

Trabalho Prático - Agar.Io

Pedro O.S. Vaz de Melo

October 13, 2017

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 *AGAR.IO* ou ao *OSMOS*. Nesse jogo, você controla uma célula em um mapa que representa uma placa de Petri. O seu objetivo é absorver (ou comer) a célula inimiga. Para isso, você deve aumentar a sua quantidade de massa ao comer células menores que a sua, evitando colidir com as células maiores que você. Se você conseguir comer a célula inimiga, que deve estar destacada no jogo (pela cor, por exemplo), o jogo termina.

Além disso, o jogo deverá registrar algum tipo de pontuação. No caso mais simples, esse pontuação pode ser o tempo em que a nave demorou para comer a célula inimiga. 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 *osmos.exe* e *ultraosmos.exe* para exemplos de jogos que podem ser implementados. Ambas as versões receberiam mais que os 20 pontos do trabalho. Use o *mouse* para movimentar a sua célula.

2 Critérios de Avaliação

2.1 Solução Apresentada

Os seguintes itens **serão** avaliados:

- Movimentação fluida das células, ou seja, sem muitos saltos (teletransportes) na tela (2 pontos);
- O clique do *mouse* deve expelir da célula do jogador uma célula menor que se movimentará na direção do clique do *mouse* e que deve acelerar a célula do jogador na direção oposta (3 pontos);
- Pelo menos 50 células (além da sua e a inimiga) podem estar na tela do seu jogo (2 pontos);
- A sua célula deve reduzir de tamanho no clique do *mouse* (2 pontos);
- Contagem e exibição dos pontos (2 pontos);

- Exibição e armazenamento do recorde (2 pontos);
- Colisão das células deve eliminar a célula menor e aumentar o tamanho da maior (2 pontos);
- Fazer com que a área das células no processo de absorção e de aceleração (quando células menores são expelidas) respeite as leis da física, ou seja, se uma célula de área x fundir com uma de área y , a célula final deverá ter área $x + y$ (2 pontos);
- As células sempre devem ser exibidas na tela, ou seja, se elas encostarem em um dos lados da tela, elas devem ricochetear (como no exemplo `bouncer`) (1 pontos);
- Gerar várias células aleatórias para compor o cenário (2 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 desenvolvidos 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:

- Permitir que a sua célula realize outras operações como, por exemplo, se dividir em duas;
- Tornar o mapa maior que o tamanho da tela (ver jogo `ultraosmos.exe`);
- Fazer com que a colisão absorva apenas a parte da célula menor que está em contato com a célula maior (ver jogo `ultraosmos.exe`);
- Implementar fases;
- Implementar obstáculos;
- Criar *addons* e *power-ups* que podem, por exemplo, dar ao jogador mais velocidade ou invencibilidade por um tempo;
- Criar tipos diferentes de células;
- 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>