

Relatório POO

Simulador de reserva natural

2022/2023

Trabalho realizado por:
Carlos Pinto | a2021155627 | P4
Alexandre Ferreira | a2021138219 | P9

Estrutura do programa e organização de ficheiros

De forma a dividir o programa de forma eficaz, o código além de se encontrar dividido por vários ficheiros (.cpp e .h) está também separado em pastas consoante o papel de cada class/função.

Como objetivo das principais classes temos:

- Animal:
 - Espécie:
 - Foram criadas subclasses derivadas de Animal e com funções “overriden” da função principal.
- Alimento:
 - Categoria:
 - De forma semelhante foram criadas subclasses de Alimentos.
- Comida:
 - Class utilizada para gerir o histórico de alimentos de cada animal.
- Reserva:
 - É a class principal já que é nela que estão desenvolvidas as principais funções de manipulação dos objetos.

Quanto à função main está implementado um ciclo que depois de ler e executar os comandos do utilizador, mostra a área visível da reserva (15 linhas/15 colunas).

Principais Classes implementadas

As classes (e respetivas subclasses) Animal e Alimento têm um funcionamento semelhante, visto que ambas são construídas com base na especificação das suas subclasses. Com isto, por exemplo um Coelho cria um Animal igual ao criado pelo lobo, porém com valores diferentes, depois de criado adiciona-se automaticamente a um vetor de Animais.

Quanto à Reserva é uma classe com o objetivo de gerir as informações da reserva natural e com isto é onde estão implementadas as principais funções de manipulação dos dados, de forma a agrupar estas funções num ficheiro comum.

Requisitos implementados

- Todos os comandos estão implementados, porém alguns encontram-se parcialmente incompletos (abordados à frente).
- A leitura de constantes do ficheiro “constantes.txt” está implementada.
- A leitura e execução dos comandos do ficheiro comandos está implementada e funcional.
- Todas as funções específicas dos animais(andar/comer/reproduzir/morrer) estão implementadas, porém algumas destas funções não estão completamente funcionais de forma a estarem apenas parcialmente implementadas(serão abordadas de seguida) .
- O comando “slide” foi (como permitido pelo enunciado) evitado, utilizando a leitura das setas do teclado como forma de movimentar a área visível ao longo da reserva.
- Foi utilizada a biblioteca disponibilizada como forma de imprimir na consola através de um sistema linha/coluna
- Quanto ao animal mistério foi criado um animal que anda ao longo da reserva de forma aleatória de forma a limpar todos os alimentos com cheiro a ketchup, a toxicidade destes alimentos transforma-se em saúde perdida(já que ketchup não é propriamente saudável). Além disto o animal morre quando a sua saúde chega a 0 (saúde inicial = 10) ou quando a tem uma idade igual a 30 instantes.

- O alimento mistério é gerado de forma a replicar os restos de ketchup dos bifés, isto é tem o cheiro a ketchup, duração de 20 instantes valor nutritivo e toxicidade 1 e 3, respetivamente.

Requisitos parcialmente implementados

- O comando “next” encontra-se parcialmente implementado já que a opção de iterar x vezes com uma paragem de y segundos entre cada não está completamente implementada, isto porque o terminal utilizado (através da biblioteca disponibilizada pelos docentes) apaga o conteúdo durante o intervalo de tempo.
- As funções de manipulação de dados da reserva estão completamente implementadas, porém algumas contém erros que por vezes fazem o programa “ir abaixo”, estes erros encontram-se principal (e talvez apenas) nas funções de movimento dos animais.