

Programação Concorrente

Trabalho Prático

Choque de Glutões

Grupo de Sistemas Distribuídos
Universidade do Minho

5 de Maio de 2021

Informações gerais

- Cada grupo deve ser constituído por até, e de preferência, três elementos.
- O trabalho deve ser entregue até 13 de Junho de 2021 às 20h00;
- Deve ser entregue o código fonte e ainda um relatório até 6 páginas no formato pdf.
- A apresentação do trabalho será nos dias 14 e/ou 15 de Junho de 2021.

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 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.
- Participação no jogo: o jogo decorre em contínuo, numa única instância no servidor, com até no máximo 3 jogadores em simultâneo. Quando o limite máximo é atingido, um jogador que queira participar deve ficar em fila de espera, até um jogador corrente sair (voluntariamente ou perdendo).
- Espaço: o espaço é rectangular, sendo delimitado por paredes nos quatro lados. Existem também alguns obstáculos fixos, em forma de círculo, de diferentes tamanhos, dispostos aleatoriamente no espaço.
- Avatares: todos os avatares (jogadores e criaturas) são em forma de círculo, e com uma indicação da direcção para onde estão voltados. O tamanho (raio) varia entre um mínimo e um máximo, diminuindo com o tempo até atingir o valor mínimo.

- 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.
- A velocidade máxima depende do tamanho, sendo tanto menor quanto maior o avatar. A agilidade (aceleração) é independente do tamanho, diminuindo com o tempo até atingir o limite mínimo.
- Criaturas: deverão ir aparecendo aleatoriamente no mapa dois tipos de criaturas, até um limite máximo: As verdes, que representam comida, que quando capturadas (via colisão) dão bónus de tamanho e agilidade ao jogador (até serem atingidos os limites máximos; sugestão: use ganhos decrescentes). As vermelhas, venenosas, retiram agilidade ao jogador. As criaturas deverão ter um movimento rápido e com mudanças de direcção suaves.
- Colisões: uma colisão entre jogador e criatura faz desaparecer a criatura. Uma colisão entre criaturas ou com paredes ou obstáculos fixos deve ser perfeitamente elástica.
- Pontuação: um jogador ganha um ponto quando consegue atingir outro jogador de menor tamanho. Ganha também tamanho mas perde agilidade. O jogador atingido sofre reset da pontuação, reaparece num local aleatório, perde tamanho mas ganha agilidade.
- Um jogador perde (saindo do jogo) quando, tendo o tamanho o mínimo, colide com uma criatura venenosa, obstáculo, ou parede.
- Listagem de pontuações: deverá ser mostrado a cada jogador as pontuações correntes de todos, bem como o top das pontuações máximas atingidas desde que o servidor foi arrancado.

Cliente

Deverá ser disponibilizado um cliente com interface gráfica que permita suportar a funcionalidade descrita acima, nomeadamente: visualizar o avatar de cada jogar, as criaturas, os obstáculos, e as pontuações. 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 actualização da interface gráfica.