Atividade Prática 01
Flappy Bird

Universidade Tecnológica Federal do Paraná (UTFPR), campus Apucarana
Curso de Engenharia de Computação
Sistemas Inteligentes 2 - SICO7O
Prof. Dr. Rafael Gomes Mantovani

1 Descrição da atividade

Flappy Bird (Figura 1) é um jogo para celular desenvolvido pelo programador de videogame vietnamita *Dong Nguyen*. O jogo é um *side-scroller* onde o jogador controla um pássaro, tentando voar entre colunas de canos verdes sem acertá-los. O jogo foi lançado em maio de 2013, mas recebeu um aumento repentino de popularidade no início de 2014 e se tornou um grande sucesso.

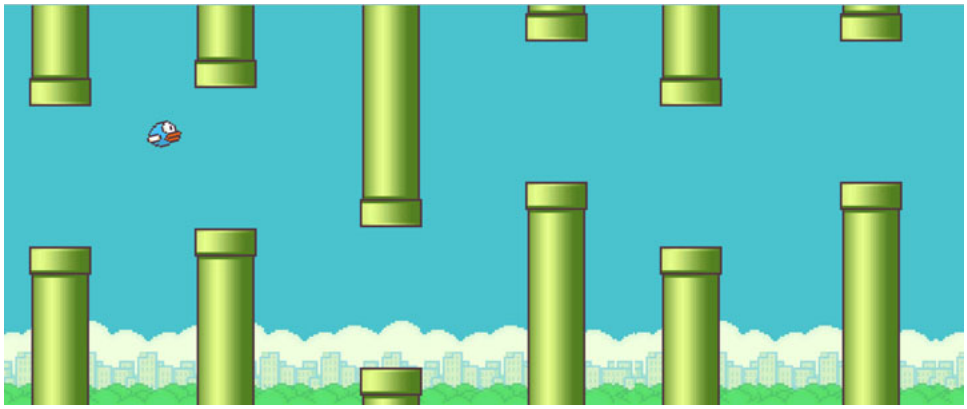

Figura 1: *Flappy Bird*: o jogador controla um pássaro cuja única ação possível é bater as asas, e evitar colisões com canos e o chão.

Flappy Bird será o objeto de estudo de nosso trabalho da disciplina. O objetivo é explorar o uso de Algoritmos Genéticos (AGs) para evoluir a arquitetura de um Perceptron Multicamadas (*Multilayer Perceptron* (MLP)) que controla o pássaro durante um jogo, e que maximize o tempo de jogo do pássaro. Consequentemente, o que vamos manipular é um problema de classificação, o que no jogo corresponde a dada entrada específica, determinar se o pássaro deve pular ou não para evitar a colisão com os canos.

Para facilitar o desenvolvimento, uma estrutura de jogo implementada em `Python` e usando a biblioteca *PyGame*[1] é disponibilizada. O código necessário para implementação de

[1] https://www.pygame.org/news

um jogo demo está disponível no Moodle (e github da disciplina) com o nome `Flappy.zip`. Na versão disponível é possível simular o jogo com um jogador humano, cujas ações são controladas pelo clique do mouse. A Tabela 1 descreve os arquivos contidos neste zip. Esses arquivos devem ser manipulados para que diferentes indivíduos sejam gerados pelo AG, avaliados, recombinados, e o melhor(es) selecionado(s) ao final da execução.

Tabela 1: Arquivos contidos no pacote de implementação do *Flappy Bird* em Python.

Arquivos que devem ser editados	
`GeneticAlgorithm.py`	Classe que define um Algoritmo Genético (AG) e seus operadores
`MLP.py`	Classe para implementação de uma MLP e suas operações
`Demo.py`	O arquivo principal que instancia e roda jogos de *Flappy Bird*. Pode ser usado como base para elaboração do arquivo principal da execução do agente inteligente.
Arquivos que podem ser ignorados	
`assets/`	Arquivos utilizados para a interface gráfica do jogo

2 Instruções Gerais

Implemente um Algoritmo Genético (AG) que otimize arquiteturas de MLP para jogar o *Flappy Bird*, maximizando a pontuação do agente durante o jogo. O projeto e modelagem do problema é livre e sem restrições. Logo, na modelagem em si, é preciso que vocês (equipe) definam:

- O que é um indivíduo no AG? Qual a sua codificação? Quais escolhas de um MLP o indivíduo deve conter?
- O que considerar como valores de entrada (*input*) para a MLP?
- Como implementar uma função de *fitness* que avalie bons indivíduos?
- Como selecionar os indivíduos mais aptos em cada geração? Roleta? Torneio? Qual o tamanho do torneio?
- Como realizar a reprodução (*crossover*) dos indivíduos em uma geração?
- Como realizar mutação na codificação escolhida?
- Usar ou não algum critério de parada antecipada no treinamento da MLP?

Façam a implementação do programa principal em um arquivo chamado `AT01-SI2-MainFlappy.py`, de maneira que seja possível executar a aplicação com o seguinte comando:

```
python AT01-SI2-MainFlappy.py
```

2.1 Dicas

1. lembrem-se que a definição dos operadores do AG é diretamente afetada pela forma que os indivíduos são codificados. Justifique todas as escolhas da modelagem adotada pela equipe no relatório;

2. se for o caso, façam execuções que possam mostrar se elitismo é vantajoso ou não;

3. além disso, como o AG é aleatório, façam diferentes execuções com diferentes *seeds* para ver como o algoritmo se comporta para diferentes problemas;

4. façam experimentos com diferentes operadores e configurações iniciais considerando tamanhos de população e gerações. Discorra sobre a convergência ou não das soluções.

2.2 O que entregar?

Vocês (equipe) devem entregar um único arquivo compactado contendo os seguintes itens:

- os arquivos fonte do projeto codificado;
- um relatório (em PDF) apresentando análise e comentários/explicações sobre os resultados obtidos. Descreva e explique também partes relevantes do código implementado.
- O trabalho deve ser submetido pelo Moodle até o prazo final estabelecido na página do curso.

3 Links que podem ser muito úteis !!!

- https://pygad.readthedocs.io/en/latest/
- https://pypi.org/project/geneticalgorithm/
- https://github.com/Cleyton200K/Perceptron_MultiLayer-Flappy_Bird
- https://www.pygame.org/project/4970
- https://medium.com/@sanilkhurana7
- https://github.com/ikergarcia1996/NeuroEvolution-Flappy-Bird

Referências

[1] LUGER, George F. Inteligência artificial. 6. ed. São Paulo, SP: Pearson Education do Brasil, 2013. xvii, 614 p. ISBN 9788581435503.

[2] RUSSELL, Stuart J.; NORVIG, Peter. Inteligência artificial. Rio de Janeiro, RJ: Elsevier, 2013. 988 p. ISBN 9788535237016.

[3] LUKE, Sean. Essentials of Metaheuristics. Second edition. Lulu: 2013. Available for free at http://cs.gmu.edu/$\sim$sean/book/metaheuristics/

[4] DE JONG, Kenneth A. Evoluationary Computation: A Unified Approach. The MIT Press: 2006.