



Curso: Mestrado Integrado em Engenharia Informática
Disciplina: Laboratórios de Informática I
Ano letivo: 2018/2019

Universidade do Minho

Relatório

Jogo de Tanks

Grupo: 2018li1g101

Autores:

A84288 José Pedro Gomes Rodrigues dos Santos

A84220 Pedro Miguel Queirós Gomes

Data: 28 de Dezembro de 2018

Introdução

O trabalho que nos foi proposto, que consiste na realização de um jogo de tanks em haskell, foi um projeto que no nosso ponto de vista muito bom visto que nos ajudou bastante a melhorar a programação na linguagem em questão pois foi nos apresentado um conjunto de desafios, que vão ser aprofundados mais a frente, que nos fez crescer como programadores.

Relatório referente à Tarefa 3

Introdução

A tarefa 3 do projeto da cadeira consistia em criar duas funções uma que comprimissem o estado do jogo e outra que descomprimisse de modo a que no fim não houvesse alterações no estado com a melhor compressão possível. Isso na nossa ótica foi alcançado visto que tivemos uma compressão na banda dos 97% sem qualquer perda de informação.

Objetivos

Para tais resultados que foram indicados na Introdução usamos várias técnicas de compressão/descompressão sendo elas divididas em três partes: jogadores, mapa e disparos. Vamos por começar por aprofundar a compressão dos jogadores.

Nos Jogadores nós optamos por primeiro comprimir a Direção por o simples facto que nós acharmos que futuramente ao descomprimir seria mais fácil encontrar o início de cada jogador na string, para isso usamos a função show que transforma qualquer coisa em uma string. De seguida foi a posição do Jogador que para tal criamos uma função, posicaoString, que passa a posição grelha do Jogador para uma string usando o show, mas retira os parênteses. De seguida as vidas, lasers e choques que seguiram a mesma lógica se for maior que zero fica uma letra a representar v,n,b e o número respetivamente, utilizando o show e se não for maior que zero fica vazio. As letras a representar serve para depois na compressão identificar que números são das vidas choques laser. Isto tudo com o objetivo de os jogadores criarem uma string o mais pequena possível.

Agora vamos falar da parte de comprimir o mapa. Aqui nós começamos por comprimir a dimensão do mapa, mas com a particularidade que se o mapa for quadrado só vai retornar um número e se não for retorna o número de linhas e colunas. Em relação ao mapa quando o comprimimos definimos os dois tipos de bloco e a Vazia como #, | _ para diminuir o número de caracteres e usamos estes caracteres ao invés de números porque nós decidimos usar a estratégia de contar quantas vezes um elemento do mapa aparece repetido seguido ou seja se por exemplo na 1 linha temos as 5 primeiras posições são vazias vai aparecer na string _5, isto com o objetivo de reduzir sempre o número de caracteres, mas sem contabilizar a borda do mapa que não escreviamos visto que já sabíamos pela dimensão do mapa a onde teria de ter borda.

Por fim na parte na compressão dos disparos demos uma letra a,l,c para representar cada disparo seguido do número do jogador, depois a direção do disparo e a seguir a posição do disparo sem os parênteses tal como fizemos para a posição do jogador tudo isto para como sempre obter o menor número possível de caracteres sem perder informação.

Na descompressão invertemos o processo da compressão identificando os disparos pela letra representativa deles e separando os jogadores na string sabendo que eles começavam por umas destas letras B,E,D,C e deste modo nunca perdemos informação do mapa dos jogadores ou dos disparos conseguindo assim formar o estado de novo.

Conclusão

Na nossa opinião os nossos resultados foram muito bons visto que obtivemos uma compressão bastante alta sem perda de informação. Por outro lado, também sabemos que podíamos ter feito melhor visto que a compressão podia ser mais alta se usássemos um dicionário, mas não tivemos tempo para implementar tal estratégia.

Relatório referente à Tarefa 5

Introdução

A tarefa 5 do projeto da cadeira consistia em construir uma interface gráfica para o jogo que temos vindo a desenvolver. Acrescentado algumas coisas que achássemos que traziam benefícios ao jogo.

Objetivos

Na tarefa5 nos começamos por dividir por módulos para melhor compreensão e melhor organização e também para melhores tempos de compilação. Na parte de desenhar o mapa preferimos ir com um estilo mais simplista com uma linha a definir a borda do mapa e um fundo branco. Decidimos fazer umas caixas com informações sobre os jogadores que são atualizadas sempre que existe uma alteração nas informações do jogador. Damos a possibilidade de escolher entre vários mapas pré-definidos por nós e também de escolher em qual jogador quer que o bot seja ativado. Decidimos também dar várias cores aos tanks para melhor compreensão de qual é o tank de cada jogador. O nosso objetivo desde o início foi fazer uma interface gráfica fácil de usar para um utilizador que a use pela 1ª vez mas que seja precisa e fiável na representação do jogo dando ao mesmo tempo varias maneiras e mapas de jogar. Além disso , tem Menu de pausa, Inicio e quando um jogador mata tudo aparece que jogador ganhou.

Conclusão

Na nossa opinião achamos que foi um resultado positivo a realização da interface gráfica, mas sabemos que podíamos ter feito melhor no caso de dar a possibilidade de o utilizador criar os seus próprios mapas, mas infelizmente devido a alguns contratempos, não tivemos tempo. Embora mesmo assim achamos satisfatório o nosso resultado.

Relatório referente à Tarefa 6

Introdução

A tarefa 6 do projeto da cadeira consistia em construir um robô que jogasse sozinho contra outros adversários.

Objetivos

Na tarefa 6 começamos por pensar que o mais importante seria o robô se defender, quer seja desviando as balas se conseguir ou então destruir as balas com os seus canhões, ou se o jogador estiver alinhado destruindo com o laser. A seguir testamos a ver se um jogador suficientemente perto num raio de 3 posições está ao alcance de um choque. Se estiver usamos o choque, imobilizando o jogador, partindo para o ataque, ao atacar, basicamente atacamos todos os jogadores que estejam ao alcance de 5 posições e ao alcance do canhão. Caso não consigamos atacar, começamos a destruir paredes destrutíveis de forma a conseguir chegar ao jogador mais próximo. Por fim, fazemos com que o robô se mova para o jogador mais próximo.

Conclusão

Pelos nossos testes o robô parece praticamente indestrutível, muito embora, às vezes descobramos coisas a melhorar. Achemos que conseguimos uma performance bastante razoável.

Conclusão Final

Por fim, penso que o projeto cumpriu com a maior parte das especificações que foram dadas, apesar de já termos feito LI2 e já sabermos trabalhar em grupo, cada vez mais notamos que é mais fácil o trabalho de equipa e este projeto contribuiu muito para isso. Este projeto serviu e muito para consolidar o nosso “knowledge” de haskell, pois já tivemos haskell o ano passado mas não tínhamos LI1 e, portanto, não “trabalhamos” muito a linguagem. Era suposto acrescentarmos mais algumas funcionalidades à parte gráfica, tanks mais fortes, por exemplo, mas por falta de tempo não foi possível.