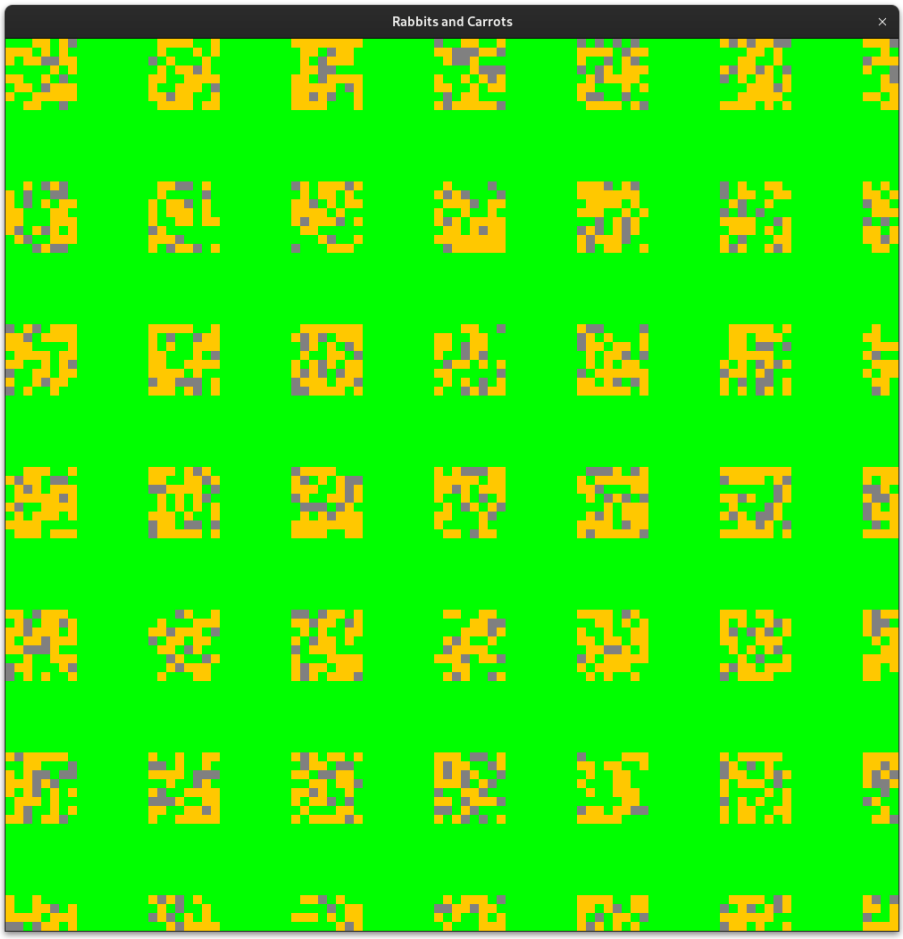
UNIVERSIDADE
DE ÉVORA

Relatório Programação de II

David Marinho
54560

Axel Amoroso Carapinha
55248



Introdução

Desenvolvimento

Principais dificuldades

Array intermediário ou Grassland como instância de classe?
Não quisemos escolher nem um nem outro pois achamos que seriam más abordagens.
Primeiramente tentamos

Conclusão

Em suma, as dificuldades foram mitigadas aplicando os resultados das regras diretamente no novo prado (Grassland)

Procedimentos

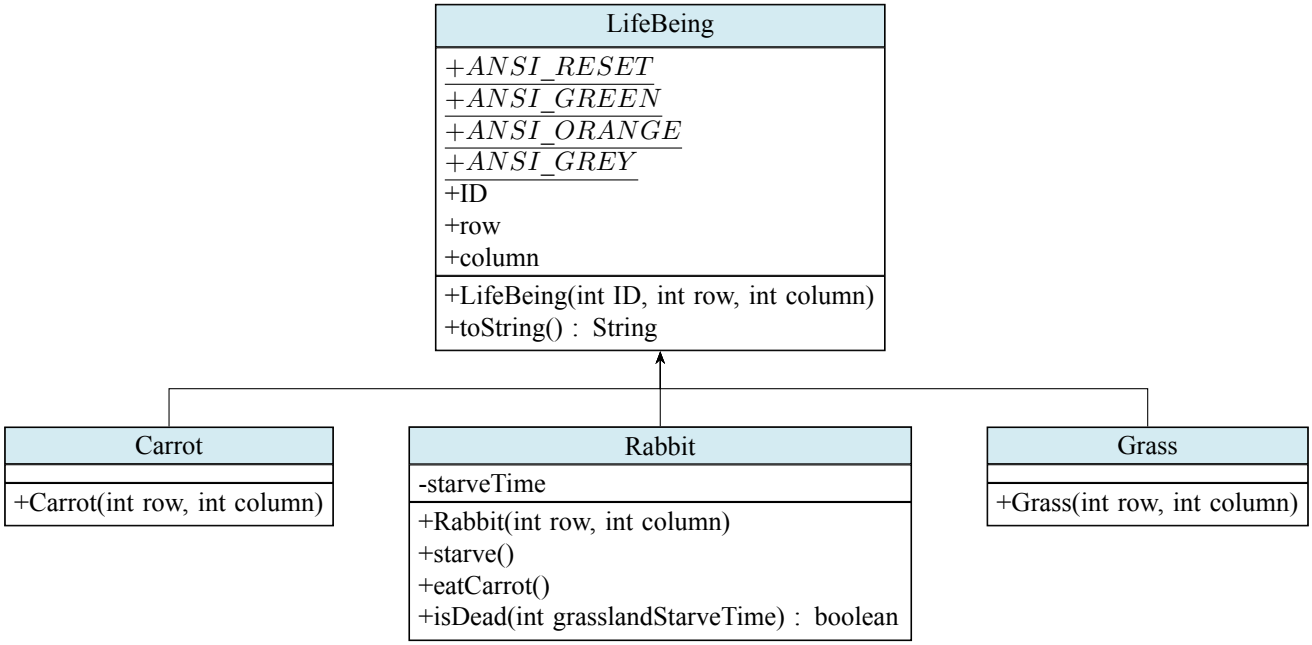
1. Na execução do programa podem ser passados 3 argumentos pela linha de comandos:
    - i → O comprimento do prado.
    - j → A largura do prado.
    - starveTime → Número de timesteps a que os coelhos sobrevivem sem comer qualquer cenoura. Por defeito, se não passados quaisquer argumentos, temos i = 100, o j = 100 e starveTime = 5.
2. Um prado de comprimento i x j é preenchido com objetos do tipo 'Grass' e são gerados aleatoriamente outros objetos, objetos de tipo 'Rabbit' e 'Carrot'.
3. A cada timestep é gerado um novo prado respeitando as regras a que a geração anterior esteve sujeita.

Design Patterns

Recorremos aos três pilares da Programação Orientada a Objetos, encapsulamento herança e polimorfismo, e criamos quatro classes para representar entidades específicas:

- LifeBeing → Classe mãe e abstrata das classes 'Rabbit', 'Carrot' e 'Grass'.
- Rabbit → Classe filha que herda 'LifeBeing'. Tem como função organizar todos os dados a respeito dos coelhos.
- Carrot → Classe filha que herda 'LifeBeing'. Organiza toda a estrutura de dados a respeito das cenouras.

- Grass → Classe filha que herda 'LifeBeing' e organiza toda a estrutura de dados a respeitos das ervas.
- Grassland → Classe responsável por organizar todos os dados a respeito do campo, marioritariamente a posição dos objetos no campo e o tempo de longevidade dos mesmos.
- Simulation → Classe principal do programa. Responsável pela renderização e execução do programa.



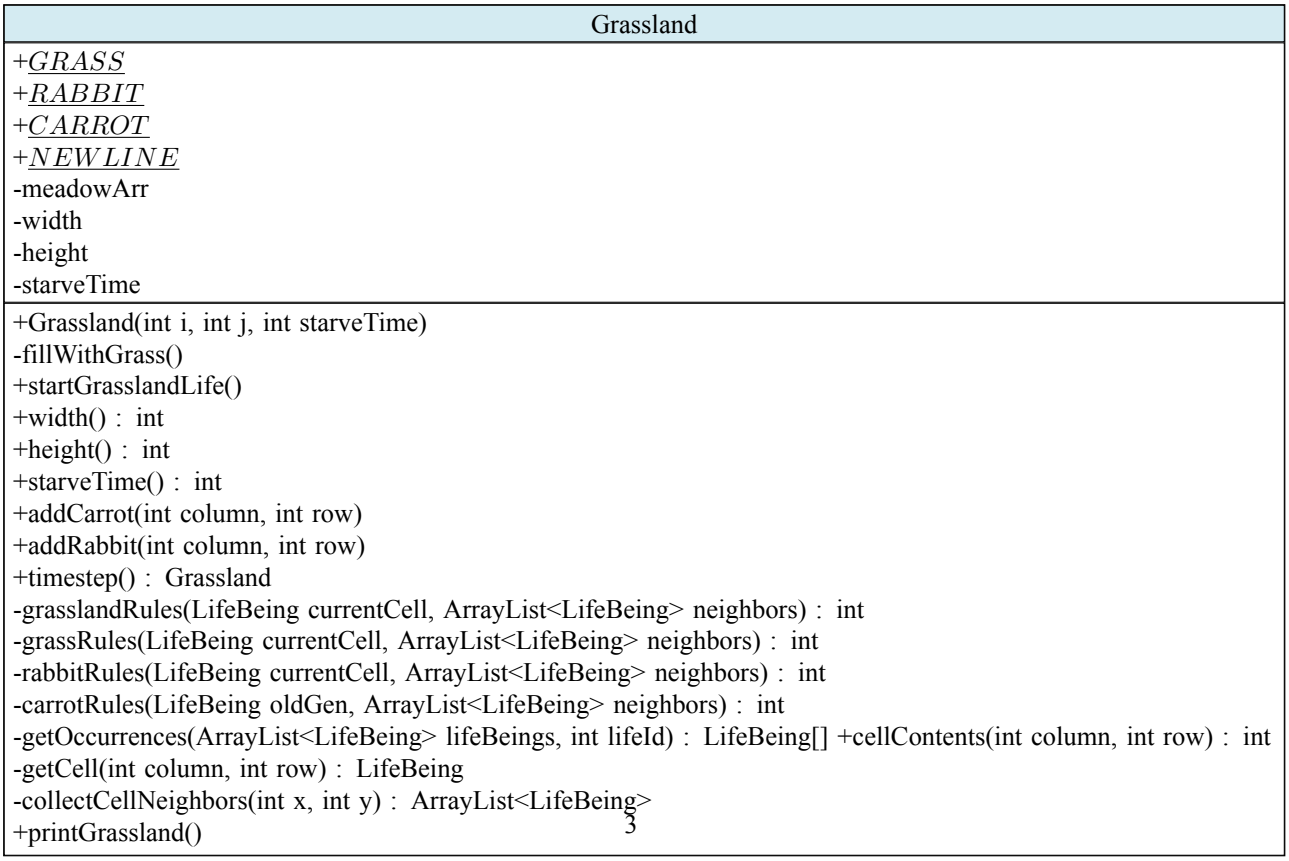


Figure 1: Diagrama de classes