

Programação Concorrente

Trabalho Prático

Gravidade

Grupo de Sistemas Distribuídos
Universidade do Minho

1 de maio de 2017

Informações gerais

- Cada grupo deve ser constituído por até três elementos.
- O trabalho deve ser entregue até 31 de Maio de 2017.
- Deve ser entregue o código fonte e ainda um relatório até 6 páginas no formato pdf.
- A apresentação do trabalho será dia 2 de Junho de 2017.

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.
- Participação no jogo: o jogo decorre em contínuo, numa única instância no servidor, com até no máximo 4 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 2D, rectangular, vazio, sem limites nos quatro lados.
- Planetas: existem alguns (poucos) planetas, de diferentes massas, dispostos aleatoriamente no espaço, com velocidades iniciais aleatorias. Em alternativa, e preferencialmente, poderá ser escolhido aleatoriamente um de vários cenários iniciais previamente gerados, que provoquem movimentações “interessantes”.

- Avatares: os avatares (de jogadores e planetas) são em forma de círculo, com o tamanho (área) de acordo com a massa do avatar. Os jogadores têm massa igual e muito menor do que os planetas. Os jogadores têm uma direcção para onde estão voltados, sinalizada graficamente.
- Simulação de movimento: cada jogador e planeta deslocam-se pelas regras da mecânica clássica, considerando a massa concentrada no seu centro. Os jogadores têm ainda velocidade angular, que não afecta esta simulação.
- 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.
- Escudo: cada jogador tem um escudo de energia, que causa uma força de repulsão, em qualquer jogador que se aproxime, inversamente proporcional ao quadrado da distância entre eles.
- Colisões: colisões entre planetas ou entre jogadores são ignoradas (considerados como ponto); se um jogador colide com um planeta, perde imediatamente.
- 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 algum planeta.
- Listagem de pontuações: 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: 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.