ECE4179 Final Project Brief

Team members
Wong Zhi Wei 28326865, Ivan Chan Jing Ming 28325028, Eugene Low You Jun 28696492

Title: Deep Reinforcement Learning for Playing Snake Game

Background into the problem/Task you wish to explore

"The Snake" is a popular game created in 1997 being first introduced in Nokia Phones. The snake begins with a length 1 and increases its length by 1 after eating a "fruit". After that, a new randomized positioned fruit is generated on the game board. This has introduced a competitive environment between players. However, the problem lies with the limitation of the human as it requires a combination of spatial and mechanical skills.

Objectives for this project is:

- Outperform human performance/human benchmark on playing snake games.
- Developing a Deep Learning Model to effectively complete the game with the highest score.

Brief overview of your proposed method/solution

The following model will adopt a Deep Q-Network (DQN)methodology:
The Q value is updated accordingly to the Bellman's equation, Q value is initialised with random weights. Initialize memory, *D* with capacity of *N.*
Epsilon-greedy, ε initialised = 1.
For episode = [1 , M]:
For *i* = [0 , *T*]:

1) Get the current state, s_t (observation space).
2) With epsilon-greedy, ϵ, select random action, a_t. Otherwise, $a_t = max_a \ Q^*(a),\ 1 - \epsilon$. So as the training goes on, the random action would minimize and relies more on its learnt network. (Exploration or Exploitation)
3) Execute a_t and and snake will move according to the a_t. Reward, r_t is given and the new state, s_{t+1} is captured.
4) Stores the a_t, s_t, s_{t+1} *and* r_t into the memory, *D*. Update the stack of the last 4 frames.
5) Sample random minibatch of transition ($s_{t+1},\ a_t,\ s_t,\ r_t$) from *D*. (Replay Memory)
6) Then update Q-value, $Q_{NEW} = y_i$ according to the Bellman's equation. But if the game ended at s_{t+1}, $y_i = r_i$.
7) Back propagation: $Loss = MSE\ (Q_i - y_i)\ or\ \frac{1}{N}\sum_{i=0}^{N-1}(Q_i - y_i)^2$ where $y_i = Q_{NEW}$

As for Step 5, after storing the experiences into the memory buffer, *D* with enough experiences, then sample a random batch of experiences from the memory buffer [1-6].

Initial research conducted

Initial possible algorithm patterns [7]:

- Best search distance = |p1-p2|+|q1-q2|, the distance of the snake head and fruit.
- Greedy best first search.

Proposed datasets/training environments

Proposed dataset: States from each frame when running “The Snake” from Pygame modules.
Proposed training environment: “The Snake” from pygame

References:

[1] V. M. and K. K. , "Playing Atari with Deep Reinforcement Learning," *DeepMind Technologies,* pp. 1-9, 2013.

[2] M. C. "towards data science," 15 November 2018. [Online]. Available: https://towardsdatascience.com/how-to-teach-an-ai-to-play-games-deep-reinforcement-learning-28f9b920440a. [Accessed 20 April 2021].

[3] "DeepLizard," 1 December 2018. [Online]. Available: https://deeplizard.com/learn/video/0bt0SjbS3xc. [Accessed 25 April 2021].

[4] J. T. "towardsdatascience," 16 August 2020. [Online]. Available: https://towardsdatascience.com/deep-q-network-dqn-ii-b6bf911b6b2c. [Accessed 25 April 2021].

[5] "DeepLizard," 3 November 2018. [Online]. Available: https://deeplizard.com/learn/video/Bcuj2fTH4_4. [Accessed 25 April 2021].

[6] freeCodeCamp.org, “An introduction to Deep Q-Learning: let's play Doom,” *freeCodeCamp.org*, Mar. 29,2018. [Online]. Available: https://www.freecodecamp.org/news/an-introduction-to-deep-q-learning-lets-play-doom-54d02d8017d8/. [Accessed 30 Apr 2021].

[7] S. Sharma, S. Mishra, N. Deodhar, A. Katageri, and P. Sagar, “Solving The Classic Snake Game Using AI”, 2019.