Inteligența Artificială în Jocul Flappy Bird

Implementarea algoritmului NEAT (NeuroEvolution of Augmenting Topologies) in Flappy Bird pentru antrenarea rețelelor neuronale artificiale. Scopul este de a evolua rețele neuronale care controlează mișcările unui personaj (pasăre) în joc, în încercarea de a maximiza scorul sau timpul de supraviețuire.

Configurare NEAT:

Configurația NEAT este încărcată din fișierul 'config-feedforward.txt'. Acest fișier conține diferiți parametri care controlează comportamentul algoritmului NEAT în timpul procesului de evoluție.

Clasa Bird:

Definește clasa Bird, care reprezintă pasărea controlată de jucător în joc. Include metode pentru sărit, mișcare și desenarea păsării.

Clasa Pipe:

Definește clasa Pipe, care reprezintă obstacolele în joc. Include metode pentru stabilirea înălțimii, mișcare, desenare și detectarea coliziunilor cu pasărea.

Clasa Base:

Definește clasa Base, care reprezintă solul în mișcare din joc. Include metode pentru mișcare și desenare a bazei.

Funcție de Utilitate:

blitRotateCenter: O funcție utilitară pentru rotirea și desenarea imaginilor în jurul centrului.

Bucla de Joc și Funcția de Evaluare:

Funcția eval_genomes este funcția principală de evaluare folosită de algoritmul NEAT. Rulează jocul pentru fiecare generație, actualizând fitness-ul fiecărui genom în funcție de performanța sa. Bucla de joc gestionează interacțiunea dintre păsări, obstacole și sol. Actualizează pozițiile, verifică coliziunile și ajustează fitness-ul genomelor în consecință.

Evoluție NEAT:

Funcția run inițializează algoritmul NEAT cu configurația furnizată și rulează procesul de evoluție (funcția eval_genomes) pentru un număr specificat de generații. Afișează cel mai bun genom găsit în timpul procesului de evoluție.

Config-feedforword.txt:

- fitness_criterion: Criteriul pentru evaluarea performanței este maximizarea fitness-ului.
- fitness_threshold: Fitness-ul minim necesar pentru a atinge un obiectiv.
- pop_size: Dimensiunea populației (numărul de indivizi într-o generație) este 50.
- reset_on_extinction: Dacă să se reseteze populația atunci când toate speciile se sting.
- activation_default: Funcția de activare implicită este tangenta hiperbolică (tanh).
- aggregation_default: Funcția de agregare implicită este suma.
- bias_init_mean, bias_init_stdev: Media și deviația standard a bias-ului la inițializare.
- bias_max_value, bias_min_value: Valorile maximă și minimă ale bias-ului.
- bias_mutate_power, bias_mutate_rate, bias_replace_rate: Parametri care controlează mutația și înlocuirea bias-ului.
- compatibility_disjoint_coefficient, compatibility_weight_coefficient: Parametri pentru calculul compatibilității genomelor.
- conn_add_prob, conn_delete_prob: Ratele de adăugare și ștergere a conexiunilor.
- enabled_default, enabled_mutate_rate: Starea implicită și rata de mutație a activării conexiunilor.
- feed_forward, initial_connection: Configurarea structurii rețelei.
- node_add_prob, node_delete_prob: Ratele de adăugare și ștergere a nodurilor (neuroni).
- num_hidden, num_inputs, num_outputs: Numărul de straturi ascunse, de noduri de intrare și ieșire.
- response_init_mean, response_init_stdev, response_max_value, response_min_value: Parametri pentru răspunsul nodurilor.
- response_mutate_power, response_mutate_rate, response_replace_rate: Parametri pentru mutația și înlocuirea răspunsului nodurilor.
- weight_init_mean, weight_init_stdev, weight_max_value, weight_min_value: Parametri pentru inițializarea și limitarea valorilor greutăților conexiunilor.
- weight_mutate_power, weight_mutate_rate, weight_replace_rate: Parametri pentru mutația și înlocuirea greutăților conexiunilor.

Bibliografie:

https://www.pygame.org/docs/ref/font.html

https://neat-python.readthedocs.io/en/latest/

https://www.youtube.com/watch?v=ihX3-WDua2I