

Programação Concorrente

Trabalho Prático

Duelo

Grupo de Sistemas Distribuídos
Universidade do Minho

1 de abril de 2025

Informações gerais

- Cada grupo deve ser constituído por até (e de preferência) quatro elementos.
- O trabalho deve ser entregue até 18 de maio de 2025 às 23h59.
- Deve ser entregue o código fonte e ainda um relatório até 6 páginas no formato pdf.
- A apresentação do trabalho será a em data a confirmar, possivelmente entre 30 de maio e 5 de junho de 2025.

Resumo

Implemente um mini-jogo onde vários utilizadores podem interagir usando uma aplicação cliente com interface gráfica, escrita em Java, intermediados por um servidor escrito em Erlang. O avatar de cada jogador movimenta-se num espaço 2D. Os vários avatares interagem entre si e com o ambiente que os rodeia, segundo uma **simulação efetuada pelo servidor**.

Funcionalidade

Este jogo deverá suportar as seguintes funcionalidades:

- Registo de utilizador: dado *username* e *password*; deverá ainda ser possível um utilizador cancelar o registo. Sempre que um jogador quer entrar no jogo deverá ser autenticado pelo servidor. Um jogador começa no *nível* 1 quando se regista. O registo poderá ser feito por um cliente específico (independente do cliente gráfico principal) que guarde em ficheiro a informação do registo.
- Progressão: um jogador passa do nível n para o nível $n + 1$ depois de ganhar n partidas consecutivas (antes de mudar). Desce de nível se perder $\lceil n/2 \rceil$ partidas consecutivas (antes de mudar), até chegar ao limite inferior $n = 1$.
- Partidas: cada partida é constituída por 2 jogadores, com diferença máxima de um nível entre eles; quando um jogador pede para jogar tem que aguardar até tal ser possível. O servidor deve permitir que em cada momento possam estar a decorrer várias partidas em simultâneo.
- Espaço: o espaço é 2D, retangular, vazio, delimitado por paredes nos quatro lados.

- Avatares: os avatares (jogadores) são em forma de círculo. Inicialmente, aparecem no mapa em lados opostos.
- Movimentação dos jogadores: deverá ser feita através de quatro teclas (cima, baixo, esquerda, e direita), que controlam a aceleração. Admite-se que os avatares têm inércia, ou seja, acelerar para um dos lados aplicará uma força que irá aumentar a velocidade do jogador progressivamente para esse lado, até atingir uma velocidade máxima pré-definida. Já acelerar para o lado oposto irá progressivamente desacelerar o movimento atual, invertendo eventualmente a direção.
- Projéteis: cada jogador pode disparar um projétil a cada intervalo de tempo pré-definido, em forma de círculo. A velocidade do projétil é pré-definida, enquanto que a direção é controlada pela posição do cursor em relação ao avatar do jogador.
- Modificadores: deverão ir aparecendo aleatoriamente, ao longo do tempo, quatro tipos de modificadores com diferentes cores, até um dado máximo (por tipo) de modificadores presentes. Estes são também em forma de círculo, de tamanho entre o dos projéteis e o dos jogadores, estando sempre parados. Quando consumidos, desaparecem e aplicam as seguintes modificações ao respetivo jogador:
 - Verdes (Laranjas): Aumentam (diminuem) a velocidade de saída de projéteis subsequentes. A velocidade de saída voltará progressivamente ao valor base.
 - Azuis (Vermelhos): Diminuem (aumentam) o intervalo de tempo entre disparos. O intervalo de tempo voltará progressivamente ao valor base.
- Colisões: uma colisão entre um jogador e um projétil resultará na adição de 1 ponto na pontuação do outro jogador, bem como na eliminação do projétil. Uma colisão entre um projétil e uma borda eliminará o projétil. Uma colisão entre um jogador e uma das bordas do mapa resultará na adição de 2 pontos na pontuação do outro jogador, e recolocar os dois jogadores nas suas posições iniciais. Uma colisão entre um jogador e um modificador aplicará o respetivo efeito nesse jogador e removerá o modificador. Uma colisão entre jogadores ou entre projéteis e modificadores deverá ser ignorada.
- Pontuação: ganha o jogador que, ao fim de 2 minutos, tiver a maior pontuação. Caso tiverem os mesmos pontos, assume-se o empate, sendo a partida ignorada para efeitos de nível e *top*, como se não tivesse ocorrido.
- Listagem do *top* 10 de jogadores: ordenada primeiro por nível e depois pela última série de vitórias/derrotas consecutivas.

Cliente

Deverá ser disponibilizado um cliente com interface gráfica que permita suportar a funcionalidade descrita acima. Este cliente deverá ser escrito em Java e comunicar com o servidor via sockets TCP. Sugestão: utilize Processing (<http://processing.org>) para a interface gráfica.

Servidor

O servidor deverá ser escrito em Erlang, mantendo em memória a informação relevante para fazer a simulação do cenário descrito, receber conexões e input dos clientes bem como fazer chegar a estes a informação relevante para a atualização da interface gráfica, de acordo com a simulação que efetua.