

FACULTAD DE CIENCIAS MATEMÁTICAS

PROGRAMACIÓN PARALELA

CURSO 2021-2022

Práctica 3: Programación Distribuida

ESTHER ARRIBAS GARCÍA, LUCÍA BRAGADO PÉREZ Y ÁLVARO SECO SILVA

Mayo, 2022



1.- Introducción

Esta práctica consiste en el planteamiento, diseño e implementación de una aplicación distribuida utilizando el sistema listener-client.

2.- Planteamiento

Se trata de un conjunto de naves situadas en un eje vertical hacia las que se dirigen horizontalmente varios meteoritos a velocidades aleatorias. Las naves sólo pueden moverse hacia arriba o hacia abajo en el eje vertical para intentar esquivar a los meteoritos.

Se cuenta con un sistema de puntuación que cuenta el número de meteoritos que impactan con cada nave. El ganador es el jugador cuya nave impacta con menos meteoritos. El criterio de parada del juego es alcanzar el tiempo máximo establecido.

3.- Implementación

Se crean dos archivos en Python:

- *sala.py*, que hace el papel de Listener o servidor
- *player.py*, que hace el papel de Client

En *sala.py* se definen las clases Nave y Proyectoil, donde se detallan los movimientos permitidos a los objetos pertenecientes a dichas clases. En el caso de las naves, sólo hay un atributo (la posición) y dos movimientos válidos: moveUp (hacia arriba) y moveDown (hacia abajo). Tenemos en cuenta que no se salgan de la interfaz gráfica, estableciendo topes en ambos extremos. Los proyectiles (meteoritos) tienen dos atributos (posición y velocidad) se mueven horizontalmente a velocidades aleatorias, y cuando llegan a la derecha de la pantalla de interfaz gráfica se reinician a la izquierda. Esto lo hemos llevado a cabo con el método restart. También hacemos un método choca_nave que, mediante las distancias entre las posiciones de nave y proyectil, decide si se produce una colisión o no. Estas clases se emplean para definir la clase principal Game, que controla la información general del juego.

En *player.py* también se definen las clases Nave y Proyectoil para organizar las posiciones de los objetos. Además se crean las clases Nave_Diseño y Proyectoil_Diseño para construir la representación gráfica de naves y proyectiles. La interfaz gráfica del juego queda elaborada con la clase Diseño, que emplea sprites. Además, incluimos música al juego.