

Práctica 3: Mundo Hormigas

Objetivo

El objetivo de esta práctica es utilizar el polimorfismo dinámico en tipos de datos definidos por el usuario en un programa en lenguaje C++.

Entrega

Esta práctica se entregará en la sesión de laboratorio realizada entre las fechas del 15 al 19 de marzo. Durante esta sesión se podrán solicitar modificaciones sobre la práctica. En el actual escenario de presencialidad adaptada, la entrega se realizará en tareas separadas para la rotación con asistencia presencial y la rotación con asistencia online.

Enunciado

A partir de la implementación de la **hormiga de Langton modificada**, realizada en la práctica 2, en esta práctica se pide realizar las siguientes modificaciones:

1. En las prácticas anteriores se han implementado dos versiones alternativas del objeto **Mundo**. En la primera práctica la malla bidimensional tiene unas dimensiones constantes, fijadas al inicio de la ejecución y que no varían durante la misma. Cuando una hormiga rebasa alguno de los bordes de la malla, reaparece por el borde opuesto. En la segunda práctica se asume una malla bidimensional infinita. Inicialmente se establecen unas dimensiones, pero estas aumentan cada vez que la hormiga alcanza una posición fuera de la malla actual.

En esta práctica vamos a implantar un programa que permita al usuario seleccionar la implementación del objeto **Mundo** que quiere utilizar. Para ello, la clase **Mundo** se convierte en una interfaz (clase abstracta) que establece las operaciones que deben implementar las clases que definen los objetos mundos particulares (**MundoFinito**, **MundoInfinito**).

2. En el objeto **Hormiga** se quiere separar la implementación del código de actualización del movimiento. Para ello se crean métodos privados que calculan por separado el giro y el desplazamiento. Estos métodos privados son llamados desde el método público que actualiza el movimiento en cada paso de la simulación.

Se incrementa el número de hormigas que coexisten en el mundo. Inicialmente se crean en posiciones del mundo, y direcciones, dadas por el usuario. Las hormigas se almacenan en una lista ordenada por la posición que ocupa, de forma que el orden de las hormigas en la lista coincida con el orden en que se las encontrará el método de visualización de la malla del mundo.

3. El programa principal solicita al usuario que elija el tipo de mundo, así como sus dimensiones, y también el número de hormigas que coexisten en el mundo. Instancia al objeto **Mundo** y la lista de objetos **Hormiga** y los pasa como parámetro al constructor del objeto **Universo** para que realice la simulación.

4. El constructor del objeto **Universo** recibe como parámetros un objeto de tipo **Mundo** y una lista de objetos **Hormiga** colocadas sobre la malla del mundo. Controla el paso del tiempo, la actualización de las hormigas y la visualización del mundo hasta que el usuario detenga la simulación.