

# Programação Concorrente

## Trabalho Prático

### Buracos

Grupo de Sistemas Distribuídos  
Universidade do Minho

24 de Abril de 2019

#### Informações gerais

- Cada grupo deve ser constituído por até três elementos.
- O trabalho deve ser entregue até 5 de Junho de 2019.
- Deve ser entregue o código fonte e ainda um relatório até 6 páginas no formato pdf.
- A apresentação do trabalho será dia 7 de Junho de 2019.

#### Resumo

Implemente um mini-jogo onde 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 avatares interagem entre si e com o ambiente que os rodeia, segundo uma simulação efectuada 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.
- Partidas: cada partida é constituída por dois jogadores; 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. Cada partida demora dois minutos.
- Espaço: o espaço é 2D, rectangular, vazio, limitado nos quatro lados, sendo todo visível no ecrã. Quando um jogador atinge um limite, é impedido de sair, mas mantém a componente da velocidade tangencial à fronteira.
- Avatares: todos os avatares (de jogadores e objectos) são em forma de círculo, cujas cores são: preto para os jogadores, verde para objectos *comestíveis*, e vermelho para objectos *venenosos*. No caso dos jogadores, a circunferência do avatar (border) tem cor azul para o próprio e vermelha para o outro jogador. Cada objecto tem um tamanho fixo definido na sua criação, podendo existir

objectos de vários tamanhos. O tamanho de cada jogador varia ao longo do jogo, sendo a área do círculo proporcional à massa do jogador, que nunca baixa de um dado valor mínimo.

- Movimentação dos jogadores: deverá ser feita através de 4 teclas (esquerda, direita, cima, baixo) que provocam uma força na direcção respectiva enquanto premidas. A aceleração resultante depende da massa do avatar. Várias teclas podem estar premidas ao mesmo tempo, e.g., direita e cima.
- Colisões: qualquer sobreposição entre jogadores e objectos venenosos fazem o objecto desaparecer e penalizam o jogador, subtraindo a massa do objecto à do jogador. Todas as outras colisões são ignoradas.
- Capturas: quando um jogador se sobrepõe completamente a outro avatar de tamanho menor, de outro jogador ou objecto comestível, engole-o, capturando a massa de um objecto comestível, ou um quarto da massa do outro jogador. Um jogador que tenha sido capturado reaparece noutra posição aleatória.
- Povoação do espaço: no início do jogo o espaço é povoado com objectos dos dois tipos, nomeadamente com objectos comestíveis de diferentes tamanhos, menores e maiores do que os jogadores. À medida que objectos são capturados, aparecem outros em posições aleatórias. Deve ser garantido que existe sempre pelo menos um objecto comestível menor do que o menor jogador.
- Pontuação: a pontuação de um jogador é a maior massa atingida durante a partida.
- Tops: para além da sua pontuação corrente, deverá ser mostrado a cada jogador o top das pontuações desde que o servidor foi arrancado.

## 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: caso não esteja familiarizado(a) com nenhuma API para gráficos, 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 esta informação relevante para a actualização da interface gráfica.