

Programação Concorrente

Trabalho Prático

Vaga Vermelha

Grupo de Sistemas Distribuídos
Universidade do Minho

23 de Abril de 2018

Informações gerais

- Cada grupo deve ser constituído por até três elementos.
- O trabalho deve ser entregue até 30 de Maio de 2018.
- Deve ser entregue o código fonte e ainda um relatório até 6 páginas no formato pdf.
- A apresentação do trabalho será dia 1 de Junho de 2018.

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. Um jogador começa no *nível* 1 quando se regista,
- Progressão: um jogador passa do nível n para o nível $n + 1$ depois de ganhar n partidas.
- Partidas: cada partida é constituída por dois 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, rectangular, vazio, sem limites nos quatro lados.
- Movimentação dos jogadores: deverá ser feita através de 3 teclas: (esquerda, direita, frente) que controlam 3 propulsores, provocando aceleração enquanto estão a ser premidas; as duas primeiras provocando aceleração angular, na direcção respectiva, e a terceira aceleração linear na direcção para onde está voltado o jogador. Cada propulsor gasta energia enquanto está a ser actuado; as baterias são carregadas lentamente.

- Criaturas: existem dois tipos de criaturas: as verdes, existindo duas em cada momento, que quando capturadas (via colisão, desaparecendo) dão energia, até ao máximo da bateria, aparecendo uma nova quando alguma é capturada; e as vermelhas, aparecendo uma nova a cada 10 segundos numa posição aleatória. As criaturas deverão ter um movimento lento e suave, tentando aproximar-se do jogador mais perto.
- Avatares: os avatares (de jogadores e criaturas) são em forma de círculo. Os jogadores têm uma direcção para onde estão voltados, sinalizada graficamente.
- Escudo: cada jogador tem um escudo de energia, que causa uma força de repulsão no outro jogador, inversamente proporcional ao quadrado da distância entre eles.
- Colisões: colisões entre criaturas ou entre jogadores são ignoradas.
- Pontuação: a pontuação de um jogador é o tempo decorrido entre este começar a jogar e perder (ou sair voluntariamente). Um jogador perde se sair do espaço de jogo ou colidir com alguma criatura vermelha.
- Tops: para além da sua pontuação corrente, deverá ser mostrado a cada jogador o top das pontuações, e o top dos níveis de jogador, 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 estes a informação relevante para a actualização da interface gráfica.