



**Instituto Superior
de Engenharia**

Politécnico de Coimbra

Programação Avançada 2023-2024

JavaLife

Trabalho Realizado:

Rodrigo Reis , a2022137090

Dlogo Ribeiro , a2022136604

Chelsea Duarte , a 2021100010

Índice	2
Introdução	2
Decisões de Implementação	3
Diagrama da Máquina de Estados Finitos (FSM)	4
Diagramas de Outros Padrões Aplicados	5
Descrição das Classes	5
Ecosystem Manager	5
Ecosystem	5
IMemento,CareTaker,IMementoOriginator	5
FaunaStateMachineTest	5
Fauna	5
Flora	5
Inanimate	6
CommandManager	6
CommandAdapter	6
FaunaState	6
FaunaStateAdapter	6
FaunaStateContext	6
IFaunaState	6
AttackingState,EatingState,MovingState,ReproduceState,SearchFoodState	6
ElementsFactory	6
MainScene,CreateEcosystemScene,StartScene,SplashScreen	7
Relacionamento entre Classes	7
Estado de Implementação das Funcionalidades	8
Ficheiro	8
Eventos	8
Ecossistema	9
Simulação	9
Aplicação	10
StartScene	10
Create Ecosystem	11
MainScene	12

Introdução

Este trabalho foi desenvolvido no âmbito de um trabalho para a disciplina de Programação Avançada, do 2º ano do curso de Engenharia Informática do ISEC. Neste trabalho foi desenvolvida uma aplicação em java com o objetivo de simular a evolução de um ecossistema.

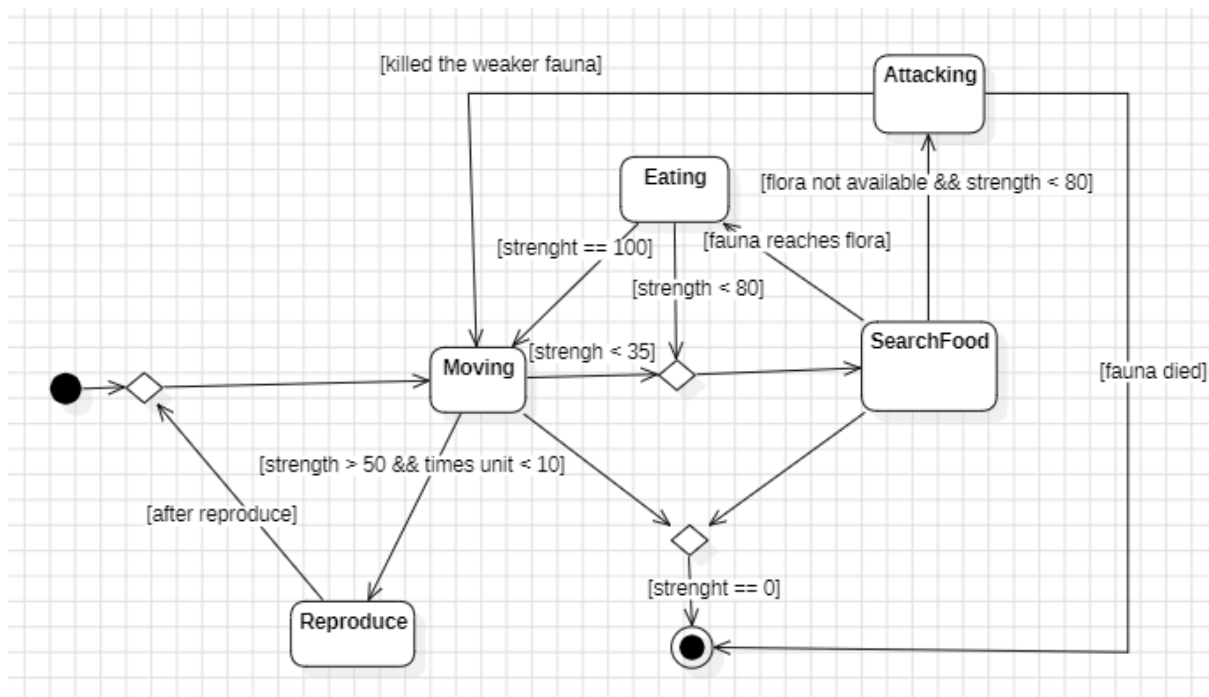
Decisões de Implementação

Considerando os diversos estados da fauna, optamos por implementar uma máquina de estados, onde cada instância de fauna possui a sua própria máquina de estados encapsulada.

Além disso, foi efetuada a criação de uma classe chamada "EcosystemManager" para servir como fachada, integrando componentes como o GameEngine e CareTaker com o Ecosystem.

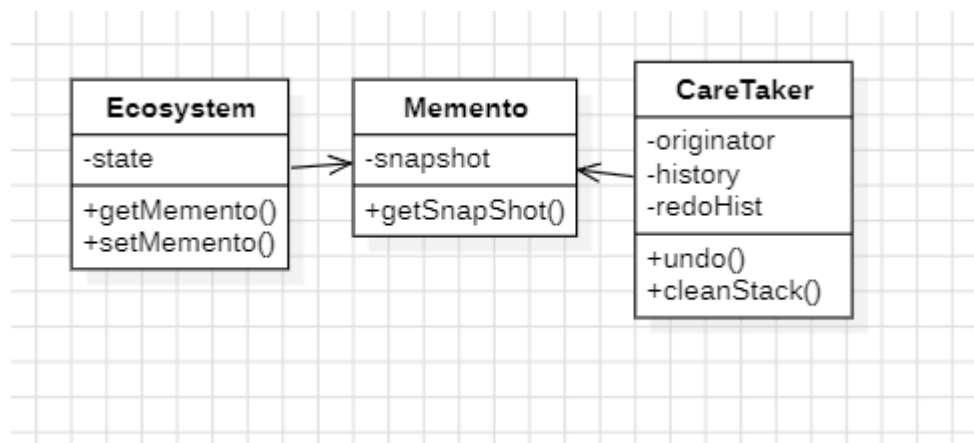
Na GUI, optamos por dividir em scenes, onde cada cena possui widgets personalizados na package `pt.isec.pa.javalife.ui.gui.components`. Cada scene recebe o `primaryStage` e a classe de fachada como parâmetros.

Diagrama da Máquina de Estados Finitos (FSM)



Estado	
MOVING	A fauna está em movimento pelo ecossistema. Durante este estado, pode mudar de direção aleatoriamente e decide se deve se reproduzir ou procurar comida, dependendo da sua força.
SEARCH_FOOD	A fauna procura por comida no ecossistema. Se encontrar flora próxima, muda para o estado de alimentação. Se não, verifica se existe fauna fraca nas proximidades para atacar.
EATING	Durante este estado, a fauna está se alimentando de flora próxima. Se a flora não estiver disponível ou estiver muito distante, muda para o estado de procura de comida.
ATTACKING	A fauna está atacando outra fauna no ecossistema. A força da fauna determina o resultado do ataque, com a possibilidade de ambas as faunas morrerem ou uma sobreviver e a outra morrer
REPRODUCE	Este estado é acionado quando a fauna decide se reproduzir. A reprodução só ocorre se a fauna mais forte estiver próxima. Após 10 unidades de tempo a reprodução é completada.

Diagramas de Outros Padrões Aplicados



Na implementação do padrão memento, a classe Caretaker é responsável por armazenar o conjunto de estados guardados do jogo, que são provenientes da classe Memento. A classe Memento apenas guarda um objeto genérico dentro do contexto do programa.

Na classe "Ecosystem", é guardado um array dos elementos e algumas configurações.

Descrição das Classes

Ecosystem Manager

Esta classe gerencia as operações e o estado de um ecossistema, utilizando comandos para modificar elementos e propriedades, além de suportar funcionalidades de desfazer/refazer e notificações de mudança de estado. Ela atua como uma fachada, coordenando as interações entre GameEngine, Ecosystem e outros componentes.

Ecosystem

A classe Ecosystem armazena parâmetros de configuração e instâncias dos elementos, permitindo a evolução, adição e remoção desses elementos, além de lidar com as colisões no ecossistema.

IMemento, CareTaker, IMementoOriginator

Esta classe serve para tirar e armazenar as informações do snapshot, para não ter de guardar num ficheiro. O caretaker serve para guardar vários momentos que depois podem ser restituídos caso chamemos uma função.

FaunaStateMachineTest

A classe FaunaStateMachineTest contém testes unitários e esses testes abordam o movimento da fauna, sua morte, busca por comida, alimentação, ataque a outras entidades e reprodução.

Fauna

Uma classe responsável por representar e gerenciar as entidades de fauna no ecossistema, controlando seu estado, movimento e interações com outros elementos. Essas entidades possuem características como força, velocidade e direção, influenciando sua movimentação e comportamento no ambiente.

Flora

Uma classe que representa elementos de flora no ecossistema, controlando sua força e capacidade de reprodução. Esses elementos têm a capacidade de evoluir ao longo do tempo, podendo reproduzir-se em áreas adjacentes quando atingem determinado nível de energia.

Inanimate

Esta classe representa elementos inanimados no ecossistema.

CommandManager

O CommandManager é responsável por gerir a execução, anulação e reintegração de comandos, utiliza duas pilhas para armazenar os comandos executados e os comandos a reintegrar.

CommandAdapter

Classe abstrata CommandAdapter é implementada pelos outros comandos e tem como objetivo de obter o receiver para gerir as ações do comando.

FaunaState

Enumeração que define os diferentes estados em que uma fauna pode estar no ecossistema.

FaunaStateAdapter

Esta classe serve como uma estrutura base para os diferentes estados de uma fauna no ecossistema, facilitando a transição entre estados e a execução de ações específicas. É herdada por classes como MovingState, SearchFoodState e ReproduceState.

FaunaStateContext

Esta classe gerencia o estado atual de uma fauna dentro do ecossistema, permitindo a transição entre diferentes estados e a execução de ações correspondentes.

IFaunaState

Uma fábrica de estados para entidades de fauna

AttackingState, EatingState, MovingState, ReproduceState, SearchFoodState

Estas classes representam cada estado da máquina de estados da Fauna

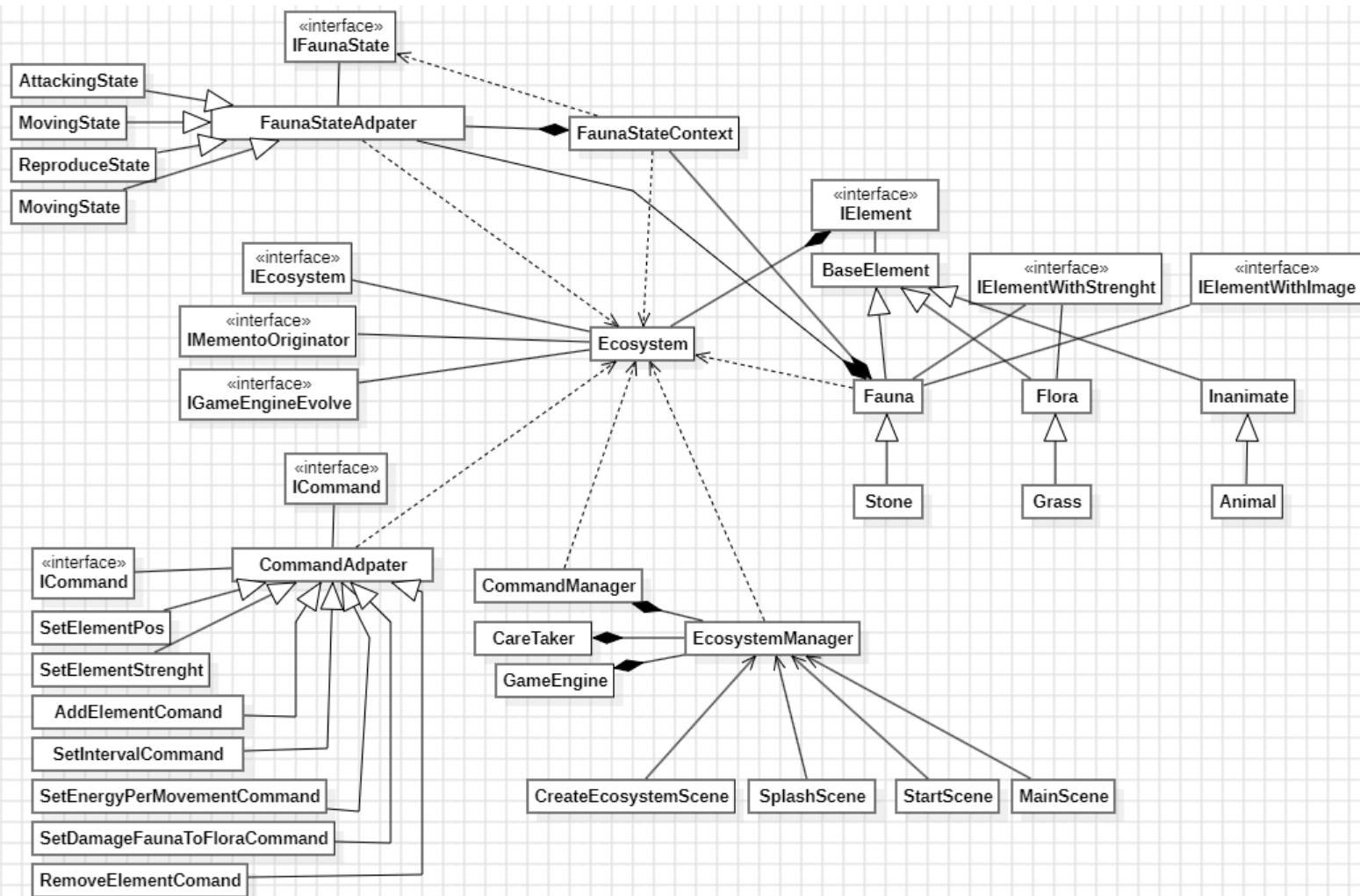
ElementsFactory

Uma fábrica de elementos para o ecossistema, criando entidades de fauna, flora e objetos inanimados com base nos parâmetros fornecidos.

MainScene, CreateEcosystemScene, StartScene, SplashScreen

Estas classes representam diferentes interfaces da simulação.

Relacionamento entre Classes



Estado de Implementação das Funcionalidades

Ficheiro

Funcionalidade:	Estado da implementação:
Criar – permite criar uma nova simulação, entrando no modo de configuração;	Completo
Abrir – permite abrir e continuar uma simulação previamente gravada;	Completo
Gravar – permite gravar o estado atual da simulação, indicando o nome do ficheiro;	Completo
Exportar – exportar para ficheiro de texto (formato csv) os atuais elementos da simulação, incluindo informação sobre o tipo, força e posição x e y;	Completo
Importar – importar os elementos guardados num ficheiro de texto (formato csv). Os elementos importados que se sobrepõem (posição) aos existentes na simulação em curso deverão ser ignorados, assim como aqueles cuja posição fique fora da área da simulação;	Completo
Sair – sair da simulação, perguntando se pretende gravar caso não o tenha feito anteriormente.	Completo

Eventos

Funcionalidade:	Estado da implementação:
Aplicar Sol – durante 10 unidades de tempo a flora ganha força ao dobro da velocidade e a fauna desloca-se a metade da velocidade;	Completo
Aplicar herbicida – aplica-se apenas a elementos do tipo flora, matando o elemento selecionado;	Completo
Injetar força – aplica-se apenas a elementos do tipo fauna, permitindo aumentar a força do elemento selecionado em 50 unidades.	Completo

Ecossistema

Funcionalidade:	Estado da implementação:
Configurações gerais do ecossistema – definição de valores alternativos para os valores por omissão referidos no enunciado, bem como a dimensão do ecossistema. A dimensão do ecossistema só pode ser alterada antes da simulação ser iniciada;	Completo
Adicionar elemento inanimado;	Completo
Adicionar elemento flora;	Completo
Adicionar elemento fauna;	Completo
Editar elemento – permite alterar os parâmetros do elemento selecionado;	Completo
Eliminar elemento – ter em atenção que os elementos correspondentes à cerca não poderão ser eliminados (a cerca deve ser criada automaticamente quando uma nova simulação é criada e, por isso, não devem ser eliminados);	Completo
Undo/Redo (de todas as operações no contexto da configuração do ecossistema).	Completo

Simulação

Funcionalidade:	Estado da implementação:
Configuração da simulação – por exemplo, para definir a unidade de tempo e as dimensões da área visual de simulação;	Completo
Executar/Parar;	Completo
Pausar/Continuar;	Completo
Gravar snapshot – permite registar o momento atual da simulação (esta informação nunca é gravada para ficheiro, nem quando se ordena a gravação do estado atual da simulação para ficheiro);	Completo
Restaurar snapshot – permite retomar a simulação num ponto salvaguardado.	Completo

Aplicação

StartScene



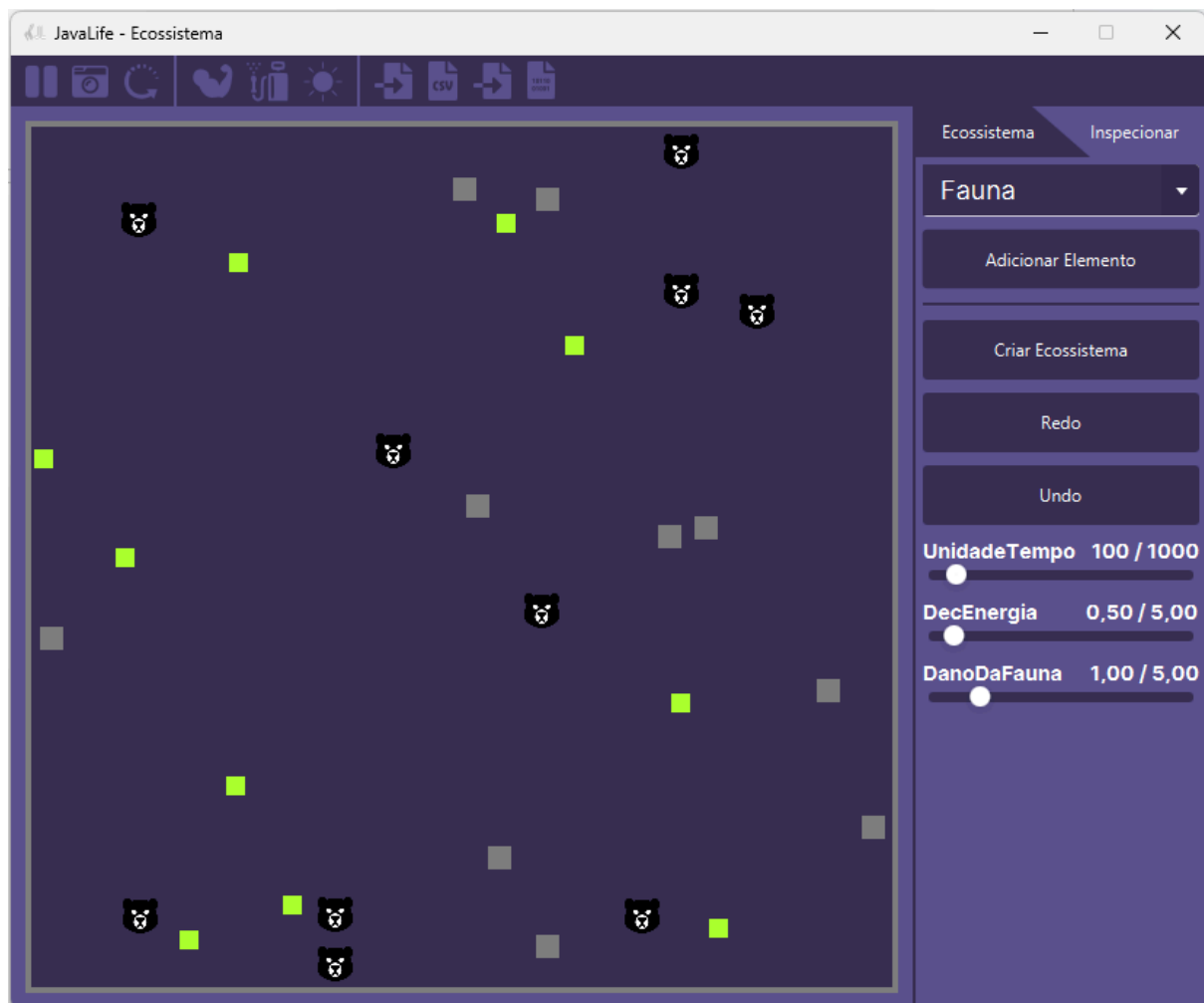
Aqui podemos ver a “StartScene” onde podemos escolher entre criar um novo ecossistema ou podemos importar um ecossistema que foi guardado num ficheiro de texto “.csv”.

Create Ecosystem



Nesta parte do programa podemos ver a "CreateEcosystemScene" aqui podemos escolher o tamanho do jogo, as quantidades da Fauna, Flora e inanimados(pedras), podemos também mudar a unidade de tempo, podemos alterar também o dano dado pela fauna e a sua energia em cada movimento da fauna, por último podemos escolher o ícone que desejamos para a fauna.

MainScene



Main scene onde decorre a simulação do ecossistema onde podemos configurar variáveis do ecossistema , adicionar,remover,editar elementos entre outras opções como snapshot,salvar,eventos...