Instituto Superior de Engenharia de Lisboa

Modelação e Simulação de Sistemas Naturais

Relatório do Trabalho Final

A Sustentabilidade de Recursos Renováveis

Shun Wang nº45410 - João Cunha nº45412 - Arman Freitas nº45414 Grupo 02

Docentes:

Arnaldo Abrantes

Paulo Vieira

7 de Janeiro de 2019

Conteúdo

Lista de Figuras

1 Introdução

O seguinte trabalho tem como alvo de estudo a sustentabilidade do ambiente, tal como sabemos os recursos que chamamos de renováveis conseguem ser repostos até a um certo ritmo, se este ritmo for ultrapassado os recursos começam a escassear e podem desaparecer por completo do nosso planeta.

Este ritmo pode se aplicar tanto a recursos como aos animais existentes num ecossistema, quando existe muita caça a algum tipo de animal e este não tem habilidade de se reproduzir à mesma velocidade que são caçado começam a ficar em vias de extinção e se mesmo assim não forem impostas leis para a proteção do animal corremos o risco de perder para sempre uma espécie de animal.

A primeira parte do é pedido trabalho é criar no Anylogic um sistema de *stocks e flows* que simule um ecossistema sustentável entre pesca e a quantidade de peixe existente no mar, obtendo o máximo lucro possível.

Na segunda parte do trabalho foi pedido para fazer um ecossistema no *Processing* baseado em matéria lecionada das aulas, *boids* e células, em que os *boids* representam os animais e as células o terreno onde os animais vivem. O objetivo seria criar um sistema sustentável de presas e predadores em que nenhuma espécie se extingue.

2 Simulador de Ecossistema usando o AnyLogic

Com o objetivo de otimizar o projeto em questão, utilizando as várias regras do formulário, primeiramente foi alterado a quantidade de produção dos barcos, desta forma, apenas são encomendados 20 barcos novos.

Figura 1: Encomenda de Barcos

Tendo em conta que os barcos não são destruídos ou perdidos em alto mar, não é preciso a reposição dos mesmos. Este exercício foi realizado de modo a obter o máximo lucro possível, por isso algumas variáveis foram evitadas.

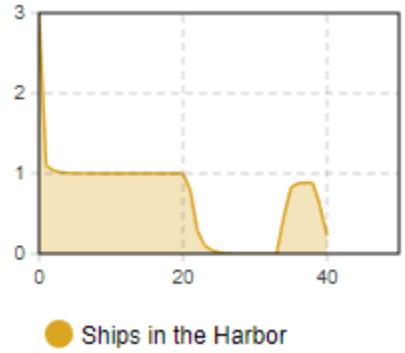

Figura 2: Barcos no hangar

Como podemos ver na imagem, inicialmente estão 3 barcos no hangar pois esse é o número inicial de barcos. Durante os primeiros 20 anos, o *harbor* tem sempre um barco que simboliza a sua construção. Após um ano, o barco está finalizado e pronto para ir para o alto mar, mas como outro barco é encomendado, assim que um sai, outro começa logo a ser construido. Por isso durante os primeiros 20 anos, 19 barcos são construidos, e 22 barcos vão para alto mar.

Observa-se na imagem a quantidade de barcos em alto mar ao longo do tempo:

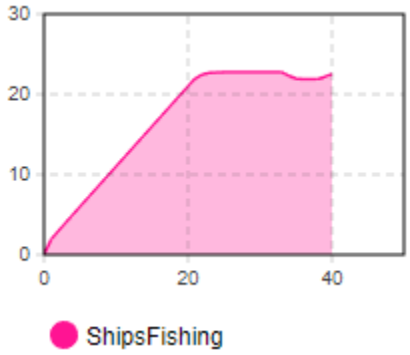

Figura 3: Barcos a Pescar

Reparando que quando se aproxima do fim, apenas 21 barcos vão para alto mar, pois os peixes começam a ser poucos para a quantidade de barcos, por isso um dos barcos fica no hangar de maneira a otimizar as despesas e os ganhos, e após vários anos, o barco no hangar volta a ir para alto mar pois a quantidade de peixe disponível aumenta.

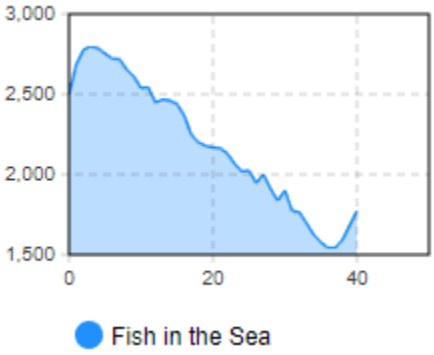

Figura 4: Peixes no mar

Podemos ver agora o desenvolvimento do exercício, de salientar que a produção de barcos novos está diretamente dependente da quantidade de peixe no mar e que apenas 1 barco é encomendado por ano.

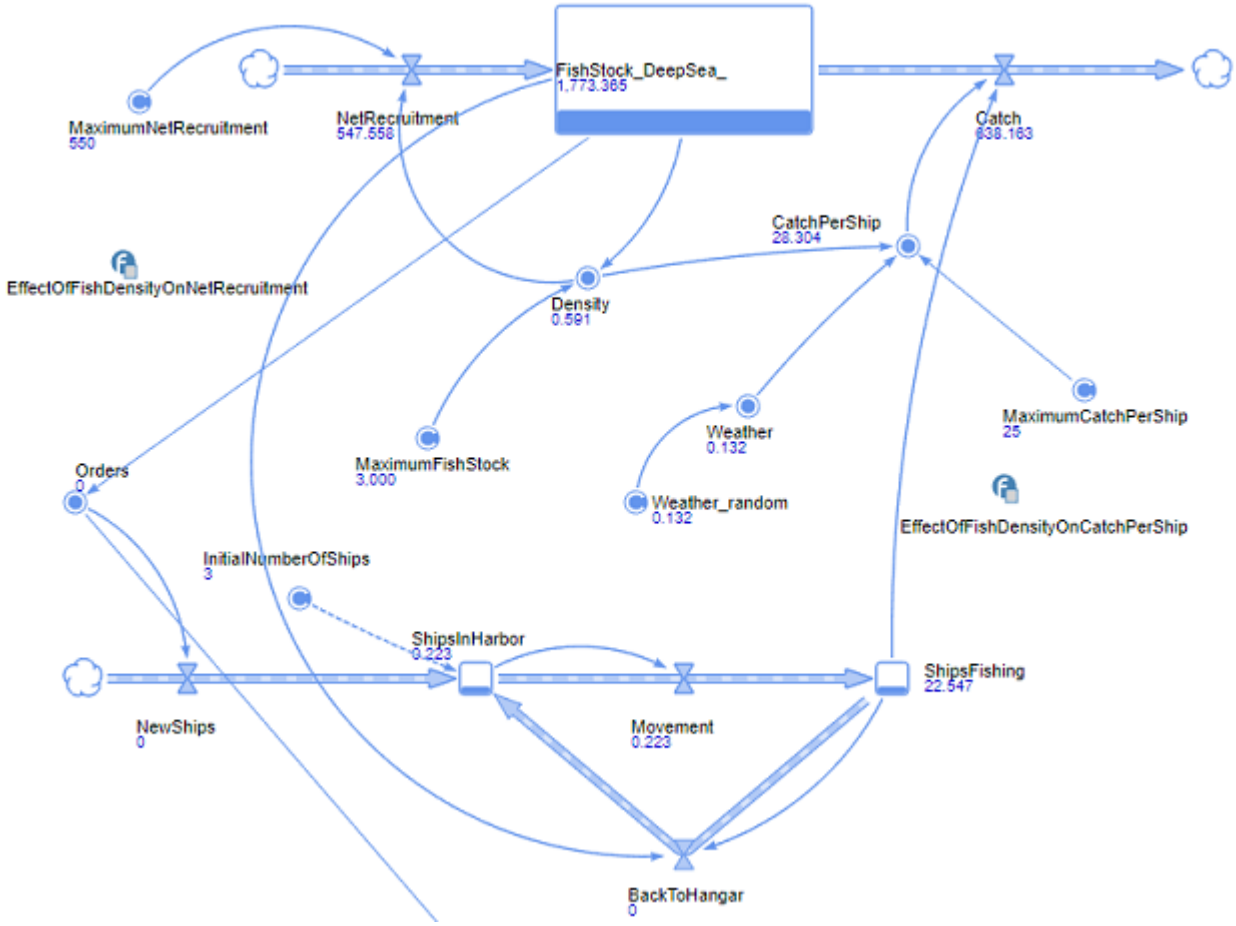

Figura 5: AnyLogic - Barcos e Mar

Como podemos ver na figura 5, as encomendas de barcos novos ("Orders") está diretamente relacionado com a quantidade de peixe no oceano, ou seja, enquanto o valor for acima de 2172 toneladas, o hangar recebe um barco novo, caso contrário, não recebe. Esta informação é enviada para "OperatingCosts"(figura 6), que calcula o preço de produção de cada barco e a despesa de cada barco no hangar ou em alto mar. Não foi possível encontrar o problema pois apesar do *harbor* só receber os valores 0 e 1, ele acaba sempre por ficar com valores decimais, o que adiciona uma imprecisão ao trabalho.

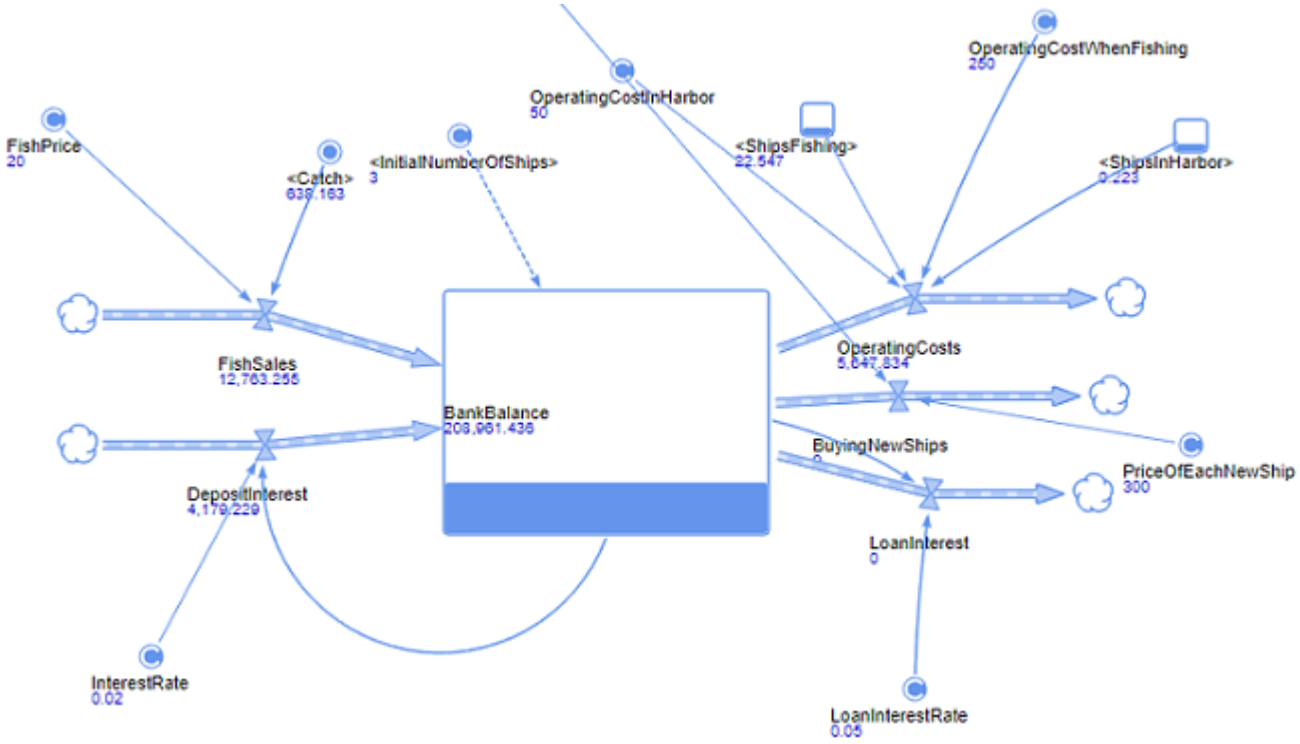

Figura 6: Encomenda de Barcos

Com o objetivo de lucrar o máximo possível, o valor da densidade não foi alterado pois maximiza o número de peixes no mar, que por sua vez maximiza as pescas.

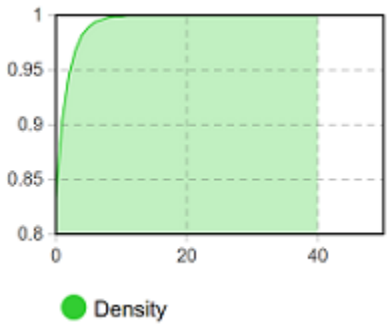

Figura 7: Densidade

3 Processing - Simulador/Jogo de Ecossistema

O objetivo desta parte do projeto era criar e simular um ecossistema baseado em presas e predadores, usando os conhecimentos adquiridos ao longo do semestre, nomeadamente a implementação de modelos baseados em agentes (as presas e os predadores) e de autómatos celulares (o terreno onde evoluem os agentes).

3.1 Terreno

De modo a construir o terreno, foi utilizada a Regra da Maioria.

Esta regra permite-nos ter um terreno sempre diferente. Utilizamos a mesma pois queremos ter sempre resultados diferentes no nosso simulador e, não só referente a uma configuração de terreno.

Assim, ao a configuração do terreno ser sempre diferente, serão colocados objetos em sítios diferentes, dando uma maior credibilidade ao simulador.

Passando então à explicação desta mesma regra, é uma regra simples que se baseia no uso de autómatos celulares.

Para a configuração das células deste autómato celular estar nos requisitos da regra da maioria, cada celula possuí as seguintes três sub regras:

1. Se a maioria das células vizinhas estar no estado 0, então, a célula muda para o estado 0.
2. Pelo contrário, se a maioria das células vizinhas estar no estado 1, então a célula muda para o estado 1.
3. Por fim, se a quantidade de células vizinhas com ambos os estados for igual então, a célula mantém o seu estado inicial.

Passamos agora à explicação de cada tipo de célula presente no autómato celular correspondente ao terreno.

O terreno é gerado com quatro diferentes de textura, cada uma corresponde a um objeto e, tem a sua funcionalidade no simulador.

Primeiramente a relva, este objeto pode ser "ingerido"por alguns dos animais presentes no meio. Quando a relva é "comida"por algum dos animais, esta desaparece, ficando apenas terra.

A relva está presente com a seguinte textura:

Figura 8: Textura da relva

a terra, à medida do tempo vai crescendo relva até se formar a relva outra vez. Esta está presente com a seguinte textura:

Figura 9: Textura da terra

As árvores são outro elemento presente no terreno. Este, desde o início ao fim, não desaparece, permanecendo até ao final da simulação.

Estas têm uso apenas para os pássaros, que permanecem nelas de modo a ganhar energia. Esta está presente com o seguinte aspeto:

Figura 10: Textura da árvore

Por fim, os espinhos presentes no jogo têm o intuito de tirar energia de animais terrestres. O mesmo tem a seguinte textura:

Figura 11: Textura de espinhos

3.2 Boids/Agentes Autónomos

Numa fase inicial foi adicionado imagens aos *boids* e para ter uma melhor perceção do ecossistema e torna-lo mais intuitivo, estas imagens foram inspiradas no videojogo Pokemon, que tem milhares de pokemons diferentes em aspeto, poder, habilidade.

3.2.1 Presas

Presa normal, *Prey*, é um agente que vagueia pelo terreno e come a relva, sem nenhuma condição adicional. Foi usado a seguinte imagem para representar a presa normal.

Figura 12: Presa

Presa inteligente, *CleverPrey*, é um agente que vagueia pelo terreno e come a relva mas tenta evitar os obstáculos. Foi usado a seguinte imagem para representar a presa inteligente.

Figura 13: Presa inteligente

Presa em grupo, *FlockPrey*, são agentes que comutam entre si, o grupo segue a mesma direção pelo terreno e comem a relva que aprece. Foi usado a seguinte imagem para representar a presa uma presa do grupo.

Figura 14: Presa de grupo

3.2.2 Predadores

Predador, *Predator*, é um agente que vagueia pelo terreno enquanto a sua energia for maior que 50 e tenta caçar as presas quando tem menos de 50 de energia, este predador não evita os obstáculos. Foi usado a seguinte imagem para representar o predador.

Figura 15: Presa inteligente

Predador Águia, *EaglePredator*, é o agente que está no topo da cadeia alimentar, este vagueia pelo terreno mas se encontrar uma árvore ele para e fica a descansar diminuindo o metabolismo perdendo assim menos energia por unidade de tempo, quando essa energia fica a menos de 50, este muda de comportamento para caçar a presa mais próxima. Este predador tem uma vantagem sobre os outros porque voa, logo não perde mais energia por passar num obstáculo. Foi usado a seguinte imagem para representar o predador.

Figura 16: Eagle Predator

3.2.3 Outros Animais

Abutre, *Abutre*, é um agente que só caça animais mortos e ficam nas árvores enquanto não houver, foi adicionado para o utilizador uma interação com o sistema: pôr armadilhas para matar animais, esses animais mortos são caçados pelos abutres. Abutres não podem ser caçados por nenhum animal e só podem caçar as presas mortas. Foi usado a seguinte imagem para representar o predador.

Figura 17: Abutre

3.3 Ecossistema

Com a junção do terreno e os boids obtemos o seguinte ecossistema:

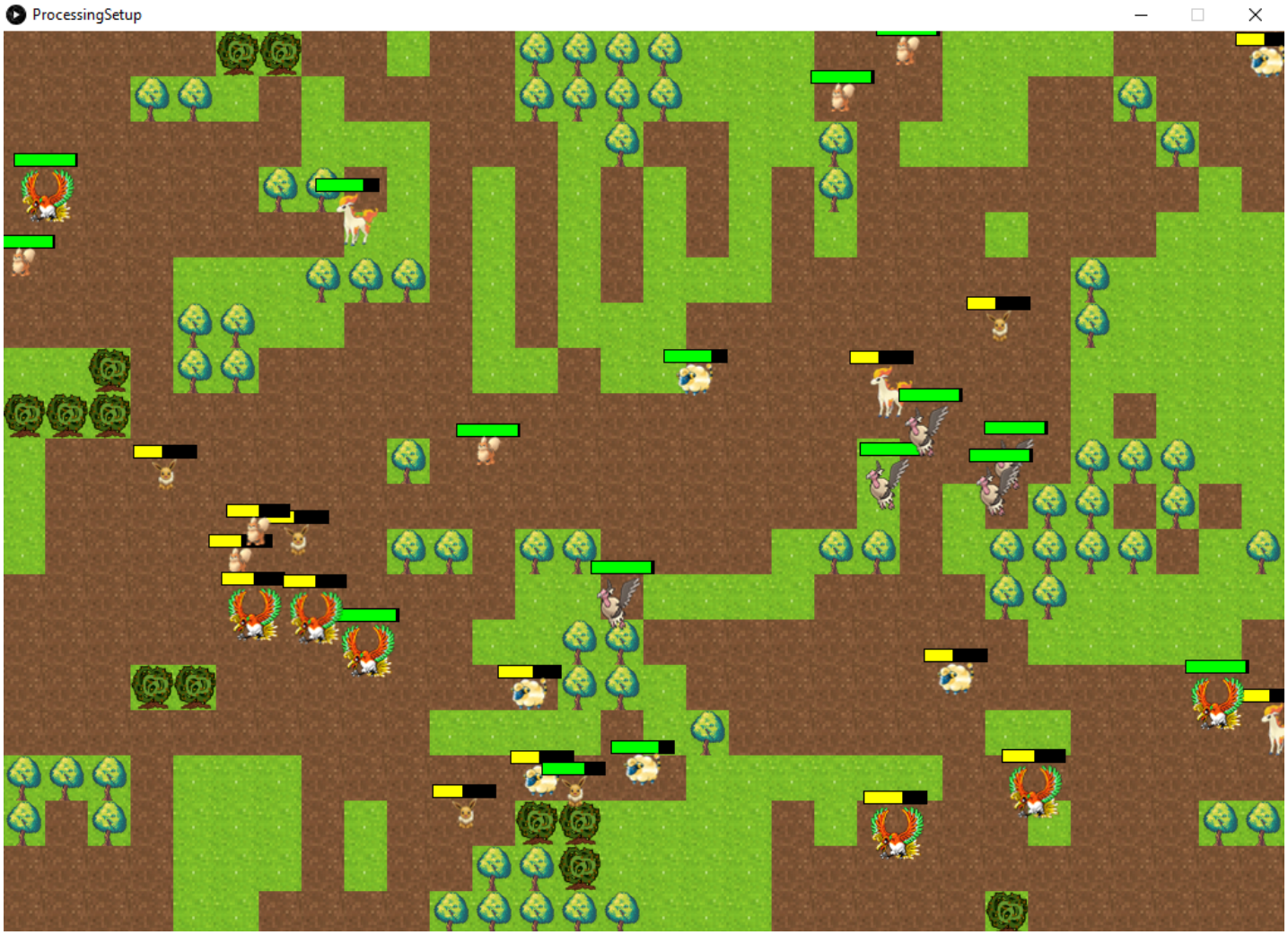

Figura 18: Ecossistema

Para a melhor perceção da vida dos animais fizemos um extra que é o *Health Bar* que indica a energia restante dos animais.

4 Diagrama UML

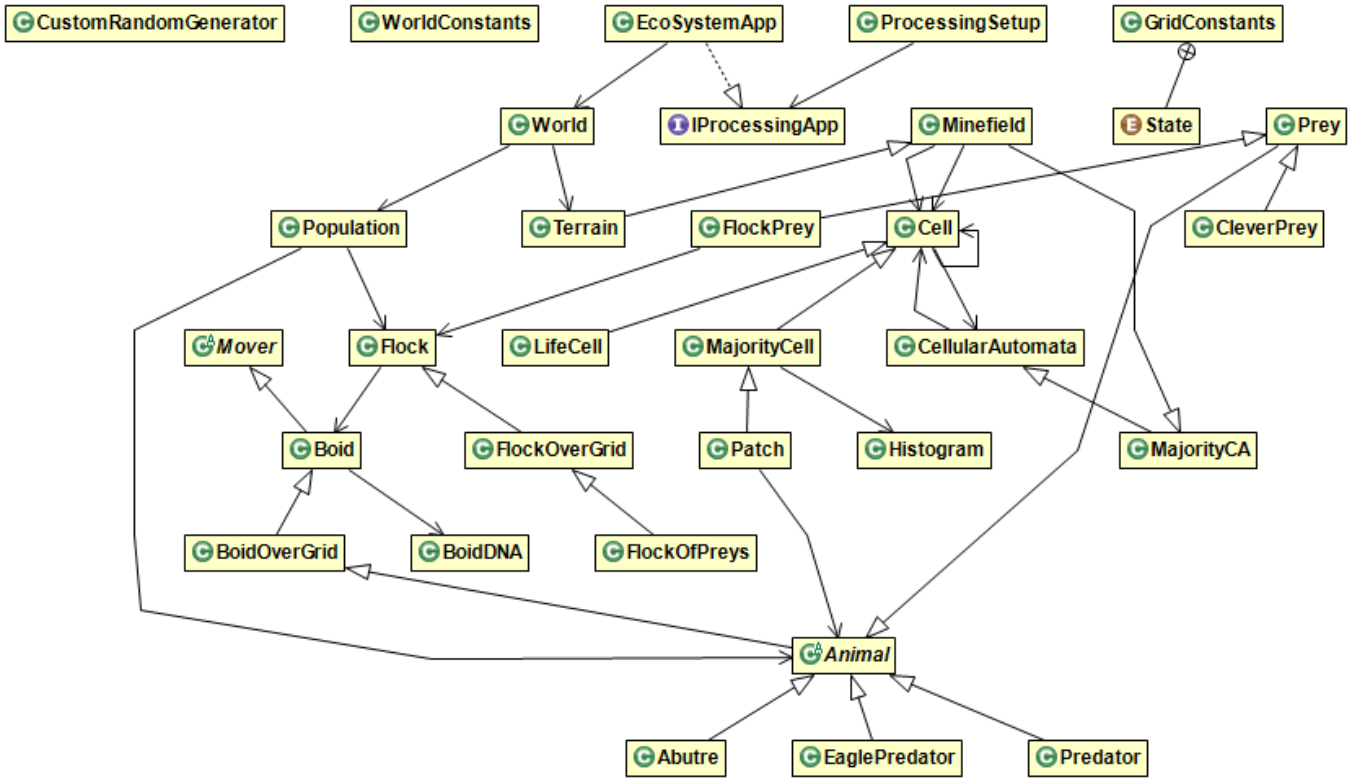

Figura 19: Diagrama UML

5 Conclusão

Com a realização deste trabalho conseguimos perceber melhor até que ponto podemos explorar um determinado ecossistema e que criar um sistema sustentável é complicado porque existe muitas variáveis pequenas ou grandes que tem um impacto no ecossistema todo.

Na realização da simulação de ecossistemas no *Processing*, encontramos várias dificuldades, nomeadamente a quantidade de processador que ocupava quando o número de *boids* ou células eram grandes que podiam ser melhorados se o código que fizemos fosse mais otimizado, outro problema foi o tempo que houve para fazer o trabalho, tínhamos planos para fazer mais mas devido ao tempo não conseguimos concretizar tudo o que tinhamos planeado.