Programação Orientada por Objetos

Primeiro trabalho prático Semestre de Verão 2017/2018

Objetivos: Neste trabalho pretende-se que os alunos adquiram prática de utilização de herança e polimorfismo usando a abordagem MVC na implementação de um jogo em consola de texto.

Entrega: Cada grupo entregará no *site* da sua turma, até dia 16 de abril, os ficheiros fonte (java) do projeto realizado e o documento que descreve a implementação do trabalho contendo o diagrama de classes.

Enunciado: Implemente o jogo designado por *PowerPlant* que permite realizar as ligações que fazem a distribuição de energia por todas as casas (representadas por H, ou por H quando têm energia) desde uma fonte (representada por P) num mapa planar. As linhas de energia são representadas por traços verticais ou horizontais, tendo fundo amarelo quando têm energia.

A figura 1 apresenta o aspeto de um estado possível deste jogo. O mapa é composto por células, de 3x3 carateres, que são rodadas 90 graus para a direita por cada *click* do jogador.

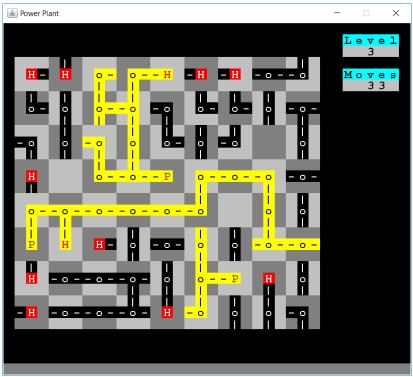


Figura 1: Aspeto do jogo

A tabela seguinte apresenta os 6 tipos de células.

Nome:	Fonte	Casa	Linha	Curva	Ramo	Espaço
Aspeto:	P -	– H	- 0 -	0 —	- o -	
Letra:	Р	Н	-	С	Т	

Os mapas são descritos no ficheiro de texto (Levels.txt) em que a primeira linha de cada mapa indica o número do nível e as dimensões do mapa (linhas x colunas), e as restantes descrevem o mapa indicando a letra de cada célula. A figura 2 apresenta o troço do ficheiro que descreve o mapa da figura 1.

#3 8 x 9												
Н	Н	С	С	Н	Η	Η	-	С				
С	-	Τ	Т	C	C	C	-	T				
С	-	С	-	С	-	С		-				
Н		C	Т	Р	C	-	С	-				
C	T	-	-	-	C		-	-				
Р	Н	Н	-	-	-	-	T	-				
Н	-	-	Т	-	T	Р	Н	-				
Н	-	-	Т	Н	С	-	-	С				
Figura 2: Troço de levels.txt												

O programa deve construir o mapa a partir da descrição do ficheiro, rodar cada célula um número aleatório de vezes, mostrar o mapa na janela e rodar cada célula por cada *click* do utilizador até que todas as casas tenham energia.

Pode ser experimentada uma versão do jogo pretendido em "PowerPlant.zip".

Este zip contém um jar com o programa, o ficheiro de descrição dos níveis, o jar da biblioteca ConsolePG e os ficheiros fonte de algumas das classes já implementadas, totalmente ou parcialmente. Destas classes também é possível depreender algumas das classes que são necessárias definir e alguns dos métodos públicos.

A figura 3 mostra as classes principais do programa, apresentando em fundo azul as totalmente implementadas e em fundo vermelho as parcialmente implementadas. Em fundo amarelo estão representadas as hierarquias de classes que devem ser definidas para cada tipo de célula no modelo e na visualização.

Implemente o jogo descrito seguindo a abordagem MVC.

- 1. Defina uma hierarquia de classes que represente os vários tipos de células (classe Cell e suas derivadas) e a classe principal do modelo (classe Plant) que implementa a lógica do jogo (as regras). Cada instância de Plant representa o estado corrente de um nível do jogo e não devem resolver aspetos de visualização. O código destas classes será reutilizado na implementação deste jogo em Android sem necessidade de alterações.
- 2. A componente de **controlo** que interage com o utilizador (classe *Game*), processa as teclas e os eventos do rato. O **controlo** carrega cada nível, usando a classe *Loader*, recebe notificações e chama métodos do **modelo** para evoluir no estado do jogo e métodos da componente de **visualização** para redesenhar as células.
- 3. A **visualização** usa a classe *ConsoLe*, da biblioteca ConsolePG incluída no *zip*, e está já implementada, faltando apenas definir as classes derivadas de *CellTile* que implementam a apresentação de cada tipo de célula.

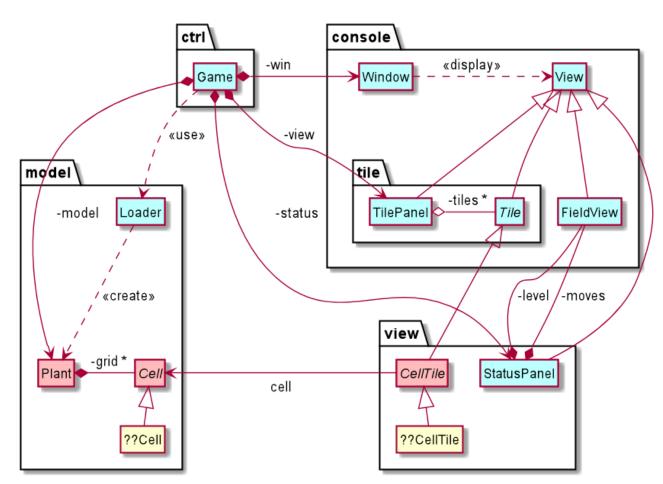


Figura 3: Diagrama de classes