

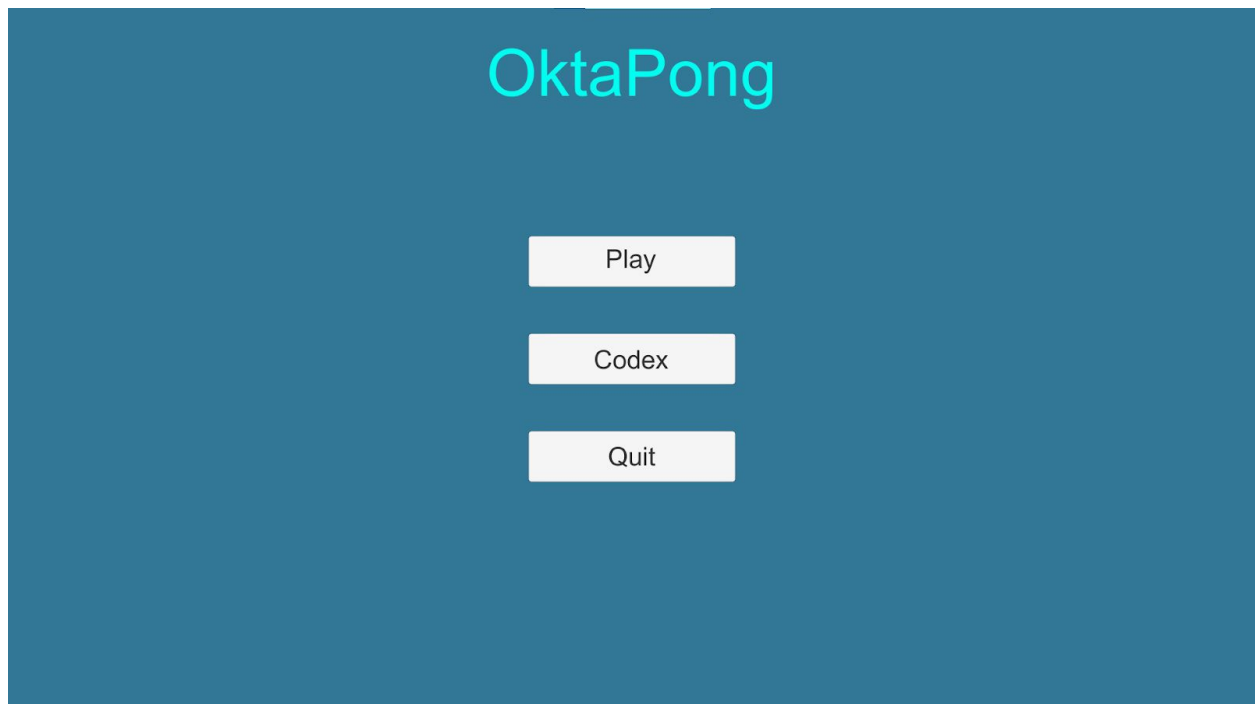
OktaPong

Visão Geral

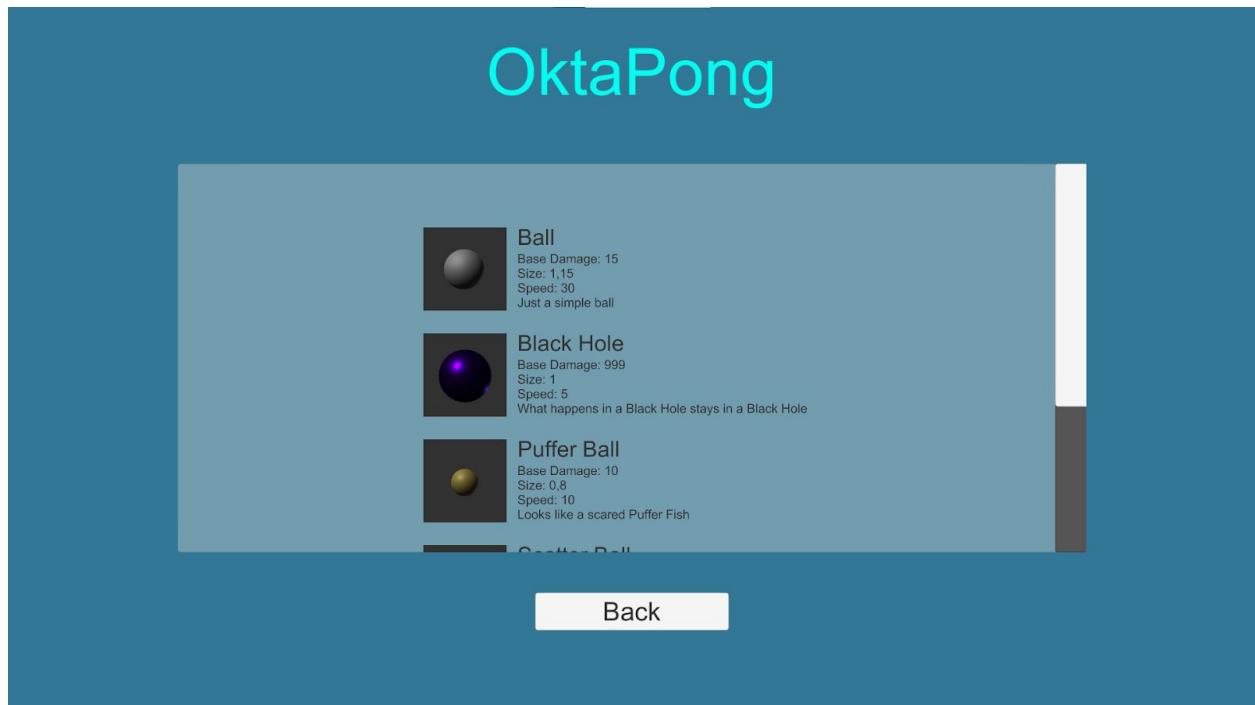
O jogo consiste em uma partida de turnos 1v1 em que o objetivo é eliminar o adversário.

Telas

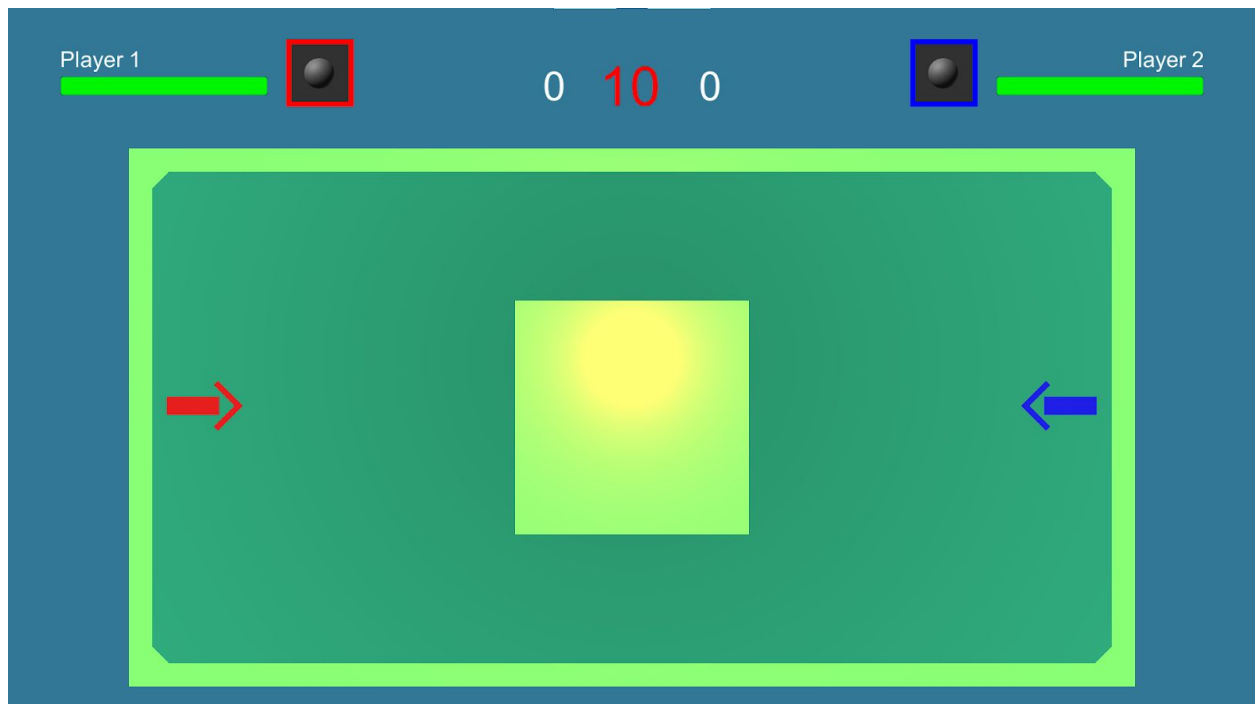
Menu Inicial:



Codex:



Jogo:



Gameplay

- O jogador que começará a partida é escolhido aleatoriamente;
- Em seu turno, o jogador poderá se mover, escolher diferentes tipos de projéteis e atirar;
- O jogador da vez tem 10 segundos para atirar, caso contrário, ele perde a sua vez;
- O dano causado por um tiro depende de seus atributos base; [1]
- Quando um jogador é morto, soma-se um ponto para o adversário e começa uma nova rodada no turno do jogador abatido;
- O jogador que alcançar 3 pontos vence a partida.

Controles

↑: Move para cima.

↓: Move para baixo.

←: Vira para a esquerda.

→: Vira para a direita.

Q : Altera o tiro selecionado

Space: Atira

Projeto

Cenas e Elementos Principais

MenuScene:

- Canvas
 - MainMenu
 - CodexMenu
 - .../ContainerInfo
 - CodexInfoBuilder.cs
- _UIManager_
 - MenuSelector.cs

MainScene:

- Canvas
 - MatchUI
 - P1_HealthBar
 - P2_HealthBar

- P1_SelectedProj
 - P2_SelectedProj
 - P1_Score
 - Scoreboard.cs
 - P2_Score
 - Scoreboard.cs
 - Timer
 - Timer.cs
- ResultsUI
 - Panel
 - ResultsPanel.cs
- _GameManager_
 - MatchHandler.cs
 - ProjectileHandler.cs
- Arena
 - Border
 - ArenaBuilder.cs

Prefabs:

- Player
 - PlayerController.cs
 - PlayerHealth.cs
- Projectile
 - Projectile.cs
- Projectile Variants
 - <Name>Proj.cs : Projectile.cs

Scripts

Informativo:

CodexInfoBuilder.cs: Busca por todos os tipos de projéteis disponíveis e cria um painel informativo para cada um.

Scoreboard.cs: Atualiza a pontuação do jogador.

Timer.cs: Atualiza o tempo restante do turno.

ResultsPanel.cs: Exibe o jogador vitorioso junto ao placar final.

PlayerController.cs: Responsável pela leitura de inputs do jogador.

PlayerHealth.cs: Atualiza a vida atual do jogador.

Core:

MenuSelector.cs: Responsável pela transição de cenas.

ArenaBuilder.cs: Escolhe aleatoriamente um preset de obstáculos e instancia a cada rodada.

MatchHandler.cs: Responsável pelo controle de turnos, instanciamento de projéteis e atualização dos painéis informativos dos jogadores.

ProjectileHandler.cs: Responsável pela seleção e exibição dos projéteis selecionados por cada jogador.

Projectile.cs: Responsável pela física do projétil e cálculo do dano.

ProjectileVariant.cs: Herda as funcionalidades de Projectile.cs e contém implementações adicionais conforme o necessário.

Detalhes

Formula do Dano:

Damage = (((baseDamage * speed / 30) * size) * Random.Range(0.75f, 1.25f));