**Flappy Bird**

Este proyecto consiste de dos módulos:

1. Juego de Flappy Bird.
2. Aplicación de una red neuronal NEAT (Neuroevolution of Augmenting Topologies) para hacer un algoritmo de aprendizaje reforzado para que el programa aprenda a jugar Flappy Bird.

Para realizar la primera parte usamos la biblioteca Pygame que permite hacer juegos interactivos en Python. En este recreamos el famoso juego de Flappy bird. Usamos las imágenes originales del juego y, siguiendo un tutorial del uso de Pygame, pudimos reproducir la jugabilidad.

El segundo módulo se realizó usando la biblioteca Neat-python (<https://neat-python.readthedocs.io/en/latest/>) que implementa el algoritmo NEAT creado por Kenneth O. Stanley (<http://www.cs.ucf.edu/~kstanley/#publications>). Esta permite implementar una red neuronal de evolución arbitraria desde Python.

**¿Cómo funciona?**

NEAT crea un conjunto de “genomas” que están compuestos de un par de “genes”, los genes pueden ser nodos o conexiones, a estos genomas se les proporciona una función de entrenamiento (fitness) que generará un valor que nos permita medir si un genoma específico está logrando resolver mejor el problema. Se selecciona un número de “generaciones” (veces que el algoritmo se ejecutará buscando mejorar la solución) y hará que los mejores genomas se “reproduzcan” y “hereden” sus mejoras a la siguiente generación.

El algoritmo da prioridad a los genomas que obtienen una calificación de entrenamiento superior y sigue su línea de “evolución”.

Todo esto suena muy similar a la teoría de evolución, donde sobrevive el más apto, y de hecho es así.

NEAT mantendrá a los genomas más aptos y a los que no mejoren o empeoren los “extingue”.

El algoritmo termina cuando se llega a la última generación o se logra superar el valor de entrenamiento que se considera “perfecto”.

En el caso específico del juego el valor de entrenamiento aumenta cuando un pájaro puede pasar uno de los tubos y disminuye si este choca con el tubo o cae al suelo.