

Jackson Celeghin Rodrigues Junior
Rafael Junqueira Martarelli
Fernando Yoiti Fujihara Nobuyasu

Brain Killer

Sobre:

O sistema consiste num jogo multi-tasking. O jogo se inicia com apenas um minijogo e com o passar do tempo irá sendo adicionado novos minijogos. Em cada nova tela, um novo minijogo é introduzido e o jogador deve jogar todos ao mesmo tempo, tal que este jogo não funcione apenas como jogo, mas também como exercício para o cérebro.

Quando o jogo é iniciado, o score é incrementado continuamente com o passar do tempo e a dificuldade vai aumentando, com o aumento da velocidade do jogo.

Algumas ideias de jogos são equilibrar uma bolinha sobre uma barra utilizando o acelerômetro do celular, um mini cobrinha, algo no estilo do Genius (seguir sequência de cores) e clicar sobre objetos de uma determinada cor num amontoado de objetos.

Procuramos por algo parecido na Play Store e só encontramos dois jogos parecidos (Multi-task Brain Teaser e MulTI Taks) o que nos encorajou a seguir com o projeto.

Requisitos:

1. Funcionais

- a. RF01 - O jogo deve fornecer no máximo 4 atividades simultâneas
- b. RF02 - A escolha das atividades deve aleatória
- c. RF03 - As atividades devem conter desafios de raciocínio ou coordenação motora
- d. RF04 - O jogo deve conter opções de som e dificuldade
- e. RF05 - O gráfico deve ser 2D
- f. RF06 - Se algum dos jogos simultâneos der game over, todos os outros jogos ativos irão parar
- g. RF07 - O jogo deverá ter um score
- h. RF08 - Deve haver um tempo mínimo para que o um novo jogo simultâneo inicie

2. Requisitos não funcionais

- a. RNF01 - O jogo ser a desenvolvido em Godot
- b. RNF02 - O jogo será desenvolvido para Android e Windows
- c. RNF03 - O score será hospedado no Google Play e em um arquivo

Diagramas:

Projeto Arquitetural:

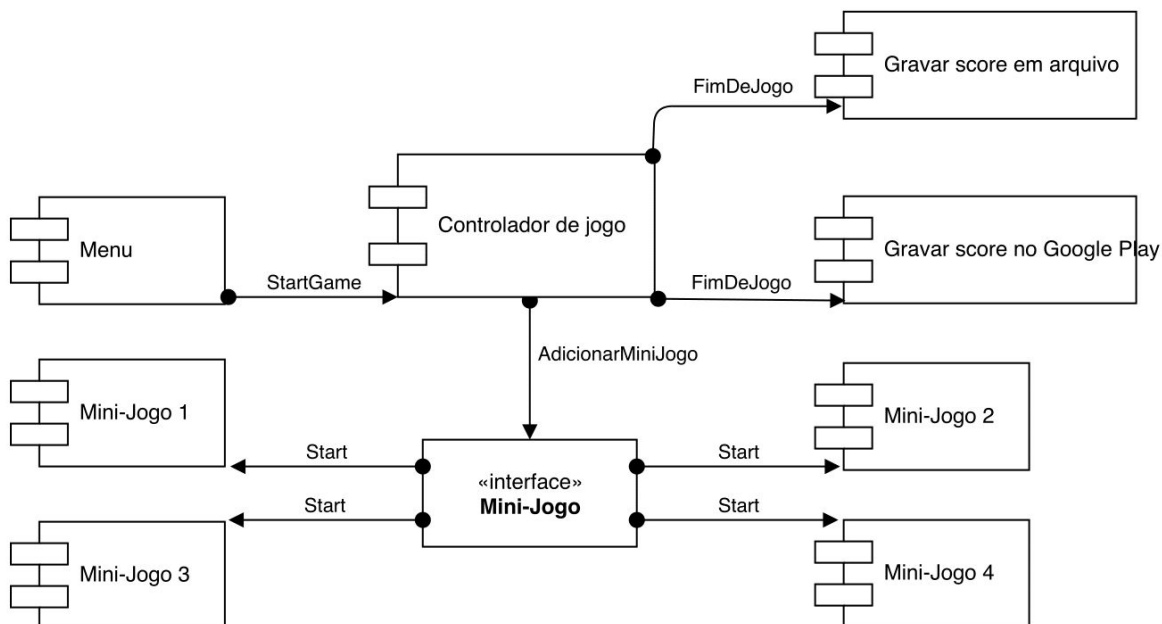
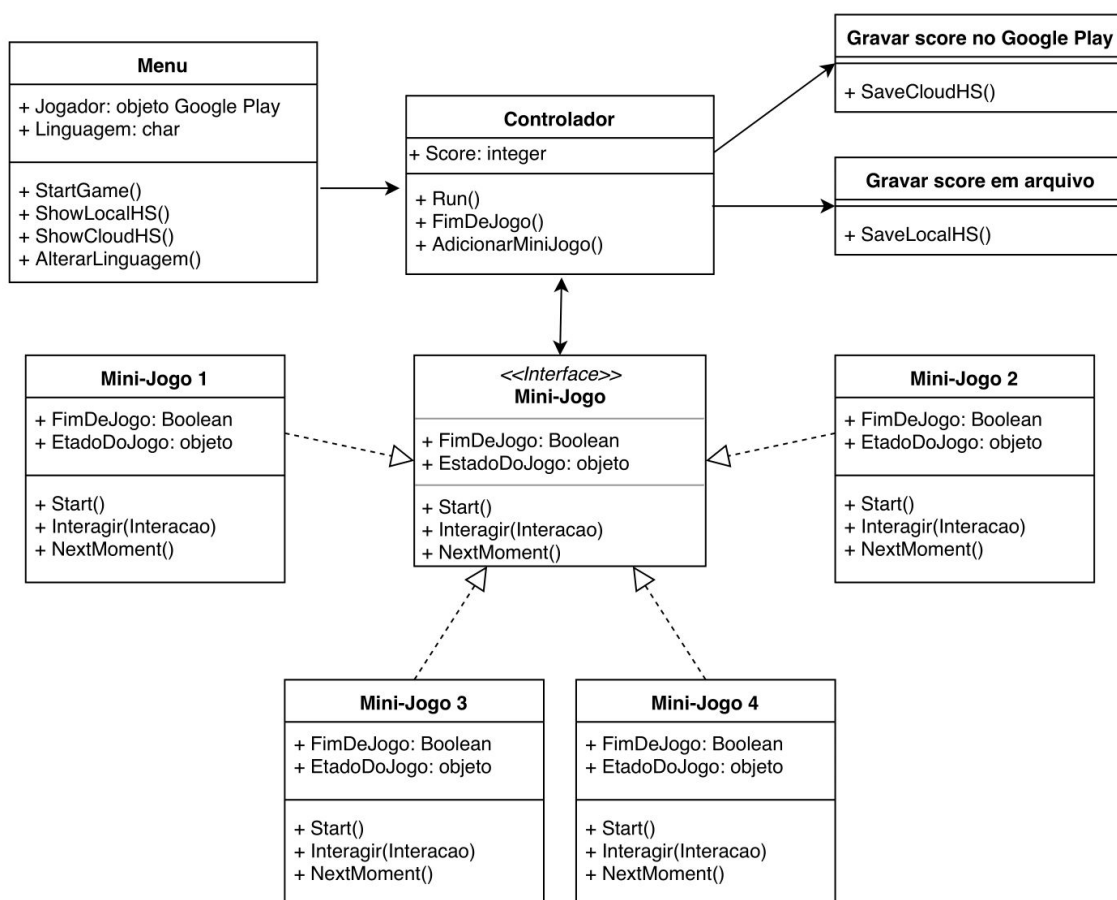
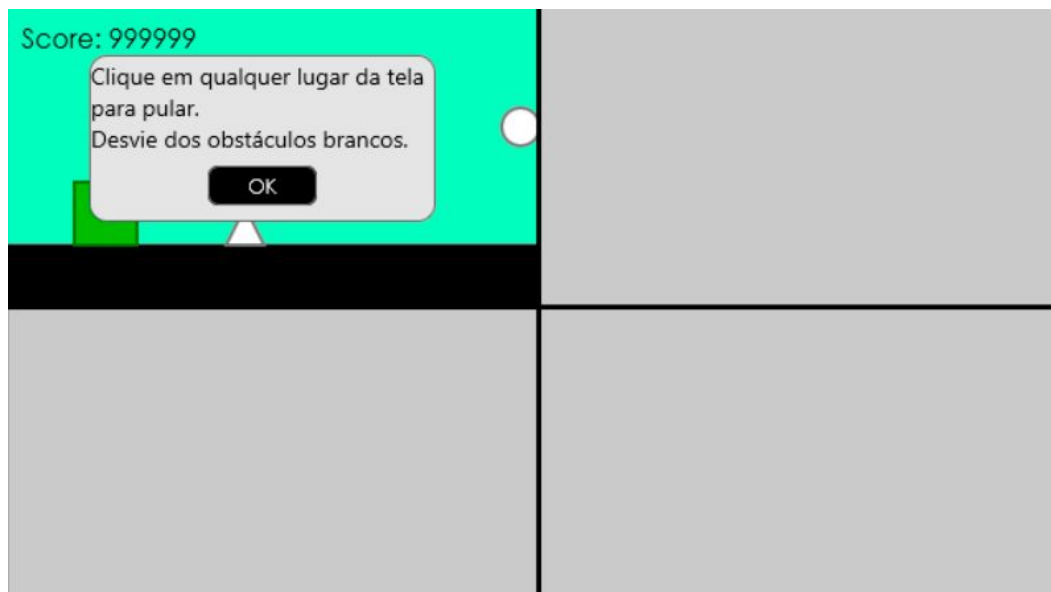
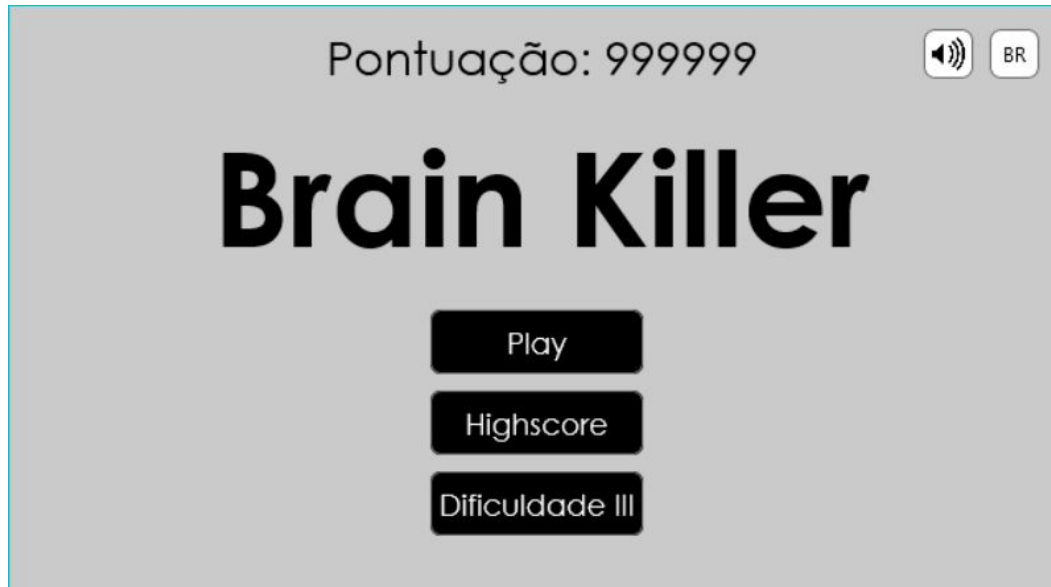


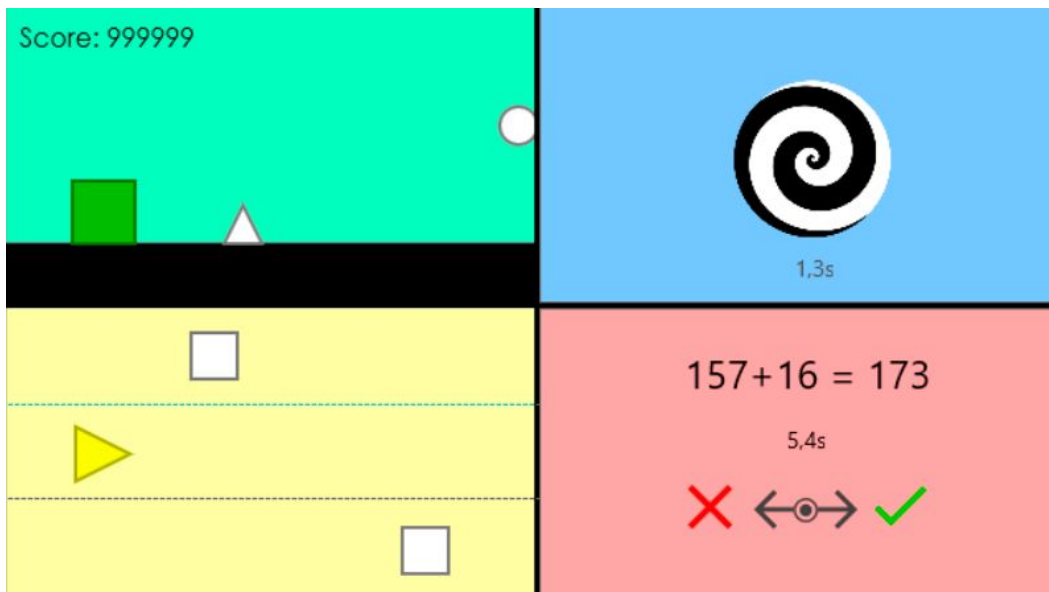
Diagrama de classes:



Como o jogo não tem banco de dados, não existe o M-ER.

Mockup:





Estimativas:

Esforço de desenvolvimento:

Acreditamos que, retirando o código gerado pelo framework, será desenvolvido 1k de linhas, através do COCOMO chegamos no valor de 3,6 person/month:

YOUR BASIC COCOMO RESULTS!!								
MODE	"A" variable	"B" variable	"C" variable	"D" variable	KLOC	EFFORT, (in person/months)	DURATION, (in months)	STAFFING, (recommended)
undefined	3.6	1.2	2.5	0.32	1	3.6	3.7666633722353353	0.9557530483175543

Esforço de manutenção:

Utilizando o ACT. Supondo que haja uma manutenção de 10% do código, podemos chegar que o custo de manutenção será:

$$E = 0,1 * 2,4 * 1 = 0,24$$

Portanto será necessário 0,24 desenvolvedores/mês para fazer a manutenção desse sistema.

Casos de Teste:

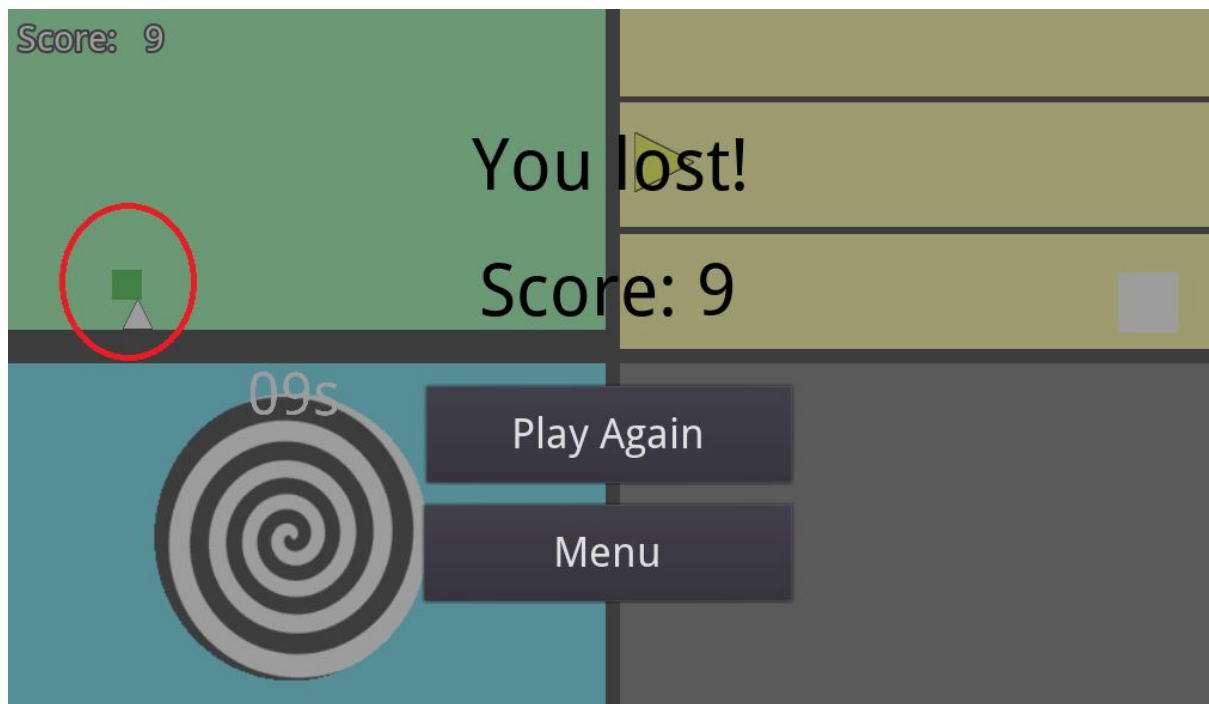
Caso de teste 1:

Colidir com um triângulo no primeiro jogo

Finalidade: testar o game over do primeiro jogo

Resultado esperado: Perder o jogo

Resultado obtido: Perdi o jogo



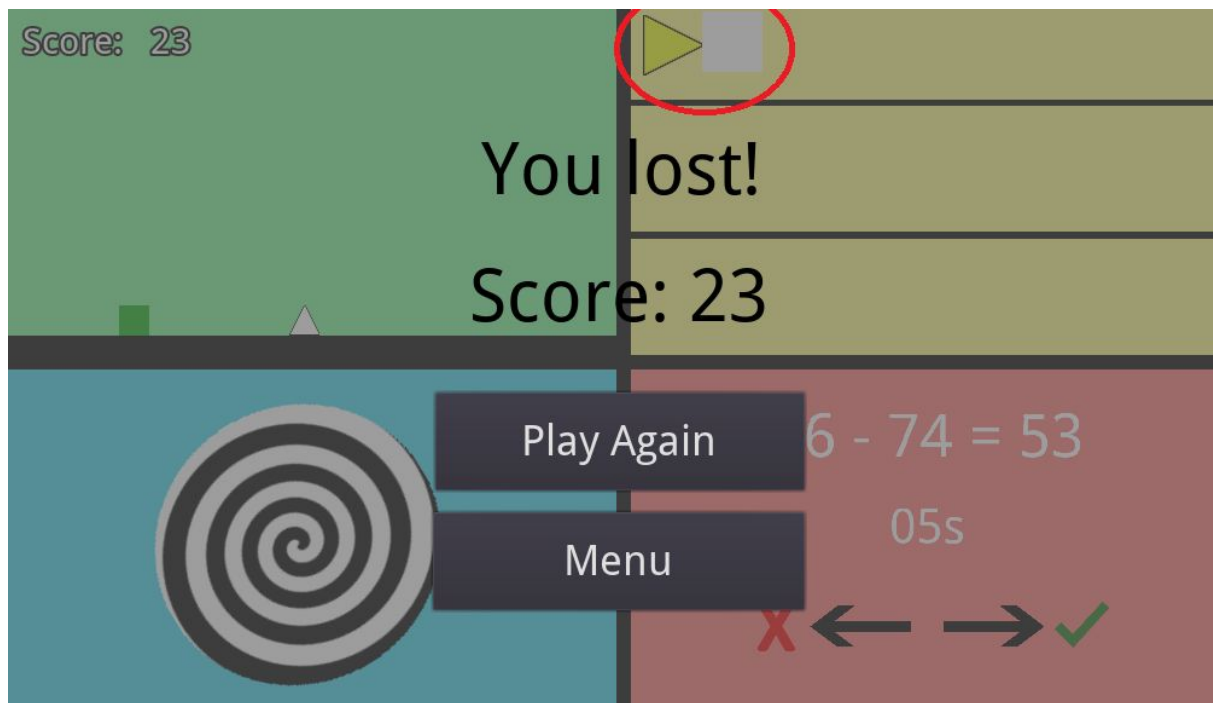
Caso de teste 2:

Colidir com um quadrado no segundo jogo

Finalidade: testar o game over do segundo jogo

Resultado esperado: Perder o jogo

Resultado obtido: Perdi o jogo



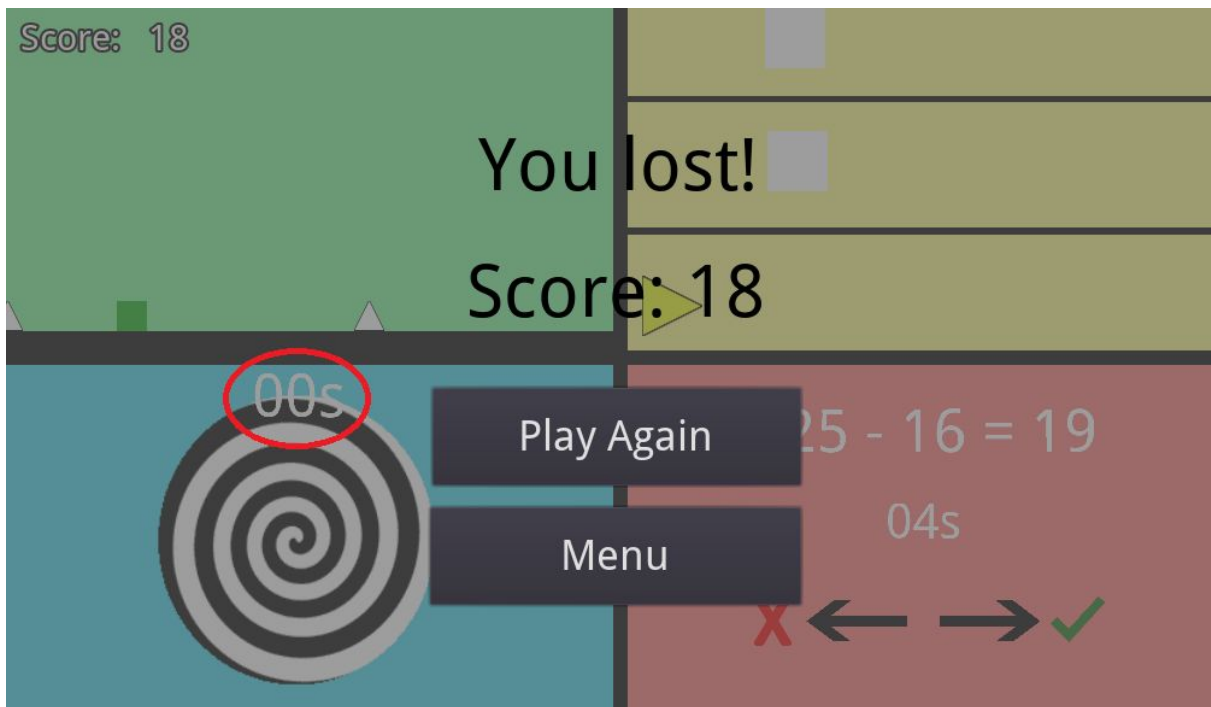
Caso de teste 3:

Não rotacionar o círculo no terceiro jogo

Finalidade: testar o game over do terceiro jogo

Resultado esperado: Perder o jogo

Resultado obtido: Perdi o jogo



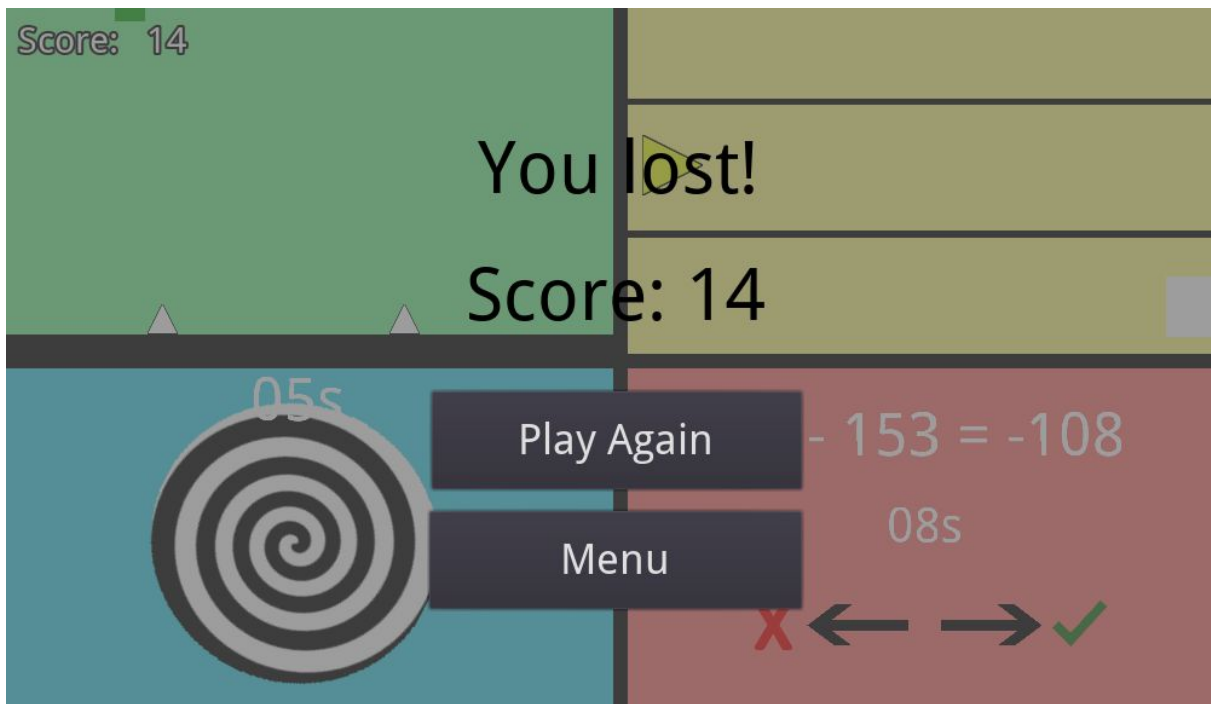
Caso de teste 4:

Errar a conta no quarto jogo

Finalidade: testar o game over do quarto jogo

Resultado esperado: Perder o jogo

Resultado obtido: Perdi o jogo



Caso de teste 5:

Fechar e abrir o jogo para ver se o score se mantém

Finalidade: testar o sistema de arquivos

Resultado esperado: Manter o score

Resultado obtido: Manteu o score

