

JOGOS EM REDE MULTIUSUÁRIOS - GRUPO 9

RELATÓRIO UNIDADE 3

Alunos:

Gabriel Cristian Melo da Silva

Thiago da Costa Monteiro

1. Descrição Geral do Jogo

Nessa atividade, foi feito um jogo **PONG** em **C++** que funciona em tempo real, utilizando [este tutorial](#) como base. Apesar do tutorial ser de 2012, poucos comandos precisaram ser mudados por depreciação dos anos. No Pong, o jogador deve mover a sua raquete para evitar que a bola chegue na borda da sua tela e rebata-lá para que chegue na borda do adversário. O jogo acaba quando um dos jogadores atingir quatro pontos, caso haja uma diferença de dois pontos entre os jogadores (E.G. caso o jogo esteja 4x3, ele não irá acabar, mas se chegar em 4x6, será uma vitória do Player2).

2. Arquitetura

Modelo de comunicação: Foi realizada a comunicação utilizando o modelo **Cliente-Servidor**, onde o servidor tem como principal propósito a transferência de mensagens, com a lógica acontecendo dentro do programa cliente. Não foram adotadas medidas de prevenção contra dessincronização.

Tipos de comandos:

- Para o player1, seta para **cima** move a raquete para cima, seta para **baixo** move para baixo;
- Para o player2, **“W”** move a raquete para cima, **“S”** move a raquete para baixo
- A tecla **“ESC”** serve para fechar jogo;

Mensagens para atualização de estado: A troca de mensagens que ocorre é feita via um array que contém o valor de um enum que representa a situação do jogo (**INTRO**, **PLAYING**, **P1WON**, **P1LOST**) e um inteiro que indica se o player moveu para cima, baixo ou não moveu. Os jogadores jogam com suas respectivas peças, de forma que o jogo ocorre na tela de cada um, mas com as raquetes sincronizadas com o do adversário. Para isso, foi convencionado que o primeiro jogador a conectar é o **Player1** e o outro jogador será o **Player2** e essa informação é enviada via mensagem de rede ao estabelecer a conexão com sucesso.

3. Implementação

Foi utilizada a biblioteca **SFML2** para o visual do jogo e as bibliotecas de Socket que utilizamos nas atividades anteriores para a conexão. Tudo foi testado num sistema WSL 2.0 Debian rodando no Windows 10.

Papéis dos membros:

Fizemos a edição do código em conjunto, utilizando a ferramenta de compartilhamento do **Visual Studio Code**, então não teve uma divisão clara das tarefas, visto que muitos dos problemas encontrados foram solucionados através de esforço conjunto.

- **Gabriel:** Preparou o ambiente, as bibliotecas e montou o esqueleto do projeto e trabalho em comum;

- **Thiago:** Fez a leitura do código para encontrar erros de sintaxe e trechos de código que não faziam sentido, propôs testes durante os momentos que erros foram encontrados e trabalho em comum.

4. Instruções de Compilação

1. Abrir o terminal na pasta do programa e execute o comando **"g++ -c pong.cpp"** para criar o código objeto do cliente;
2. Em seguida, execute o comando **"g++ pong.o -o pong-app -lsfml-graphics -lsfml-window -lsfml-system"** para compilar o cliente via **SFML2**;
3. Então, execute o comando **"g++ -o pongServer pongServer.cpp"** para compilar o servidor;
4. Abrir o servidor com o comando **"./pongServer"**;
5. Abrir os dois clientes em novos terminais com o comando **"./pong-app"**;

A tela do jogo ficará preta até ambos os jogadores estiverem conectados. Em seguida, basta pressionar qualquer tecla para começar o jogo. Para encerrar o jogo basta pressionar a tecla **ESC**. No arquivo compactado, vai existir um arquivo para a fonte **"Tomb.ttf"**, que deve ficar na mesma pasta dos executáveis ou ser instalada no sistema para que o jogo possa executar.