

Projecto Jogo Bejeweled



Projecto Elaborado por:

Bruno Madureira, nº2011161942

Christophe Oliveira, nº2011154912

Introdução

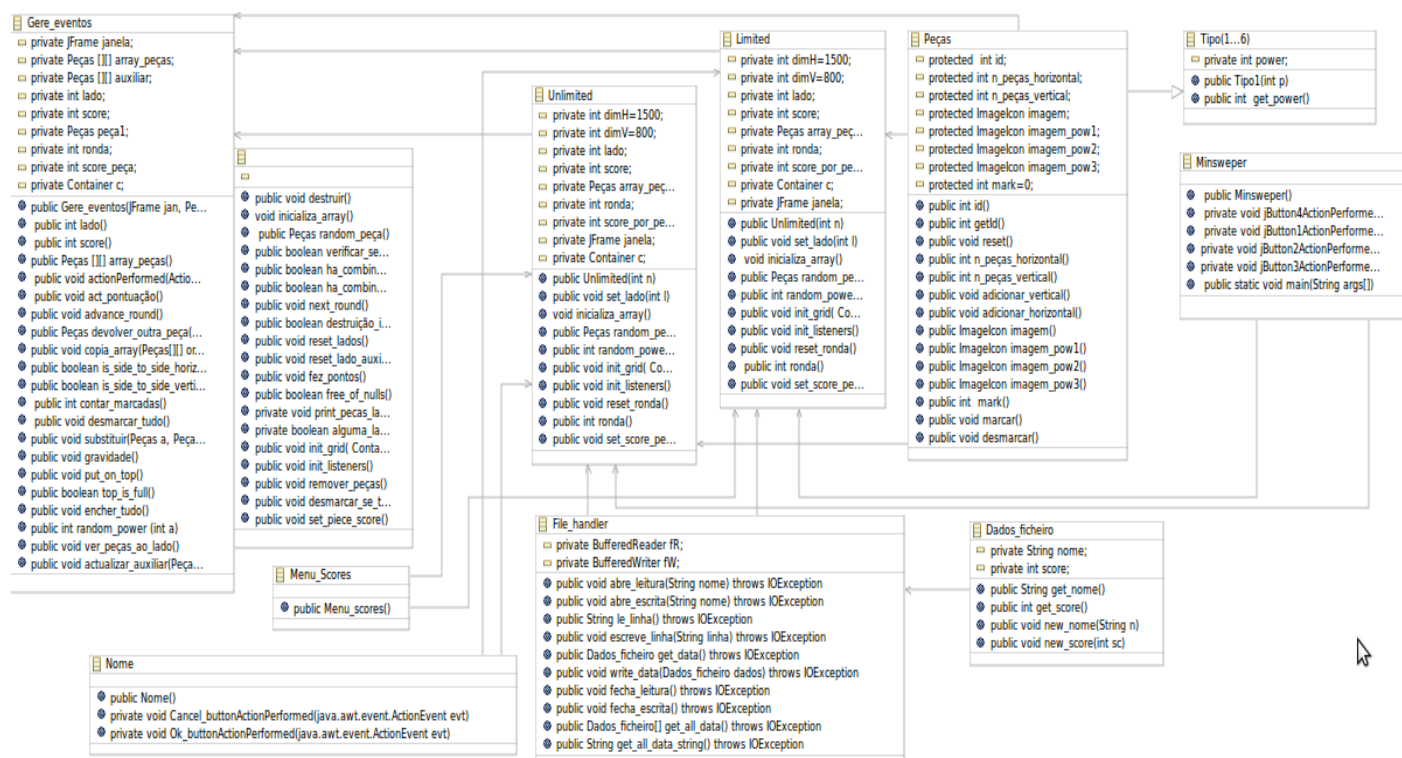
No âmbito da cadeira de Programação Orientada a Objectos foi nos proposto um projecto cujo o objectivo é a implementação de um jogo, mais propriamente a implementação do jogo Bejeweled.

Numa primeira fase foi-nos pedido que criássemos um esboço do projecto antes de começarmos a programar o jogo propriamente dito, através de um Diagrama de classes. Agora já com o projecto concluído fizemos algumas alterações no código e no diagrama, pois a parte da interface gráfica na primeira meta ainda não estava bem consolidada.

Diagrama de Classes

Para a criação do nosso diagrama de classes utilizámos uma versão do Eclipse que já contem o UML implementado. Essa versão é Eclipse Galileo SR2 Packages (v 3.5.2) disponível no site: <http://www.eclipse.org/>.

Desta vez, esta etapa só foi feita no final do projecto e o resultado final foi o seguinte:



Programa

O nosso projecto contém as seguintes classes:

Minsweper – Esta é a classe principal. Este nome pode gerar alguma estranheza, pois não está de acordo com o projecto pedido, o que prova que nunca fomos muito fãs deste jogo.

A sua função é servir de interface aos dois modos de jogo, à consulta dos de Scores e à opção de sair do jogo.

Nome – Classe que é apenas implementada no final de cada jogo e serve de interface à colocação do nome do jogador que será guardado num ficheiro juntamente com a pontuação.

Gere_eventos – Classe que implementa o ActionPerformed do jogo sem tempo, onde está a maior parte do código.

Gere_eventos_2 – Classe que faz o mesmo que o Gere_eventos mas esta contém o tempo para a realização do jogo com tempo.

Limited – Classe que implementa o jogo com tempo.

Unlimited – Classe que implementa o jogo sem tempo.

Peças – Classe que estende JButton que implementa a geração de uma peça para colocar na grelha de jogo.

Tipo(1...6) – Classes que estendem a classe de peças e especificam cada tipo de peças, ou seja, a sua cor/forma e power-ups.

File_handler – Classe que tem todas as funções que servem de auxílio a classe de ficheiros, como guardar e ler de um ficheiro.

Dados_ficheiro – Define a estrutura de dados que contem o jogador e a sua respectiva pontuação.

Menu_scores – Classe que mostra graficamente todos os Scores.

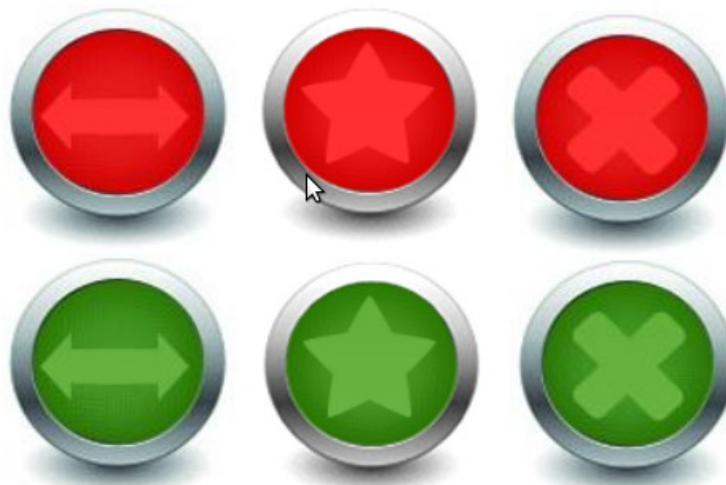
Extras

Para melhorar o nosso projecto, como foi referido no ultimo relatorio, enriquecemos a interface com um pouco de originalidade utilizando imagens em vez de criar figuras geométricas.

Para as imagens tivemos a ideia de criar algo que tivesse a haver com a nossa área, por isso pensámos em criá-las com os logótipos de 6 browsers como mostram as imagens seguintes:



Para os power-ups pensámos em criá-los com os botões dos browsers em que cada cor simboliza um power-up de um determinado logótipo.



Para enriquecer mais o trabalho tivemos ainda a ideia de colocar sons, mas tal não foi possível implementar.

Dificuldades

As dificuldades que encontramos nesta fase foi essencialmente na parte da interface gráfica, uma vez que é conceito novo e ao qual ainda não estamos muito confortados. Para combater esta dificuldade, tivemos o apoio do NeatBeans, um programa para a interface gráfica, que apesar de tudo, não facilitou de todo o trabalho, apenas serviu para isso mesmo, apoio a interface gráfica no menu inicial.

Como já tínhamos previsto na meta anterior, as maiores dificuldades que iríamos provavelmente encontrar seria na parte da interface gráfica. Tivemos muitos problemas ligados a esta área e tal causou desperdício de muito tempo fazendo com que o projecto não tenha sido concluído da forma como nós o pretendíamos.

Apesar de termos algum código pronto a ser usado, não está correctamente implementado devido a erros que nos impediram de implementar esse mesmo código.