

Trabalho Prático - Enduro

Pedro O.S. Vaz de Melo

October 17, 2018

1 Descrição do Problema

O objetivo deste trabalho é fazer com que o aluno utilize as técnicas de programação aprendidas na disciplina para desenvolver um jogo eletrônico gráfico semelhante ao *Enduro* do Atari. Nesse jogo, você controla um carro de corrida e o seu objetivo é ultrapassar o maior número de carros possíveis. O jogo pode terminar de diversas formas, mas na forma mais simples ele termina quando o seu carro bate em algum carro adversário.

Ao implementar o jogo, você deverá registrar algum tipo de pontuação. No caso mais simples, essa pontuação pode ser o número de carros que você ultrapassou até o momento. Ao final do jogo deverá ser exibida uma tela informando a pontuação do usuário e o recorde atual. Caso a pontuação do usuário seja menor que o recorde atual, um texto com essa informação deve ser exibido para o usuário. Este trabalho tem um valor total de 20 pontos. Execute os arquivos `enduro.exe` para um exemplo de jogo que pode ser implementado. A versão clássica do jogo pode ser jogada em <http://www.atari2600.com.br/Atari/Roms/01Do/Enduro>.

2 Critérios de Avaliação

2.1 Solução Apresentada

Os seguintes itens **serão** avaliados:

- Movimentação fluida das animações relacionadas ao jogo, como o movimento de aproximação dos carros e de movimentação do carro do jogador (1 ponto);
- O tamanho dos carros devem se alterar de acordo com a distância que eles estão do carro do jogador (3 pontos);
- O jogador deve ser capaz de movimentar o seu carro dentro dos limites da pista (2 pontos);
- Quando um carro adversário atingir o carro controlado pelo jogador, um impacto negativo deve ser gerado ao jogador. No caso mais simples, o jogo acaba (3 ponto);
- O seu jogo deve controlar pelo menos 10 carros adversários, que não precisam estar necessariamente no campo de visão do jogador (3 pontos);

- Os carros inimigos não podem colidir entre eles (2 pontos);
- O cenário deve ter pelo menos o desenho do chão, da pista e do céu (1 ponto);
- Contagem e exibição dos pontos (2 pontos);
- Exibição e armazenamento do recorde (3 pontos);
- Documentação (2 pontos).

2.2 Documentação

Deve conter o Manual de Uso, que descreve como operar o jogo, e detalhes da implementação, que descreve brevemente os trechos de código e as estruturas de dados desenvolvidas por você.

2.3 Conhecimento do Código

Conhecimento do aluno sobre o código apresentado será verificado via entrevista em laboratório. Sua nota total será multiplicada pela sua nota da prova oral, que vale 1. Assim, se você tirar 0.5 na prova oral, sua nota será dividida por 2.

2.4 Pontos Extras

Além dos 20 pontos, o professor pode atribuir até 10 pontos a mais caso o aluno implemente extras, tais como:

- Gerar diferentes tipos de carros adversários;
- Permitir que o jogador acelere e freie;
- Criar diferentes tipos de carros para o usuário controlar;
- Implementar fases;
- Implementar obstáculos;
- Implementar curvas;
- Implementar pistas complexas, com subidas, descidas etc;
- Implementar diferentes visualizações para o cenário, como dia, noite, montanhas etc;
- Criar *addons* e *power-ups* que podem, por exemplo, dar ao jogador mais velocidade ou invencibilidade por um tempo;
- Implementar campeonatos;
- Implementar dinheiro e permitir que o usuário melhore o seu carro à medida que o jogo avança;
- Implementar paradas de pitstop;
- Criar animações sofisticadas;

- Permitir modo de dois jogadores ao mesmo tempo;
- Colocar sons e músicas;
- **Qualquer outro extra que você ache interessante!**

IMPORTANTÍSSIMO: Pontos extras só serão dados aos alunos que obtiveram mais de 50% dos pontos nas provas, ou seja, mais de 35 no somatório das três provas.

3 Como eu faço?

Apesar da descrição fazer o trabalho parecer complicado, ele é bastante simples. Tudo que o aluno precisa saber para desenvolver este jogo são os conhecimentos adquiridos na disciplina e um pequeno entendimento de desenvolvimento de aplicações gráficas. Assim como são necessárias bibliotecas novas para a utilização de funções não nativas da linguagem C, como a `math.h`, uma biblioteca também é necessária para que se utilize funções gráficas. Para este trabalho, pede-se que se utilize a biblioteca Allegro5, que fornece inúmeras funções que podem ajudar no desenvolvimento deste trabalho. Os vídeos abaixo ensinam como instalar a biblioteca Allegro5 em um ambiente Windows com o MingW instalado:

<https://www.youtube.com/watch?v=AezxBP687n8>

<https://www.youtube.com/watch?v=cgqjzJzm00w>

4 Roteiro de Desenvolvimento Sugerido

Como o jogo é complexo, identificar a sequência de funcionalidades que devem ser desenvolvidas pode ser um problema. Assim, a seguir estão descritas etapas de desenvolvimento sugeridas, colocadas em ordem cronológica.

1. Desenhar o cenário do jogo (fácil);
2. Desenhar o carro que o jogador controla (fácil);
3. Permitir que o jogador movimente o carro para a esquerda e para a direita. Não deixe que o carro do jogador ultrapasse os limites da pista (médio);
4. Criar uma função que recebe a coordenada y de um carro e que desenha o carro naquela coordenada. O tamanho do carro deve diminuir conforme y diminui, dando a impressão que o carro está mais longe. Use trigonometria! Para testar, peça para a função desenhar o carro do jogador em diferentes coordenadas y (difícil);
5. Implemente os carros adversários, que devem se aproximar do carro do jogador à medida que o tempo passa. Use um vetor de carros para implementar os (pelo menos) 10 carros adversários (parte mais difícil do jogo);
6. Implemente o teste de colisão do carro do jogador com os carros adversários (médio);
7. Implementar o sistema de pontuação e o exibir na tela (fácil);

8. Implementar o sistema de armazenamento do recorde (fácil);
9. Divirta-se implementando funcionalidades extras (super divertido).