|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. no** | **Title** | **Authors** | **Date of publication** | **Features** |
| **1.** | **A Minimal Training Strategy to Play Flappy Bird Indefinitely with NEAT** | * Matheus Cordeiro Tele informatics Engineering Dept. (DETI) Federal University of Ceara (UFC) Fortaleza, Brazil * Paulo Serafim Instituto Atlantico Fortaleza, Brazil paulo * Yuri Nogueira, Creto Vidal, Joaquim Neto Department of Computing (DC) Federal University of Ceara (UFC) ´ Fortaleza, Brazil | 3 December 2019 | 1. The game Flappy Bird   shows itself as a promising virtual testing environment to optimize agents whose goal is to learn the behavior of nondeterministic phenomena.  2. It is a popular game that was initially developed for mobile  devices. Its goal is simply to keep the bird, the player, alive  as long as possible by passing through a gap between pair of pipes without colliding with them.  3. When the screen is touched, the bird will perform a jump, on all other  moments, it will fall gradually as a result of gravity.  4. The Neuroevolution technique is presented as a powerful strategy to evolve Artificial Neural Networks (ANN) in unsupervised learning problems (problems where there is no input and output table).  5. It offers an alternative way to find the best configuration for an ANN without depending on a correct output value, which is commonly used to generate an error to optimize the networks settings through Descending Gradient algorithms like Backpropagation. |
| **2.** | **Reinforcement Learning in Python with Flappy Bird** | Anthony Le | 20 May, 2021 | 1. In 2014 the sleeper hit Flappy Bird took the mobile gaming world by storm. It has since been implemented in PyGame but most interestingly it lends itself well to reinforcement learning. 2. The agent (bird) can only perform 2 actions (flap or do nothing) and is only interested in 1 environmental variable (the upcoming pipes). The simplicity of this problem makes it perfect for implementing reinforcement learning in Python from scratch. 3. The reward function was defined to penalise -1000 for a death and 0 otherwise, such that the agent’s focus is the get as high a score as possible. This ensures that the reward function has sufficient impact after each episode, vs an implementation where rewarding +1 for a score increase means that penalisation has little to no effect. |