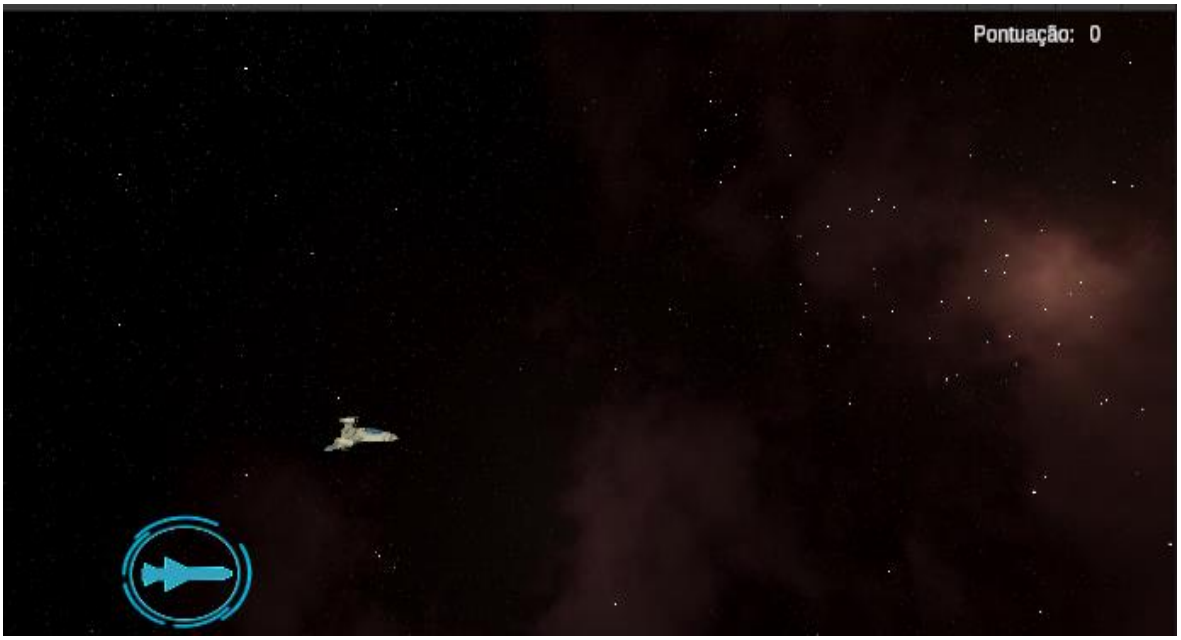

Figura 7- Câmera

A câmera acompanha o movimento da nave, permitindo que o jogador tenha controle total sobre a posição da nave em relação à tela. Isso ajuda na navegação precisa e na tomada de decisões estratégicas durante o jogo.

A visão lateral também oferece uma sensação de velocidade e movimento, aumentando a imersão e a emoção da jogabilidade. O jogador pode apreciar os efeitos visuais, como o rastro de propulsão da nave, à medida que avança pelo campo de asteroides e enfrenta as naves inimigas.

5 INTERFACES

Figura 8 - Menu Inicial

O menu (**Figura 8**) se trata de uma interface interativa que permitem aos jogadores navegar por diferentes opções, como iniciar um novo jogo, acessar o placar, e sair. O menu também contém o título do jogo com uma nave decorativa.

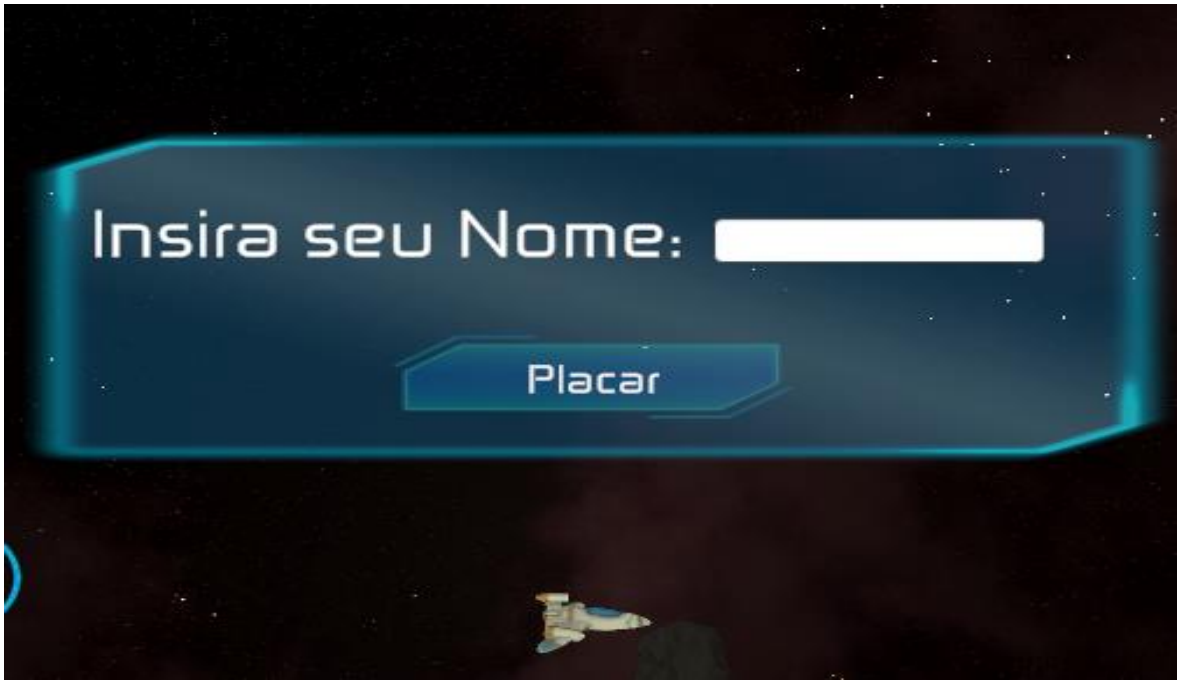

Figura 9 - Inserir nome

Quando o jogo é finalizado, a interface (**Figura 9**) é exibida, permitindo que

o jogador insira seu nome. Essa interface apresenta um campo de entrada de texto onde o jogador pode digitar seu nome e um botão "Placar" para enviar o nome para o ranking. Essa interação permite ao jogador registrar sua pontuação e participar do ranking, compartilhando seu desempenho com outros jogadores.

Figura 10 - Placar de recordes

A interface de placar (**Figura 10**) exibe as cinco maiores pontuações e um botão de sair. Os jogadores podem comparar seu desempenho com outros jogadores e buscar pontuações mais altas. É um recurso essencial para acompanhar o progresso e desafiar-se no jogo.

6 DETALHAMENTO TÉCNICO

6.1 ENGINE

A engine escolhida para a criação do jogo é a Unity3D, que é uma plataforma voltada para o desenvolvimento de jogos 3D e possui suporte a dispositivos móveis, incluindo o sistema operacional Android. A Unity3D oferece diversas funcionalidades necessárias para a criação do jogo, facilitando o processo de produção. Além disso, a plataforma possui uma linha de aprendizado mais fácil e uma interface amigável, o que torna o processo de desenvolvimento mais eficiente e acessível.

6.2 HARDWARE

Dispositivos móveis com sistema operacional Android 4.4 ou superior, com tela touch screen.

6.3 SOFTWARE

Serão utilizadas as seguintes ferramentas:

- Unity3D: plataforma de desenvolvimento de jogos 3D, utilizada para a criação do jogo Space Dodger;
- Visual Studio Community: ambiente de desenvolvimento integrado (IDE) utilizado para a codificação em C# e a criação de scripts para a integração com a Unity3D.

6.4 PRINCIPAIS REFERÊNCIAS

Neste capítulo serão apresentadas as principais referências que inspiraram o desenvolvimento do jogo Space Dodger.

O clássico arcade "Galaga" serviu como referência para o estilo de jogo e a mecânica de esquiva de obstáculos. Referência: **(https://store.playstation.com/pt-br/concept/10006627)**

A série de filmes "Star Wars" inspirou a estética do jogo, incluindo a escolha de cores e a ambientação futurística. Referência: (https://pt.wikipedia.org/wiki/Star_Wars)

O jogo "Flappy Bird" inspirou a mecânica de toque na tela para controlar a nave. Referência: (https://pt.wikipedia.org/wiki/Flappy_Bird)

6.5 ASSETS

6.5.1 SCI-FI GUI SKIN

Esse pacote do Unity é uma coleção de recursos para criação de interfaces gráficas de usuário (GUI) com temática de ficção científica. Ele inclui uma skin de GUI estilizada, um joystick móvel e diversos ícones. Além disso, esse recurso pode ser utilizado para criar skins para qualquer outro sistema de interface do usuário, como uGUI e nGUI.

O pacote inclui os seguintes elementos:

- Arquivo .guiskin para uso com a Unity UI
- Botões
- Barras de progresso
- Skin para o joystick
- Sliders (controles deslizantes) e thumbs (indicadores de posição)
- Molduras
- Setas
- Elementos extras

Esses recursos podem ser utilizados para criar interfaces gráficas personalizadas em projetos de jogos ou aplicativos com uma estética de ficção científica. Eles fornecem elementos pré-fabricados que podem ser facilmente integrados e customizados de acordo com as necessidades do desenvolvedor.

6.5.2 LOOTLOCKER SDK

O LootLocker também é utilizado para registrar os pontos do jogo e recuperar o ranking online. Os desenvolvedores podem integrar a funcionalidade do LootLocker em seus jogos para armazenar e acompanhar os pontos dos jogadores, bem como exibir rankings online. Isso permite que os jogadores comparem suas pontuações com outros jogadores, promovendo a competição e o engajamento. O LootLocker facilita a implementação desses recursos, ajudando os desenvolvedores a fornecerem uma experiência envolvente e competitiva para os jogadores.

6.5.3 VOLUMETRIC LINES

O pacote "Volumetric Lines" foi utilizado no projeto para criar os disparos

utilizados tanto pela nave do jogador quanto pela nave inimiga. Com esse recurso, foi possível adicionar um efeito visual impactante aos disparos, dando-lhes uma aparência volumétrica. Isso contribuiu para uma experiência de jogo mais imersiva e dinâmica, tornando os disparos mais visíveis e realistas. Através do uso das linhas volumétricas, os tiros ganharam uma sensação tridimensional, aprimorando a qualidade visual do jogo e proporcionando uma experiência visualmente impressionante aos jogadores.

6.5.4 BREAKABLE ASTEROIDS

O pacote "Breakable Asteroids" oferece uma coleção de asteroides únicos que podem ser destruídos e se fragmentar em pedaços menores em resposta a ações no jogo. Esses asteroides são ideais para jogos de tiro espacial e outros jogos ambientados no espaço, adicionando elementos visuais interessantes e desafiadores aos projetos dos desenvolvedores.

6.5.5 NEBULA SKYBOXES

O pacote "Nebula Skyboxes" é um conjunto de 4 skyboxes de nebulosas. Cada skybox é uma textura cubemap de alta resolução com dimensões de 8192 x 6144 pixels (2048 x 2048 por face). Esses skyboxes foram criados para serem compatíveis com os pipelines de renderização do Unity, incluindo o Built-In Renderer, URP e HDRP. Isso significa que os desenvolvedores podem usar esses skyboxes em uma variedade de projetos, independentemente do pipeline de renderização escolhido. As texturas são fornecidas no formato EXR (HDR), que é um formato de imagem de alta faixa dinâmica.

Isso permite que os skyboxes tenham uma gama mais ampla de valores de cor e detalhes, resultando em visuais mais ricos e realistas. Com o pacote "Nebula Skyboxes", os desenvolvedores têm acesso a uma seleção de skyboxes impressionantes, inspirados em nebulosas. As nebulosas são nuvens cósmicas de gás e poeira encontradas no espaço sideral, e essas texturas capturam sua aparência. Esses skyboxes podem ser usados para criar cenários em jogos ou aplicativos, especialmente aqueles com temáticas espaciais. Eles fornecem uma experiência visual imersiva.

6.5.6 GALACTIC HEROES CARTOON SPACESHIP

O pacote "Galactic Heroes Cartoon Spaceship" inclui um modelo de nave

espacial juntamente com seis skins diferentes para personalização visual. O modelo de nave espacial é projetado em um estilo cartunesco e vem com materiais de difusão e especulares.

Ao utilizar esse pacote, os desenvolvedores têm acesso a um modelo detalhado de uma nave espacial adequada para jogos ou aplicativos com temática de ficção científica. A nave espacial pode ser personalizada com uma variedade de skins, permitindo que os desenvolvedores escolham entre diferentes estilos e aparências para atender às necessidades de seus projetos.

6.6 SCRIPTS

6.6.1 ASTEROID.CS

Este script controla o comportamento do asteroide no jogo. Ele define a velocidade do asteroide, acompanha quantos tiros foram recebidos pelo asteroide, adiciona pontos à pontuação geral do jogador ao destruir o asteroide e lida com colisões entre o asteroide e os tiros do jogador ou outros objetos com a tag "Finish".

6.6.2 LOOTLOCKER_PLACAR.CS

Este script controla a exibição de um placar usando o LootLocker SDK. Ele define um ID para identificar o placar desejado e um número máximo de jogadores a serem exibidos no placar. O array **Jogadores** contém os objetos TextMeshProUGUI onde os jogadores e suas pontuações serão exibidos.

No método **Start()**, o script inicia uma sessão de convidado usando o LootLocker SDK para autenticação.

O método **MostrarPlacar()** é responsável por obter a lista de pontuações do placar usando o LootLocker SDK. Ele atualiza os objetos TextMeshProUGUI com os jogadores e suas pontuações obtidas da resposta.

6.6.3 LOOTLOCKER _SISTEMA.cs

Este script é responsável por enviar a pontuação do jogador para o placar usando o LootLocker SDK. Ele possui um campo de entrada de texto para o nome do jogador (**Nome**) e um objeto TextMeshProUGUI para exibir a pontuação do jogador (**Placar**). O ID do placar também é definido para identificar o placar correto.

No método **Start()**, o script inicia uma sessão de convidado usando o LootLocker SDK para autenticação.

O método **EnviarPlacar()** é responsável por enviar a pontuação do jogador

para o placar usando o LootLocker SDK. Ele chama o método **SubmitScore()** do SDK, fornecendo o nome do jogador, a pontuação (convertida de **Placar.text** para **int**) e o ID do placar. A resposta do envio é tratada no callback, onde uma mensagem de sucesso ou erro é exibida.

6.6.4 MENUSCRIPT.CS

Este script controla o menu do jogo e possui dois métodos associados a botões.

O método **botaoJogar()** é chamado quando o jogador pressiona o botão "Jogar" no menu. Ele carrega a cena chamada "Jogo" usando o método **LoadScene()** da classe **SceneManager**. Isso permite que o jogador inicie o jogo ao pressionar o botão "Jogar".

O método **Sair()** é chamado quando o jogador pressiona o botão "Sair" no menu. Ele chama o método **Quit()** da classe **Application**, o que encerra o aplicativo ou jogo.

6.6.5 MOSTRARPLACAR.CS

Este script automatiza o clique de um botão com um atraso específico.

No método **Start()**, o script invoca o método **PressButton()** após o tempo definido em **delay**. Isso permite que o botão seja pressionado automaticamente após o atraso especificado.

No método **Update()**, o script chama explicitamente o evento **onClick** do botão usando **button.onClick.Invoke()**. Isso permite que o botão seja pressionado novamente a cada quadro de atualização, se necessário.

6.6.6 NAVEINIMIGA.CS

Este script controla o comportamento de uma nave inimiga no jogo. Ele define a velocidade da nave inimiga, acompanha quantos tiros foram recebidos pela nave inimiga, instância tiros da nave inimiga em intervalos regulares e lida com colisões entre a nave inimiga e os tiros do jogador ou outros objetos com a tag "Finish".

6.6.7 PLAYERCONTROLLER.CS

Este script controla o jogador durante o jogo. Ele possui lógica para o jogador pular, colidir com objetos, atirar, exibir pontos, reiniciar o jogo e exibir o placar.

No método **Update()**, o script verifica se houve um toque na tela para fazer

a nave pular. Ele adiciona uma força ao Rigidbody para fazer a nave subir. Também há código comentado para inclinar a nave para a esquerda ou direita com base no movimento do mouse ou teclado.

Os métodos **OnTriggerEnter()** e **OnCollisionEnter()** são chamados quando o jogador colide com objetos com as tags "rock", "NaveInimiga" e "tiroInimigo". Eles atualizam a pontuação, pausam o jogo e exibem o menu de derrota, conforme necessário.

O método **Atirar()** é responsável por instanciar um tiro do jogador no jogo, desde que o tempo de espera (**tiroteste**) tenha passado. Ele também define a posição de instanciamento do tiro e reproduz um áudio.

Os métodos **Reiniciar()** e **Placar()** são usados para reiniciar o jogo ou mostrar o placar, respectivamente. Eles alteram a cena usando o **SceneManager** e ajustam o tempo e a visibilidade do menu de derrota.

6.6.8 SOUNDTRACK.CS

Este script controla a reprodução da trilha sonora do jogo. Ele utiliza o componente **AudioSource** para reproduzir o áudio.

No método **Start()**, o script obtém o componente **AudioSource** do objeto ao qual o script está anexado (**GetComponent<AudioSource>()**). Em seguida, ele chama o método **Play()** para iniciar a reprodução da trilha sonora.

6.6.9 SPAWN.CS

Este script controla a lógica de geração de inimigos no jogo. Ele utiliza um temporizador (**tempo**) para determinar quando e onde os inimigos devem ser instanciados.

No método **Update()**, o script atualiza o temporizador incrementando o valor com base no tempo decorrido (**Time.deltaTime**). Quando o temporizador ultrapassa um determinado limite (4 segundos no exemplo), o script gera uma posição aleatória no eixo Y dentro de um limite definido. Em seguida, ele instancia um inimigo com base na pontuação atual do jogador.

Se a pontuação for menor que 2, ele instancia o primeiro inimigo do array (**Inimigo[0]**). Caso contrário, ele gera um índice aleatório entre 0 e 1 para selecionar um inimigo do array e, em seguida, realiza a instanciação.

6.6.10 TIRO.CS

Este script controla o comportamento do tiro no jogo. Ele move o tiro para

frente na direção oposta ao eixo Z usando a velocidade definida (**speed**). Quando o tiro colide com um objeto com a tag "TiroFim", ele é destruído.

6.6.11 TIROINIMIGO.CS

Este script controla o comportamento do tiro inimigo no jogo. Ele move o tiro inimigo para frente ao longo do eixo Z usando a velocidade definida (**speed**). Quando o tiro inimigo colide com um objeto com a tag "Finish", ele é destruído.

6.7 DESAFIOS TÉCNICOS ENFRENTADOS

- **Desafios Visuais:** Implementação de fundos dinâmicos e imersivos, efeitos 3D dos objetos e skins nos objetos para imersão no tema espacial.
- **Lógica de Pontuação:** Desenvolvimento de scripts para calcular e rastrear a pontuação do jogador com base em ações específicas no jogo, como atingir alvos, destruir inimigos, somar pontos ou superar objetivos.
- **Interface do Usuário:** Criação de interfaces de usuário intuitivas, incluindo a exibição e atualização de elementos da interface, como pontuações e ranking geral.
- **Colisões e Física:** Implementar sistemas de colisão precisos para interações entre objetos e personagens, lidar com detecção de colisão, resolução e resposta adequada, além de aplicar conceitos de física realista para movimentos e interações no jogo.