

Trabalho Prático – Fundamentos de Computação Gráfica

JOGO 2D SIMPLES



Figura 1. Uma boa fonte de inspiração são os jogos de Arcade das décadas de 70 e 80. Fonte: Imagem da série Stranger Things (Netflix)

Informações gerais

- **Individual** ou **duplas**
- **Entrega** e apresentação: **11/05/2024**

O objetivo deste trabalho é desenvolver um protótipo de jogo 2D aplicando os conceitos vistos em aula. Para o Grau A, os conhecimentos sobre *desenho de primitivas*, transformações geométricas, câmera 2D e mapeamento de textura serão explorados. Sugere-se como inspiração para pesquisa para a escolha do tema e mecânicas jogos de fliperama, especialmente os mais antigos, como ilustrado na Figura 1.

Instruções de desenvolvimento

O programa deve possuir, como requisitos mínimos:

- Correta criação e utilização dos buffers
 - VAO, VBO(s) e EBO (este último é opcional)
- Correta utilização das transformações de câmera e nos objetos
 - Como está mapeada a matriz de projeção ortográfica 2D?
 - Como são feitas e atualizadas as transformações nos objetos da cena?
- *Sprites* como textura mapeadas em polígonos, para o desenho de:
 - personagem, objetos e itens
 - imagens de fundo
 - As imagens podem ser obtidas da internet ou elaboradas pelos alunos
- Controle da movimentação personagem 2D (com input de teclado e/ou mouse)
- Colisão entre *sprites* (coletar itens, tiros etc, conforme a proposta do jogo)
- Sprites com animação

Requisitos **extras** (adicionam até +1.0 pto à nota do Trabalho):

- Cenário formado por camadas, para implementação do efeito de “parallax scrolling”
- HUDs (Head-Up Displays)¹ com texto (usando, por exemplo, a [FreeType](#))
- Telas de entrada, de jogo e *gameover*
- Som (SDL mixer, OpenAL, FMOD etc)

É recomendado que o jogo tenha um arquivo de configuração, para carregar imagens das camadas e demais configurações necessárias.

SUGESTÕES DE JOGOS

A escolha do tipo de jogo está a critério do aluno. Se estiverem sem nenhuma “inspiração”, seguem algumas ideias que se encaixam perfeitamente em nosso escopo:

- Talvez o jogo mais simples que contenha tudo que precisamos é daqueles em que caem objetos (por exemplo, de cima pra baixo) e você precisa pegá-los movimentando um outro objeto (por exemplo, esquerda e direita). O primeiro jogo nessa linha foi o Catch do Atari (1977), mas existem milhares de jogos similares, normalmente com maçãs caindo.
- Jogos do tipo *endless runner* com visão lateral também se enquadram bem no escopo, como por exemplo o do dino do Google (T-Rex Game)
- Jogos antigos de Atari ou jogos próprios para fliperama mais antigos são ótimas pedidas também. Você pode se inspirar em sites como <https://www.retrogames.cz/index.php>
- Jogos educativos e/ou conscientizadores simples: <http://planeta42.com/biology/>

Apresentação e entrega

- A apresentação será feita em aula. É importante e de grande peso na avaliação que a dupla esteja sincronizada e entenda toda a estruturação do código (classes, decisões de implementação). Pontos que devem ser abordados na apresentação:
 - Apresentação do grupo (nomes)
 - Estrutura geral do código (classes e/ou funções)
 - Estrutura dos buffers e shader(s) (como você especificou os vértices e seus atributos e como você os envia para os shaders)
 - Gerenciamento dos *sprites*: como são armazenadas as informações das transformações, ID de texturas, infos adicionais
 - Como é feito o controle do personagem (teclado e/ou mouse)
 - Como são verificadas as colisões
 - Como é feita a movimentação de outros objetos do cenário
 - Como é gerenciada a animação dos sprites
 - Como são gerenciadas as camadas para o efeito de *Parallax* (se houver)

¹ HUDs são elementos gráficos exibidos na tela para transmitir informações ao jogador (como pontuação, vidas etc). Fonte: <https://www.designerd.com.br/design-de-games-voce-sabe-o-que-e-hud/>

o Mostrar o programa em funcionamento (sugestão: pode fazer isso no início, logo após a apresentação). Se julgar necessário, pode mostrar em outros momentos junto com a explicação.

- **DOCUMENTAÇÃO É TUDO!** Não esqueça de adicionar ao diretório do projeto um arquivo LEIAME.md com informações pertinentes (como instruções para compilação e uso do programa)
- **COMUNICAÇÃO E COLABORAÇÃO SÃO AS CHAVES PARA O SUCESSO.** PROIBIDO FICAR TRANCADO COM DÚVIDAS! Utilize o nosso canal da disciplina no Teams.

BOM TRABALHO! 😊

Lembre-se: um problema grande e complexo pode ser quebrado em problemas menores e mais simples de se resolver!