

**Manual de Utilização**

**Projecto Final**

****

Programação Concorrente e Distribuída

**2011/2012**

Trabalho realizado por:

Miguel Oliveira Nº 38427

João Barreto Nº 38443

****** ***Server***

Antes de qualquer operação, o ficheiro ***Server.java*** deverá ser inicializado para que todos os clientes possam ligar-se ao mesmo. O servidor suporta um máximo de 3 jogadores por jogo criado, sendo este valor expansível até 8 numa próxima *release*. Nesta versão apenas existem 3 packs de *characters*, ou seja, três conjuntos de personagens de jogo com cores distintas.

Operações incluídas na consola do servidor:

* Comando “/reboot” – reinicializa o servidor. Apenas funciona em alguns sistemas operativos.
* Comando “/clear” – limpa a consola de todos os caracteres.
* Comando “/?” – mostra os comandos disponíveis no servidor.

 ***Client*** (optimizado para uma resolução de 650 x 850)

A quando da inicialização, o cliente efectuará uma ligação ao *Master Server*, caso tal não aconteça, o IP do servidor deverá ser verificado/modificado no ficheiro *ClientFinal.java*.

***Create Game*:**

Após a ligação automática, surgirá o ecrã de boas vindas, o jogador deverá criar 1 jogo, introduzindo o nome pelo qual será reconhecido no jogo, o número máximo de jogadores e o nome que pretende atribuir ao jogo.

**ATENÇÃO:** cada jogo criado só será inicializado após todos os lugares estarem preenchidos. O criador de um jogo jogará sempre com ***characters* vermelhos**.

***Join Game*:**

Após entrada na aplicação, poderá ligar-se a um dos vários jogos a decorrer no servidor. A cor atribuída ao jogador será de acordo com a ordem de entrada no jogo seleccionado, ou seja ao segundo jogador serão atribuídos *characters* azuis e amarelos para o terceiro jogador.

**Início de Jogo:**

O início do jogo será indicado pelo sinal sonoro e pelo aparecimento dos *characters* no ecrã.

**Mover *characters*:**

Para mover um *character* deverá selecciona-lo com *mouse* *click* esquerdo e novamente com mouse *click* esquerdo deverá seleccionar o local de chegada. Se por engano for seleccionado o *character*, deverá usar o *mouse click* direito para cancelar a selecção do *character* escolhido. Poderá, se assim o desejar, indicar várias instruções de movimento.

Existem espalhadas pelo mapa pedras que funcionam como bloqueadores de movimento. O *character* terá de contornar o obstáculo para poder prosseguir a sua rota. Todos os *characters* têm velocidades de deslocação diferente.

***Combat:***

Os combates entre *characters* serão iniciados automaticamente caso existam inimigos na sua área de ataque. Uma situação de combate será indicada no ecrã através de uma linha vermelha que liga o atacante ao seu *target* e respectiva perda de vida indicada no *HUD[[1]](#footnote-1)*. O *character* atacado irá ripostar contra o agressor.

***Attack Power:*** cada *character* tem um *attack* *power* diferente, sendo que a unidade mais forte é o *King*.

***Attack Mitigation:*** o ataque será mitigado pela defesa de cada inimigo e reduzido no caso de existirem vários inimigos no respectivo *range*.

***End Game:***

O jogo funciona numa filosofia *gaming* de LMS, *Last Man Standing*, ou seja, o jogo termina apenas quando um dos jogadores conseguir derrotar os *characters* do tipo *King* em jogo. Só assim o jogo terminará.

***Chat System***

Através do sistema de *chat* é possível comunicar com todos os jogadores no mesmo jogo.

**Som**

O sistema de música está ligado por defeito. Caso queira desligar, terá de aceder ao ficheiro *ClientFinal.java* e modificar a variável booleana *soundIsOn*.

1. **HUD** (**heads-up display**) método pela qual a informação é visualmente passada ao jogador. [↑](#footnote-ref-1)