

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS E INFORMÁTICA
UNIDADE EDUCACIONAL PRAÇA DA LIBERDADE
Bacharelado em Engenharia de Software**

Nome dos integrantes do grupo:

Gabriel Burdignon
João Pedro Silva

Nome do sistema:

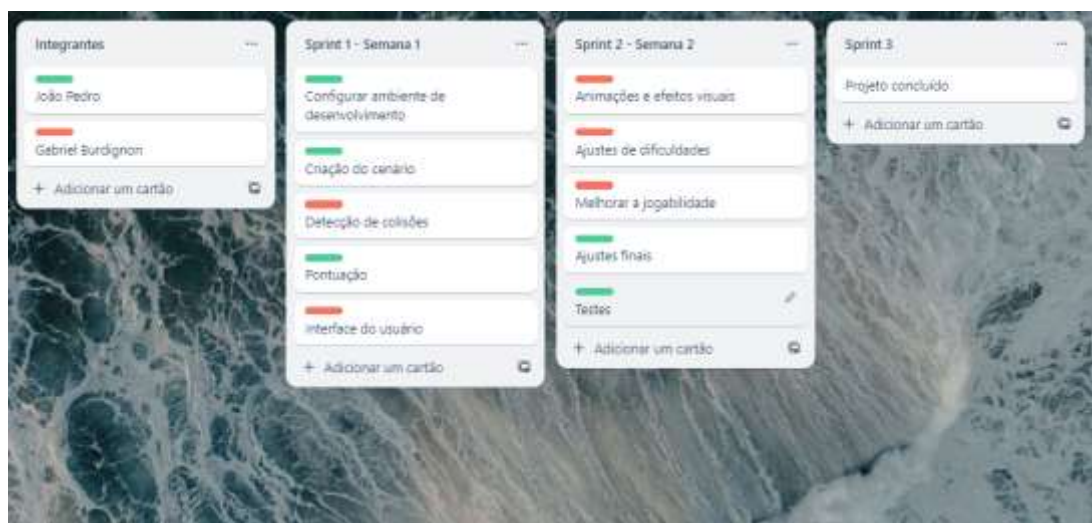
Flappy Bird

Apresentação:

Flappy Bird é um de enorme sucesso que envolve controlar um pássaro e desviar de tubos para obter a maior pontuação possível. Os jogadores tocam na tela para fazer o pássaro bater as asas e evitar colisões. Com gráficos minimalistas e uma trilha sonora agradável, o jogo atrai jogadores de todas as idades. A dificuldade implacável e a jogabilidade viciante são características marcantes do Flappy Bird. Desafie seus amigos, teste suas habilidades e estabeleça recordes neste jogo clássico.

Backlog to produto:

O backlog consiste em três sprints para o desenvolvimento do jogo Flappy Bird. A Sprint 1 envolve a configuração do ambiente de desenvolvimento, criação do cenário e elementos básicos, detecção de colisões e implementação do sistema de pontuação. Na Sprint 2, são adicionadas animações, ajustes de dificuldade, melhorias na jogabilidade e os testes que precisam ser feitos. Na Sprint 3 é o projeto concluído.



Fonte: Elaborado pelo grupo

Lista de assinaturas das funções e parâmetros

As funções e parâmetros utilizados no programa foram:

1. **void fim_jogo(int pontuacao)**

Descrição:

Esta função é responsável por exibir a tela de fim de jogo, mostrando a mensagem "Game Over" e a pontuação do jogador.

Parâmetros:

- pontuacao (int): A pontuação do jogador.

Retorno:

A função não possui retorno.

2. **int colisao(Passaro passaro, Obstaculo obstaculo)**

Descrição:

Esta função verifica se ocorreu uma colisão entre o pássaro e o obstáculo.

Parâmetros:

- passaro (Passaro): O objeto do tipo Passaro.

- obstaculo (Obstaculo): O objeto do tipo Obstaculo.

Retorno:

int: Retorna 1 se ocorreu colisão entre o pássaro e o obstáculo, e 0 caso contrário.

TESTES

Casos de teste do software:

Os casos de teste englobam todo o código, uma vez que as funções devem receber parâmetros que podem estar dentro de outras funções ou do código principal.

Entradas	Estrutura Válida	Resultado Esperado	Estrutura Inválida	Resultado Esperado
Passaro passaro; passaro.x = 100; passaro.y = 300; passaro.velocidade = 0; Obstaculo obstaculo; obstaculo.x = 500; obstaculo.y = 250;	Passaro passaro; passaro.x = 100; passaro.y = 300; passaro.velocidade = 0; Obstaculo obstaculo; obstaculo.x = 500; obstaculo.y = 250;	A função de colisão deve retornar 0, indicando que não houve colisão entre o pássaro e o obstáculo.	Passaro passaro; passaro.x = -50; passaro.y = 400; passaro.velocidade = 0; Obstaculo obstaculo; obstaculo.x = 700; obstaculo.y = -100;	O pássaro pode ficar fora da tela ou o obstáculo pode não ser desenhado corretamente. Além disso, a função de colisão pode não funcionar corretamente.

Relatório de Execução de Testes

função colisão

Número	Varáveis de Entrada	Valores válidos	Resultado Esperado	Valores inválidos	Resultado Esperado
1	passaro.y = 100, obstaculo.y = 200	100, 200	Retornar 0 (ausência de colisão)	N/A	N/A
2	passaro.y = 300, obstaculo.y = 250	300, 250	Retornar 1 (ocorrência de colisão)	N/A	N/A
3	passaro.y = 0, obstaculo.y = 0	0, 0	Retornar 1 (ocorrência de colisão)	N/A	N/A
4	passaro.x = 400, obstaculo.x = 300	400, 300	Retornar 0 (ausência de colisão)	N/A	N/A
5	passaro.x = 100, obstaculo.x = 80	100, 80	Retornar 1 (ocorrência de colisão)	N/A	N/A
6	passaro.x = 200, obstaculo.x = -500	200, -500	Retornar 1 (ocorrência de colisão)	obstaculo.x possui valor inválido	N/A
7	passaro.y = "abc", obstaculo.y = 300	N/A	Gerar erro ou comportamento inesperado	passaro.y possui valor inválido	N/A
8	passaro.y = 300, obstaculo.y = "xyz"	N/A	Gerar erro ou comportamento inesperado	obstaculo.y possui valor inválido	N/A
9	passaro.x = true, obstaculo.x = 400	N/A	Gerar erro ou comportamento inesperado	passaro.x possui valor inválido	N/A
10	passaro.x = 400, obstaculo.x = false	N/A	Gerar erro ou comportamento inesperado	obstaculo.x possui valor inválido	N/A

função fim_jogo

<i>Número</i>	<i>Varáveis de Entrada</i>	<i>Valores válidos</i>	<i>Resultado Esperado</i>	<i>Valores inválidos</i>	<i>Resultado Esperado</i>
1	pontuacao = 0	0	Exibir "Game Over" na tela com pontuação 0	N/A	N/A
2	pontuacao = 100	100	Exibir "Game Over" na tela com pontuação 100	N/A	N/A
3	pontuacao = -10	-10	Exibir "Game Over" na tela com pontuação -10	N/A	N/A
4	pontuacao = 9999	9999	Exibir "Game Over" na tela com pontuação 9999	N/A	N/A