Logo

Description automatically generated

Instituto Superior de Engenharia de Lisboa

Licenciatura em Engenharia Informática e Multimédia

Ano Letivo 2021/2022

Modelação e Simulação de Sistemas Naturais

Trabalho Pratico 1

Henrique Pereira 48571

Docente: Arnaldo Abrantes

Novembro 2021

Índice

**No table of contents entries found.**

1. Jogo da Vida

Implementei o jogo da vida clássico (23/3) na classe JogoDaVida que herda a classe CellularAutomata desenvolvida na aula e ainda. Implementei a classe CellJDV que herda a classe Cell desenvolvida na aula. Implementei a classe RunJDV que implementa a classe IProcessingApp desenvolvida na aula.

Ainda completei os seguintes pontos facultativos:

* Implementar variantes (variante 23/2 na classe JDV23por2 que herda a classe JogoDaVida)
* Atribuir múltiplas cores às células de acordo com um dado critério
* Escolher a configuração inicial do autómato através de GUI e/ou leitura de ficheiro (apenas através da GUI)
  1. CellJDV
     1. Atributos

O atributo static deadColor armazena a cor das células mortas.

O atributo aliveColor armazena a cor da célula quando esta fica viva e é inicializado com uma cor aleatória.

* + 1. Métodos

O método getAliveNeigh() devolve um array de CellJDV com todas as células vizinhas que estão vivas, excluindo a célula que chamou o método do array retornado.

O métdodo getCommonNeighborColor() devolve o valor da cor aliveColor mais comum entre os vizinhos vivos da célula que chama o método. Caso a célula não tenha vizinhos vivos então devolve o valor da cor aliveColor da própria célula.

* 1. JogoDaVida
     1. Atributos

O atributo aliveCellsPercentage armazena o ratio de celulas vivas num valor entre 0 e 1.

* + 1. Métodos

O método createCells(Cell[][]) dá overwrite ao método na classe CellularAutomata e é bastante idêntico ao mesmo mas cria células da classe CellJDV.

O método initRandom() dá overwrite ao método na classe CellularAutomata e é bastante idêntico ao mesmo mas coloca o estado da celula como viva aleatoriamente dependendo do valor do atributo aliveCellsPercentage.

O método update() cria um novo array de células onde guardará as células no seu estado seguinte. Para cada célula cálcula o seu próximo estado tendo em conta o seu número de vizinhos vivos. No final do processo atualiza o atributo cells com o novo array das células no seu estado seguinte.

* 1. JDV23por2
     1. Métodos

O método update() dá overwrite ao método na classe JogoDaVida e é bastante idêntico ao mesmo, apenas as regras de mudança de estado variam.

* 1. RunJDV
     1. Atributos

O atributo running “ativa” e “desativa” a evolução do JogoDaVida.

O atributo aliveCellsPercentage define o ratio de celulas vivas num valor entre 0 e 1.

Os atributos nCols e nRows definem o tamanho do array de células.

* + 1. Métodos

O método setup(PApplet) cria um objeto da classe JogoDaVida ou da classe JDV23por2 dependendo de qual linha de código esteja descomentada. Após criar o objeto chama o método initRandom() do objeto criado e mostra o estado do jogo.

O método draw(PApplet) atualiza o estado do jogo para o seguinte apenas se o atributo running estiver a true, e mostra o estado do jogo.

O método mousePressed(PApplet) muda o estado da célula clicada para o estado contrário ao que se encontra.

O método keyPressed(PApplet) “ativa” ou “desativa” a evolução do jogo clicando na tecla de espaço, e evolui para o próximo estado do jogo clicando na tecla N.