

Trabalho Prático - Candy Crush

Pedro O.S. Vaz de Melo

October 18, 2019

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 *Candy Crush*. Nesse jogo, doces de diversos tipos são espalhados pela tela, na forma de um *grid* (ou matriz). O objetivo do jogador é movimentar os doces a fim de formar sequências de doces do mesmo tipo que tenham tamanho maior ou igual a três. As sequências podem ser formadas tanto nas linhas quanto nas colunas da matriz. A movimentação dos doces é feita a partir de trocas: o jogador clica no doce que quer movimentar e, mantendo o botão do mouse pressionado, arrasta o cursor para o doce alvo, ou seja, aquele que deve ser trocado de posição com o doce clicado inicialmente. Trocas só podem ser feitas com doces vizinhos. Doces vizinhos são aqueles que estão imediatamente acima, abaixo, à direita ou à esquerda do doce alvo. Ao formar uma sequência, os doces são consumidos pelo jogador, gerando pontos para ele. O consumo imediato de n doces vale mais pontos que o consumo dos n doces em instantes separados. Exemplo: um movimento que consume 6 doces imediatamente vale mais pontos que dois movimentos que consome, em cada um deles, 3 doces. O jogador tem direito a um conjunto finito de movimentos e o jogo termina quando os seus movimentos disponíveis acabarem.

Assim, ao implementar o jogo, você deverá registrar e exibir algum tipo de pontuação e também o número de movimentos disponíveis para o jogador. Ao final do jogo deverá ser exibida uma tela informando a pontuação do usuário e o recorde atual. Caso a pontuação do usuário seja maior que o recorde atual, um texto com essa informação deve ser exibido para o usuário e um novo recorde deve ser registrado. Este trabalho tem um valor total de 20 pontos. Execute os arquivos `candyc.exe` para um exemplo de jogo que pode ser implementado. A versão clássica do jogo pode ser jogada em <https://king.com/game/candycrush>.

2 Critérios de Avaliação

2.1 Solução Apresentada

Os seguintes itens **serão** avaliados:

- Controle preciso dos movimentos dos doces. (*2 pontos*);

- A matriz de doces deve ter dimensões superiores ou iguais a 9 linhas e 6 colunas. *(2 pontos)*;
- A tela deve ser inteiramente preenchida com doces aleatórios, sem nenhuma sequência que vale pontos. *(2 pontos)*;
- O jogo deverá ter pelo menos um estágio em que apenas doces vizinhos poderão ser trocados de lugar. *(1 pontos)*;
- As sequências que valem ponto devem ser realçadas antes de serem eliminadas. *(2 pontos)*;
- Enquanto sequências que valem ponto são eliminadas, o jogador não poderá realizar movimentos. *1 ponto*;
- O jogo deve ter pelo menos 4 tipos de doces. *(2 ponto)*;
- O cenário deve exibir o número de jogadas ainda disponíveis para o jogador *(1 ponto)*;
- O cenário deve exibir os pontos ganhos pelo jogador. *(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.

Exemplos de documentação podem ser baixados em: <https://www.dropbox.com/s/bh1rrsuemci2cwm/exemplosDoc.zip?dl=0>.

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:

- Usar imagens e animações do tipo *sprite*;
- Gerar diferentes tipos de cenários;
- Permitir diferentes tipos de doces, que produzem efeitos diferentes;
- Colocar sons e músicas;
- Implementar animações para o movimento dos doces;

- Implementar o preenchimento da tela assim que uma (ou mais) sequência é eliminada;
- Criar *addons* e *power-ups* que podem, por exemplo, dar ao jogador trocas mais poderosas ou eliminação automática de linhas ou colunas;
- Implementar modo de dois jogadores;
- Implementar dinheiro e permitir que o usuário compre *addons* e *power-ups*;
- **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, sem os doces;
2. Dispor os doces aleatoriamente na tela;
3. Implementar a troca de doces;
4. Implementar a identificação de sequências de doces que valem ponto;
5. Implementar a eliminação dessas sequências;
6. Adicionar os pontos pela eliminação;
7. Implementar o sistema de jogadas disponíveis;
8. Implementar o sistema de armazenamento do recorde;

9. Divirta-se implementando funcionalidades extras;
10. Escrever a documentação.