



UNIVERSIDADE AUTÓNOMA DE LISBOA

*DEPARTAMENTO DE ENGENHARIAS E CIÊNCIAS DA COMPUTAÇÃO
LICENCIATURA EM ENGENHARIA INFORMÁTICA*

Autor: Gonçalo Marques

Número: 30012478

Professor: Laercio Cruvinel

Data da Entrega: 15/01/2026

INDICE

- Objetivos e conceito do jogo **3**
- Descrição técnica. 3-5
- Referência às transformações, iluminação, projeções e interatividade implementadas. 5-7
- Capturas de ecrã e explicação dos efeitos visuais. 7-9
- Reflexão sobre o processo de desenvolvimento em de A (quando aplicável)..... 9-10

- Objetivos e conceito do jogo

O “Super Cat Runner” é um jogo de 2D onde o jogador controla um gato que corre num cenário contínuo, desviando-se de obstáculos, apanhando power-ups e enfrentando um Boss no nível final.

O principal objetivo é sobreviver o máximo de tempo possível e completar todos os níveis, demonstrando em simultâneo, os conceitos gráficos pedidos na unidade curricular.

- **Objetivos:**

- Aplicar conceitos de computação gráfica 2D com Pygame.
- Demonstrar transformações (translação, escala, rotação/flip), alguma simulação física simples (salto) e interatividade.

- **Conceito do jogo:**

- “Runner 2D” com um gato como protagonista.
- Níveis progressivos (1–5) com dificuldade crescente.
- Obstáculos (troncos, pássaros).
- Power-up que serve para acertar obstáculos/ inimigos
- Tenta sobreviver e acumular pontos até ao Boss.

- Descrição técnica

Biblioteca e arquitetura geral

- **Biblioteca:** Pygame (renderização 2D, eventos, imagens, fonte, som se tiveres).
- **Arquitetura:**
 - Configurações globais
 - Classes (Gato, Obstáculos, PowerUp, Boss, etc.)
 - Funções de UI (menus, ecrãs de vitória/morte)
 - Função main_game com o ciclo principal
 - Função menu como ponto de entrada

Mecânicas principais

- **Movimento do jogador (Gato):**
 - Salto com o w ou espaço
 - Movimento horizontal com A e D, limitado às margens do ecrã.
- **Obstáculos:**
 - Classe Tronco: gerados com largura/altura aleatória, textura em tiling, movimento horizontal.
 - Classe Bird: inimigos aéreos com altura variável e movimento vertical ciclico (sin com `pygame.time.get_ticks()`).
- **Power-up:**
 - Classe PowerUp: sprite flutuante (efeito ondulatório) que ativa `gato.can_shoot` durante X segundos.
 - Projétil: disparo do jogador com velocidade constante para a direita.
- **Boss:**
 - Estados: normal, smash, dash, shoot_pause, returning, recoil.

Comportamento:

- Segue o jogador verticalmente.
- smash: sobe, alinha horizontalmente e desce rápido em direção ao chão.
- dash: investida rápida horizontal.
- shoot_pause: paragens em que dispara “rajadas” de projéteis (BossBullet).
- HP, barra de vida e hitbox reduzida (`get_collision_rect`).

Estruturas de dados

- **Listas:**
 - obstacles para obstáculos ativos.
 - bullets para projéteis do jogador.
 - boss_bullets para projéteis do Boss.
 - powerups para power-ups ativos.
- **Dicionários:**
 - LEVEL_DISTANCES para a distância (score) de cada nível.
- **Retângulos (pygame.Rect):**
 - Usados para posição e colisão de quase tudo.

- Transformações, iluminação, projeções e interatividade

Transformações

- **Translações:**
 - Movimento do fundo (bg_x), chão, obstáculos e projéteis (alterar rect.x / rect.y).
 - Movimento do Boss e do gato.
- **Escala:**
 - pygame.transform.scale para:
 - Ajustar tamanho do jogador (PLAYER_SIZE_NORMAL e PLAYER_SIZE_SHOOT).
 - Aumentar Boss.png, passaro.png, egg.png, power.png, etc.
 - Gerar diferentes dimensões de Cactus com tiling de textura.
- **Rotação / Flip:**
 - flip_x=True para o Boss e pássaros olharem para a esquerda.

- **Efeito de parallax simples:**

- Fundo a mover mais devagar (bg_speed) que a velocidade dos obstáculos, para simular profundidade.

“Iluminação” e efeitos visuais

Não tens iluminação real 3D, mas tens efeitos de cor e destaques:

- **Criação do chão dinamicamente:**

- Cor do chão inferida a partir do background (create_ground_surface), com ajuste de tons (mais verde, menos vermelho/azul).

- **Efeito de “flash” no Boss:**

- Quando leva dano, copias a imagem e aplicas um preenchimento branco semi-transparente (BLEND_RGBA_ADD).

- **Gradiente no menu principal:**

- draw_menu_background desenha linhas horizontais com um gradiente de cor, criando um fundo mais “trabalhado”.
- Podes mencionar isto como “simulação de iluminação/realce 2D através de manipulação de cores e alpha”.

Projeções

- Explica que o jogo usa **coordenadas 2D em ecrã**, com o sistema de coordenadas do Pygame (origem no canto superior esquerdo).
- Não tens projeção 3D, mas podes referir:
 - “Os objetos são desenhados diretamente em coordenadas de ecrã, correspondendo a uma projeção ortográfica 2D simplificada.”

Interatividade

- **Teclado:**

- W / ESPAÇO para saltar.

- A / D para mover horizontalmente.
- X para disparar (quando power-up está ativo).
- P para pausar.
- ENTER, ESPAÇO, ESC, W, S, A, D nos menus.
- **Rato:**
 - Click nos botões dos menus (menu, level_select_menu, credits, botão de voltar ao menu na pausa).

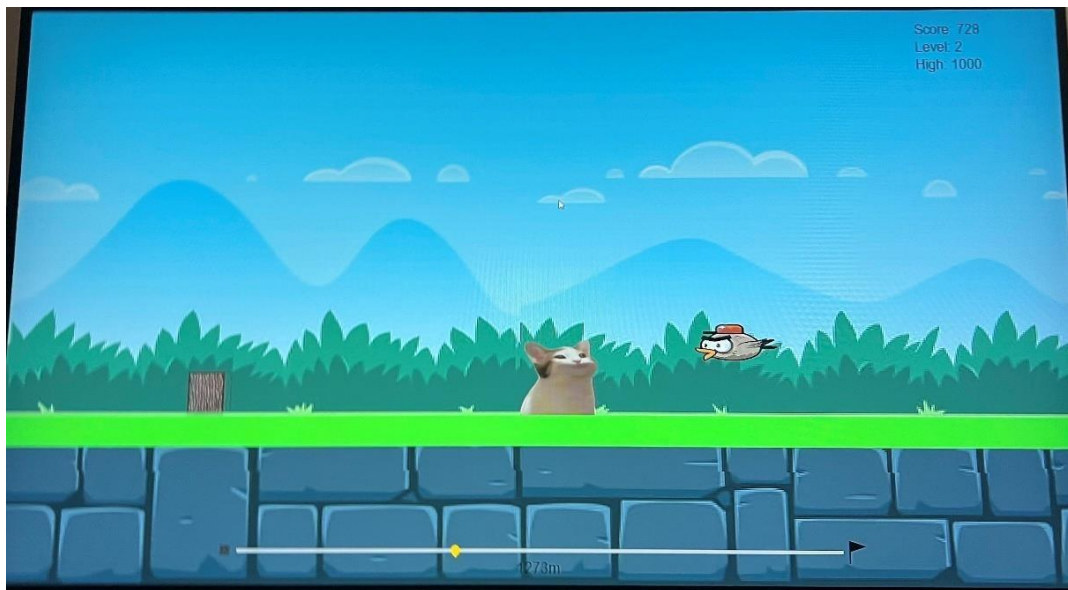
: Capturas de ecrã e explicação dos efeitos visuais

- **Menu principal:**
 - Mostrar o gradiente e os botões com sombra.
 - Explicar draw_button (shadow + hover + seleção via teclado).



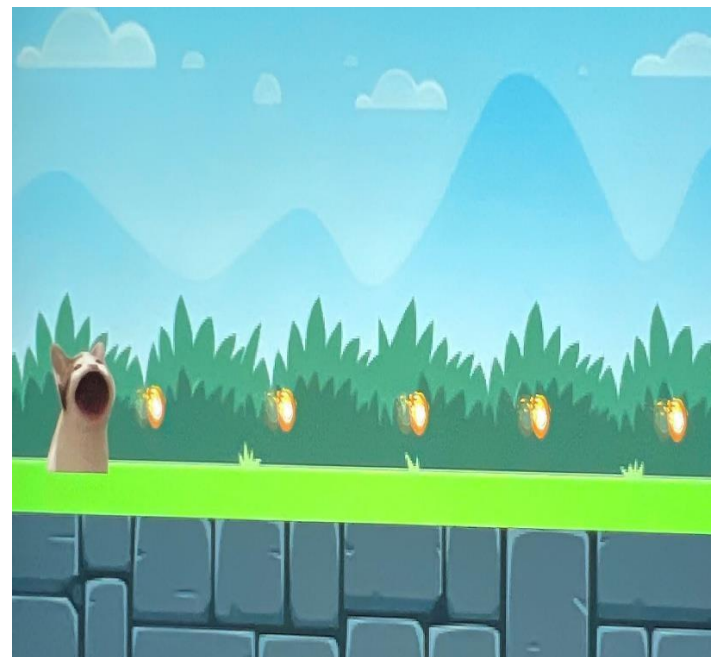
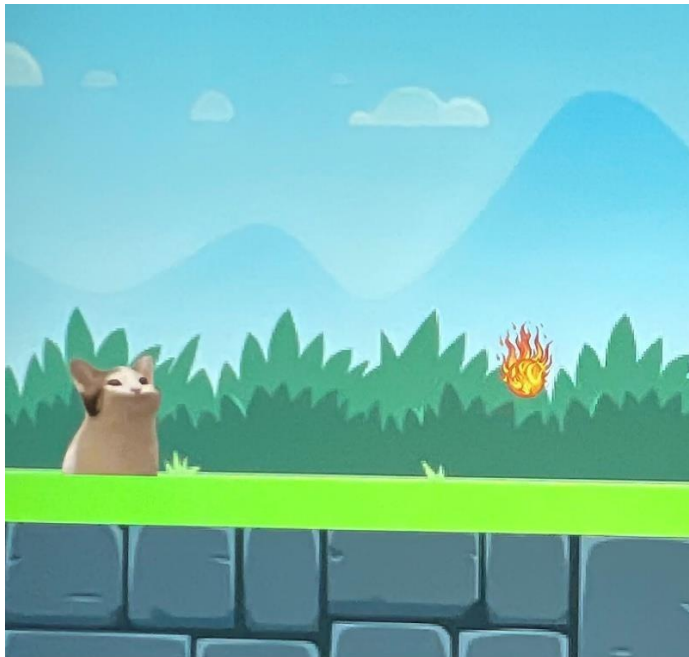
- **Jogo normal:**

- Mostra o gato, chão, troncos/pássaros, HUD de Score/Level/Highscore e barra de progresso do nível.



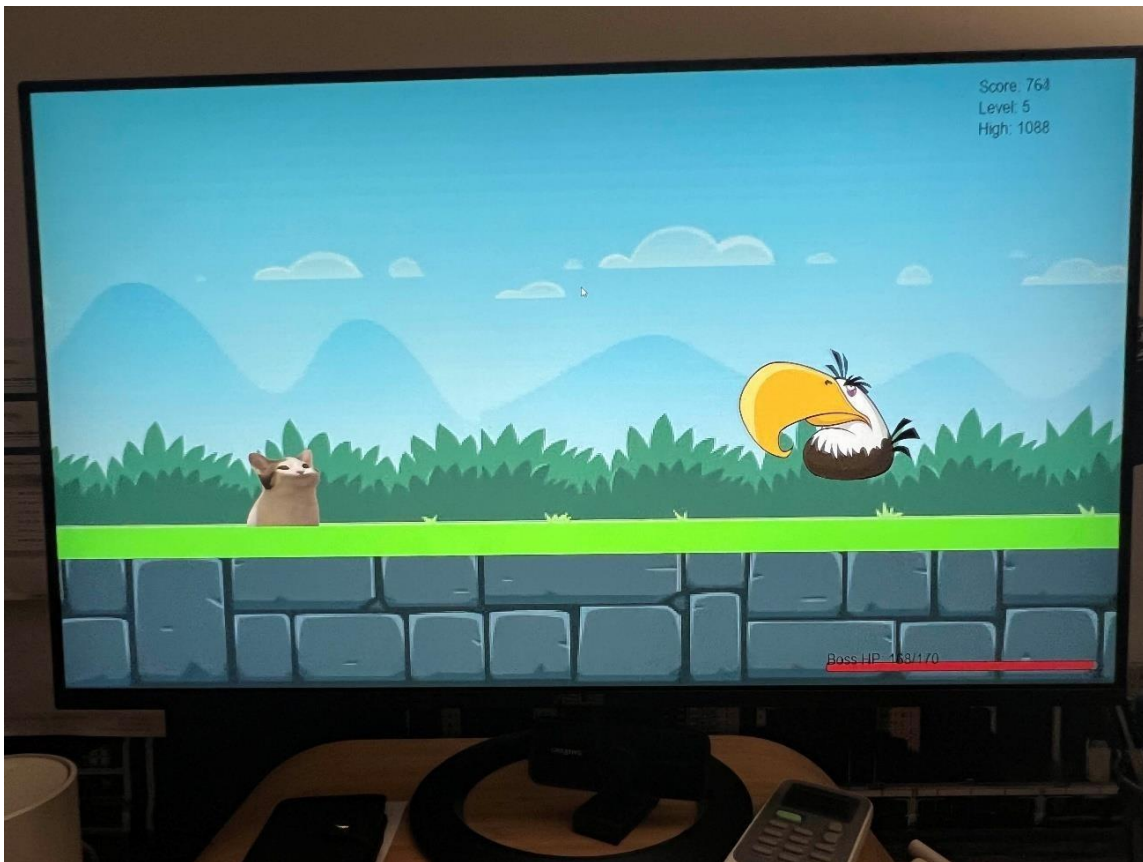
- **Power-up ativo:**

- Mostra o ícone do power-up, projétil fireball.



- **Boss fight:**

- Mostra Boss, barra de HP, projéteis do Boss, projéteis do jogador.



- Reflexão sobre o processo de desenvolvimento e uso de IA

- **Dificuldades técnicas:**

- Detecção de posição do chão a partir da imagem de background.
- Ajustar tamanhos de sprites para ficarem proporcionais.
- Tornar a hitbox do Boss mais justa (redução via inflate + offset).

- **Usei a IA para:**

- Estruturar o código em classes.
- Melhorar estética (fundo, chão, UI).
- Afinar comportamento do Boss e obstáculos.

- **O que aprendeste:**

- Melhor compreensão do loop de jogo, event handling e desenho frame a frame.
- Organização em classes facilita adicionar novas features (mais inimigos, mais power-ups).
- IA como ferramenta de **apoio**, ajuda a gerar código, não substitui além de consolidar o meu entendimento de Pygame