

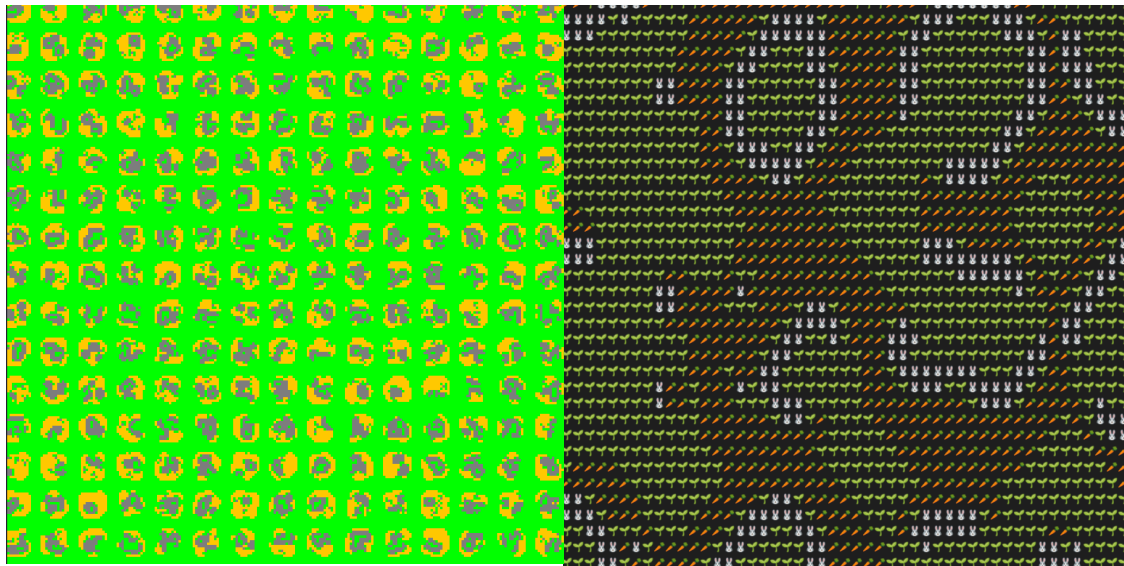


UNIVERSIDADE
DE ÉVORA

Programação 2

Licenciatura em Engenharia Informática

Ano Letivo 2023/2024



Docente: Lígia Ferreira

Trabalho realizado por: João Santos, 51966

Catarina Andrade, 55406

Introdução

Na Unidade Curricular Programação II, foi-nos proposta a realização de uma simulação chamada “Coelhos e cenouras num prado verde”, utilizando a linguagem aprendida nas aulas ao longo do semestre, linguagem Java (POO).

Esta simulação consiste num prado verde, onde habitam coelhos e cenouras. Os coelhos alimentam-se de cenouras, mas não do prado. Estes conseguem sobreviver um determinado tempo sem se alimentarem, mas, com o passar do tempo começa a existir escassez de alimento, e os mesmos acabam por morrer.

Desenvolvimento

Grassland.java

```
public Grassland(int i, int j, int starveTime){...}
```

É um Construtor que tem como objetivo inicializar a matriz e definir o tempo que um coelho pode passar sem comer.

```
public int width() {...}
```

É o método que obtém a largura do prado.

```
public int height() {...}
```

É o método que obtém a altura do prado.

```
public int starveTime() {...}
```

É o método que obtém o tempo que um coelho pode passar sem comer.

```
public int addCarrot(int x, int y) {...}
```

O método é usado para adicionar uma cenoura em uma posição específica.

```
public int addRabbit(int x, int y) {...}
```

O método é usado para adicionar um coelho em uma posição específica.

```
public int setEmpty(int x, int y) {...}
```

O método é usado para adicionar “nada” em uma posição específica.

```
public int cellContents(int x, int y) {...}
```

O método obtém o conteúdo de uma célula específica.

```
public int timeStep() {...}
```

Este método avança o prado em uma unidade de tempo, o que acaba por alterar o prado, através das regras descritas no enunciado, sempre que a unidade de tempo avança.

```
public int countNeighbors(int x, int y, int type) {...}
```

Este método é usado para saber os 8 vizinhos (como descrito no enunciado) de cada célula no prado (se é um coelho, uma cenoura ou “nada”).

SimText.java

O SimText.java é bastante parecido com o **Simulation.java** dado pela professora, muda apenas o método `private static void draw(Graphics graphics, Grassland mead){...}` para o método `private static void draw(Grassland mead) {...}`. Este em vez de percorrer todas as células e mudar a cor do pixel para cinzento (coelho), para laranja (cenoura) ou para verde (relva), altera para uma String específica.

A string “\uD83D\uDC30” representa um emoji do coelho, a string “\uD83E\uDD55” da cenoura e a string “\uD83C\uDF31” da relva no standard output.

Não é necessário o tamanho da célula nem a criação da frame visto que o Standard Output não usa nada gráfico.

Conclusão

Com este trabalho conseguimos aprofundar o nosso raciocínio lógico, bem como limar algumas arestas no nosso leque de conhecimentos no que toca à linguagem de programação Java (POO).

Tivemos algumas dificuldades no desenvolvimento deste trabalho, algumas das quais foram, por exemplo, a implementação das regras dadas pela professora. O mais difícil foi sem dúvida começar, mas como em tudo, tem que se resolver um problema de cada vez e, acabámos por ultrapassar estes obstáculos.

Em suma, ganhámos ferramentas que serão úteis futuramente no nosso percurso académico e profissional.