

Trabalho Prático - MultiTênis

Pedro O.S. Vaz de Melo

June 14, 2022

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 a um jogo de tênis. A principal diferença é que, nesta versão, os jogadores deverão jogar com diversas bolas ao mesmo tempo. Além disso, as paredes laterais são sólidas, ou seja, se a bola bater em uma parede lateral, ela ricocheteia e volta para a quadra. O objetivo dos jogadores é rebater as bolas de volta para o seu oponente. Um jogador ganha 1 ponto cada vez que o oponente falhar em rebater uma bola. Para rebater uma bola, o jogador deve pressionar uma tecla (ex: espaço). O jogo é de dois jogadores e cada jogador deve ter o seu conjunto de teclas de controle. Vence o jogo aquele que fizer 10 pontos primeiro.

O jogo começa sem bolas em jogo. A cada 5 segundos, uma nova bola deve ser arremessada na quadra. As bolas são arremessadas dos quatro cantos da quadra, começando pelo canto superior direito, depois pelo superior esquerdo, depois pelo inferior esquerdo e depois pelo inferior direito. Lançamentos de bola devem ocorrer sempre nessa ordem e só devem ser interrompidos quando o número de bolas na quadra atingir um limite. Esse limite pode ser definido pelo aluno, mas deve ser, pelo menos, de 12 bolas. É importante garantir também que o lançamento faça a bola cruzar a quadra em algum momento, mesmo que depois de bater várias vezes nas paredes laterais. As bolas devem ter diferentes tamanhos, velocidades e devem ser lançadas usando ângulos diferentes a cada lançamento.

Por fim, o seu jogo deve registrar o histórico (número de vitórias de cada jogador) das partidas realizadas entre os jogadores. Para isso, armazene cada vitória (e derrota) em um arquivo. No final do jogo, esse histórico deve ser exibido na tela, incluindo também o resultado da última partida que jogaram. Uma versão básica desse jogo vem acompanhada do **kit de desenvolvimento Allegro** disponível na página da disciplina.

2 Critérios de Avaliação

Este jogo pode ser tão complexo quanto você deseja, mas há uma versão básica que lhe garante os 20 pontos do trabalho prático. Abaixo as funcionalidades que devem ser implementadas na versão básica do jogo:

- **Controle preciso dos movimentos dos jogadores.** O jogador deve ter acesso a teclas que controlam a movimentação e rebatida dos jogadores. *(2 pontos)*
- **Rebatida.** A rebatida da bola só deve ocorrer quando o desenho da bola estiver em contato com o desenho do jogador e quando a bola estiver vindo em direção ao jogador (exemplo: de cima para baixo quando o jogador que for fazer a rebatida é o localizado na parte inferior da tela). A rebatida deve sempre arremessar a bola de volta para (na direção) o jogador oponente. *(3 pontos)*
- **Canhões de bola.** As bolas devem ser arremessadas a cada 5 segundos dos cantos da tela, começando pelo canhão do canto superior direito e passando por todos os outros cantos no sentido horário. *(3 pontos)*
- **Quantidade de bolas.** O número de bolas que podem estar ao mesmo tempo em quadra deve ser limitado a, pelo menos, 12 bolas e, no máximo, a 20 bolas. Canhões devem parar de lançar bolas quando esse limite é alcançado. A sequência de lançamento não deve ser alterada quando o limite é alcançado. *(3 pontos)*

- **Movimento das bolas.** As bolas devem ricocheteiar nos cantos laterais da tela. *(1 ponto)*
- **Limites de movimento dos jogadores.** Os jogadores não devem ultrapassar os limites do seu espaço na quadra. Em outras palavras, nenhuma porção do desenho do jogador deve sumir durante o jogo e invadir a metade da quadra do oponente. *(1 pontos)*
- **Pontuação.** Cada bola que não for rebatida deve dar 1 ponto ao jogador adversário. *(1 pontos)*
- **Fim de jogo.** O jogo deve terminar quando um dos jogadores alcançar 10 pontos. *(1 ponto)*
- **Exibição do histórico de partidas.** O histórico (número de vitórias de cada jogador) das partidas realizadas entre os jogadores deve ser armazenado em um arquivo. No final do jogo, esse histórico deve ser exibido na tela, incluindo também o resultado da última partida que jogaram. *(3 pontos)*
- **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 na página da disciplina ou através do link <https://drive.google.com/open?id=1KP15y2DVEZqTW-Rrhor5SFhGLffG0J0r>. *(2 pontos)*

2.1 Conhecimento do Código

Conhecimento do aluno sobre o código apresentado será verificado via prova oral, que será dada no formato de uma entrevista. 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.

2.2 Pontos Extras

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

1. Usar imagens e animações do tipo *sprite*;
2. Gerar diferentes tipos de cenários;
3. Permitir diferentes tipos de ataques, que produzem efeitos diferentes;
4. Colocar sons e músicas;
5. Implementar animações para o movimento das ações;
6. Implementar fases e sistema de *games* e *sets*;
7. Implementar efeitos físicos realistas (gravidade, força da rebatida etc).
8. Criar *addons* e *power-ups* que podem, por exemplo, dar ataques e defesas especiais aos jogadores;
9. Implementar modo de um jogador, que joga contra uma parede sólida (paredão);
10. Implementar modo de um jogador, que joga contra um adversário controlado pelo computador;
11. Implementar opção de salvar o jogo;
12. Implementar mais de um tipo jogador, de bola, e canhão;
13. Implementar menus e diferentes níveis de dificuldade;
14. Implementar obstáculos nas quadras que dificultam a ação dos jogadores;
15. **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 36 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.
2. Criar a estrutura dos jogadores.
3. Criar dois jogadores e desenhar eles na tela.
4. Movimentar os jogadores.
5. Garantir que eles não invadam o espaço do outro e nem saiam da quadra.
6. Criar a estrutura para os canhões e o vetor de quatro canhões, um em cada canto da quadra.
7. Criar estrutura para a bola e um vetor de 12 bolas (se o limite for 12).
8. Criar a rotina que dispara uma bola a partir de um canhão.
9. Criar a rotina que atualiza as posições das bolas.
10. Criar a rotina que desenha as bolas.
11. Criar rotina que trata colisão dos jogadores com as bolas.
12. Implementar o sistema de pontuação e sua exibição na tela.
13. Implementar o recorde.