Autómatos Celulares

"Coelhos e cenouras num prado verde"
Trabalho Prático de Programação II
Ano letivo de 2023/2024

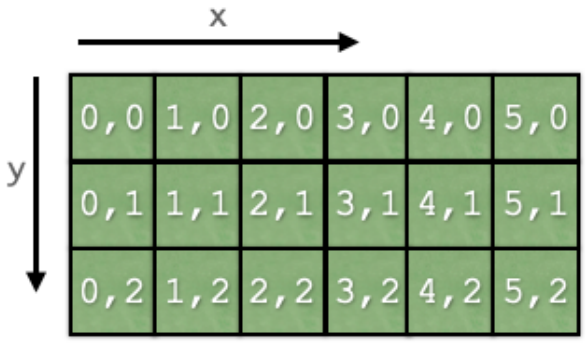
1 Objectivos

Utilização da linguagem Java e do paradigma da programação orientada a objectos para realizar a simulação duma versão do jogo da vida, esta com coelhos e cenouras que habitam um prado verde. Os coelhos alimentam-se de cenouras, mas não do prado. Conseguem sobreviver um certo tempo sem se alimentarem, findo esse tempo, com a escassez de alimento, morrem.

Este trabalho é dividido em três partes cada uma delas com a cotação que se apresenta no início de cada secção. Caso realize a parte I e a parte II deve submeter os ficheiros `Grassland.java` e `Simulation.java`. A cotação dos trabalhos corresponde à cotação das partes que submete. Neste caso serão 20 pontos. Caso decida não realizar a simulação gráfica e optar pela simulação em texto, significa que realiza a parte I e a parte III e deve submeter os ficheiros `Grassland.java` e `SimText.java`. Neste caso a pontuação máxima é 15 pontos. Caso não queira realizar nenhum tipo de simulação deve submeter unicamente o ficheiro `Grassland.java`. Neste caso o seu trabalho valerá no máximo 10 pontos.

2 Descrição

Neste projecto, irá implementar a simulação dum prado que contem coelhos e cenouras. O prado é retangular mas as extremidades são adjacentes como num toro topológico ou "donut". Tal significa que as extremidades Norte e Sul são adjacentes, bem como as extremidades Este e Oeste. O prado está dividido em células retangulares indexadas como seguidamente se apresenta. Para um prado de 6x3



Note que: a origem é o canto superior esquerdo, aumentando a coordenada em x quando se desloca para a direita, e a coordenada em y quando se move para baixo. A coordenada (0,0) pode também ser referenciada como (6,0) ou (0,3)ou (-6,3) por exemplo. Qualquer par de inteiros, lhe dará uma coordenada válida rodando em torno da arestas. Dum modo genérico as coordenadas num prado $i \times j$ são calculadas módulo i para a coordenada em x

(na horizontal) e módulo j para a coordenada em y (na vertical).

Existem duas entidades distintas neste prado: coelhos e cenouras que se reproduzem, alimentam e morrem. Cada célula neste prado pode estar ocupada por um coelho, uma cenoura ou estar vazia.

3 Parte I - Simular um prado de coelhos e cenouras(10-20 pontos)

Um prado é descrito pelo seu tamanho e pela posição inicial dos coelhos e das cenouras. É também descrito por um parâmetro:"starveTime" que especificará o número de unidades de tempo de simulação, que um coelho sobrevive sem ser alimentado.

A simulação desenrola-se em unidades de tempo ("timesteps"). Uma unidade de tempo é uma transição dum prado para outro. Não confunda prados com unidades de tempo, cada unidade de tempo começa com um prado e termina com um outro, novinho em folha.

3.1 As regras

As regras que determinam o novo prado após uma unidade de tempo, dependem única e exclusivamente dos habitantes que ocupam cada uma das células do prado no início da unidade de tempo. O conteúdo duma célula no fim duma unidade de tempo, é determinada pelo conteúdo das células dos seus 8 vizinhos, no início da unidade de tempo. Os vizinhos duma célula são as células adjacentes a norte, sul, este e oeste e também as quatro diagonais. Na figura as células de cor diferente constituem os 8 vizinhos de célula (1,1).

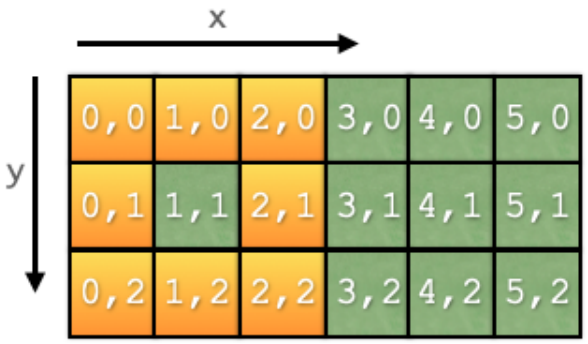


Figure 1: Vizinhos da célula (1,1)

Eis as regras:

1. Se uma célula contem um *coelho* e algum dos seus vizinhos é uma cenoura, então o coelho come a cenoura durante o período de tempo, e mantem-se na célula no final do período de tempo.(Pode acontecer que vários coelhos disputem a mesma cenoura, não há problema, as cenouras chegam para todos).

2. Se uma célula contem um *coelho*, e nenhum dos seu vizinhos é um cenoura, ele passará fome durante este período de tempo. Se se trata do nésimo +1 periodo de tempo de privação ("starveTime"), então o coelho morre (desaparece). Caso contrário manter-se-á na célula.

3. Se uma célula contem uma *cenoura*, e todos os seus vizinhos são cenouras, ou células vazias, então no fim da unidade de tempo a cenoura manter-se-á na sua célula.

4. Se uma célula contem um *cenoura*, e um dos seus vizinhos é um coelho, então a cenoura será comida pelo coelho e desaparecerá.

5. Se uma célula contem uma *cenoura*, e dois ou mais dos seus vizinhos são coelhos, então um coelho recém-nascido ocupará aquela célula, no fim da unidade de tempo. Os coelhos quando nascem estão bem alimentados, tal significando que aguentam "starveTime" undades de tempo sem se alimentarem, mas tal como os outros coelhos, morrem após "starveTime" +1 unidades de tempo sem se alimentar.

6. Se uma célula está *vazia* e menos de dois vizinhos são cenouras, então manter-se-á vazia

7. Se uma célula está *vazia*, e pelo menos dois dos seus vizinhos são cenouras e no máximo um dos vizinhos é um coelho, então uma nova cenoura nascerá naquela célula

8. Se uma célula está *vazia*, e pelo menos dois dos seus vizinhos são cenouras e pelo menos dois dos seus vizinhos são coelhos, então um novo coelho nascerá naquela célula (já sabe os coelhos recém-nascidos estão bem alimentados!)

3.2 O seu trabalho

Pede-se-lhe, nesta parte do trabalho, que preencha a implementação da classe *Grassland.java* disponibilizada juntamente com o enunciado. São fornecidas na classe os métodos públicos que deve implementar e também um construtor, que recebe três inteiros correspondentes à dimensão do prado e o tempo de privação dos coelhos (starveTime) e retorna um novo prado, com as dimensões dadas. Por exemplo:

```
Grassland mead=new Grassland(i,j,k);
```

deve criar um prado de dimensões ixj. É livre, de na sua implementação, definir quaisquer váriveis, ou métodos na classe *Grassland*, e inclusivé criar outras classes que necessite. Não se esqueça que este trabalho é realizado no âmbito da disciplina de programação 2, e portanto deve classificar as entidades envolvidas usando classes. Embora nos ficheiros apareça a indicação de que não pode alterar as assinaturas dos métodos da classe *Grassland* pode alterá-los. Eles estão lá para o guiar na sua implementação. Leia as instruções que aparecem como comentários na classe para gerar o código apropriado. O mais importante dos métodos, *timeStep()* é o método que realiza a simulação.

A sua classe *Grassland* deve representar uma prado como um ou mais arrays bi-dimensionais. Cabe-lhe a si decidir como representa cada elemento do array (se vazio, se cenoura ou coelho, e neste último caso, há quanto tempo ele foi alimentado). A sua representação interna não necessita de usar as constantes EMPTY, RABBIT e CARROT, mas se quiser pode usá-las. Em qualquer dos casos o método *cellContents()* deve retornar estas constantes, porque à priori não sei como vai representar os habitantes do seu prado.

4 Parte II - Visualizar a simulação usando AWT (10 pontos)

A classe `Simulation` é um programa que corre e anima uma simulação dos Coelhos e das Cenouras. Recebe 3 parâmetros, a largura e altura do prado e o tempo de privação e produz a referida animação. Algumas escolhas de parâmetros matam rapidamente o seu prado, enquanto outras permitirão que coelhos e cenouras convivam pacificamente durante muito tempo.

5 Parte III- Visualização da simulação no standard output (5 pontos)

Caso não queira ou não consiga usar o `AWT` pode visualizar o seu prado realizando uma simulação textual. Neste caso deve apresentar o ficheiro SimText.java que tal como o Simulation.java tem os mesmos 3 parâmetros que interferem com a vida no prado. Arranje caracteres para representar os habitantes e envie para o output uma representação em texto do seu prado.

6 Entrega e realização

O trabalho pode ser realizado em grupos de 2 alunos(eventualmente 3). O trabalho deverá ser entregue até ao dia 21 de Janeiro de 2024, sendo realizada a submissão pelo moodle, nos moldes habituais. Deve submeter os ficheiros correspondentes às partes do trabalho que realiza, e ainda um relatório adequado à apresentação do trabalho. Os ficheiros deverão ser "zipados" e submetidos num único ficheiro com o número dos alunos que realizam o trabalho. Os trabalhos serão apresentados na semana de 22 a 27 de Janeiro e só excepcionalmente poderão ser apresentados por zoom.