Geometry Dash

Reinforcement Learning

By Daniel Guerrero-Martin, Ellison Eitel, Justin Kerns, Luke Mayer, and Rithvik Malladi

Picture Credits: Geometry Dash IGN

The process and methodology

As we were in the idea phase on the project we thought that developing some sort of game agent would be a fun and interesting concept. We decided to do some sort of reinforcement learning agent for the game Geometry Dash that could solve problems related to unbeaten levels and could possibly help profession players with new strategies to beat levels or obtain new times. As we developed said reinforcement agent for the game we thought it would be an interesting idea to also create a neuroevolution model for the game to compare and contrast the models.

Picture Credits: Redbubble pciture

Picture Credits: Geometry Dash IGN

Main Goal

Our main goal was to create a model that that was able to train on then complete levels in geometry dash

We attempted to accomplish this in two ways

- A Q-learning agent
- Neuroevolution

We also had a secondary goal if we were successful of creating a generalized agent that could complete any level after being trained on a few select levels

Similarities between the two models

- Firstly both models successfully at his point don't complete the levels after being trained on them but can make significant progress and while they can be trained on multiple levels they are not generalized and must be trained on each level
- Secondly both models take in each frame from the game in order to create the observation space that is used in the calculation as well as using the progress bar in order to calculate the success of each run.
- The visual quality of the game and other performance graphics needed to be minimized in order to train the model effectively

How we did it? Q-Learning Model

- This model uses double deep q learning in combination with a CNN in order to to create a reinforcement learning agent.
- It also uses a gymnasium implementation creating a Geometry Dash environment in order to calculate the reward, step, as well as the action and observation spaces using screenshots of each frame of the game and then processing it.
- The model uses a balance of exploration and predicts the action based on the fit of the neural network which will update based on the progress made and reward given

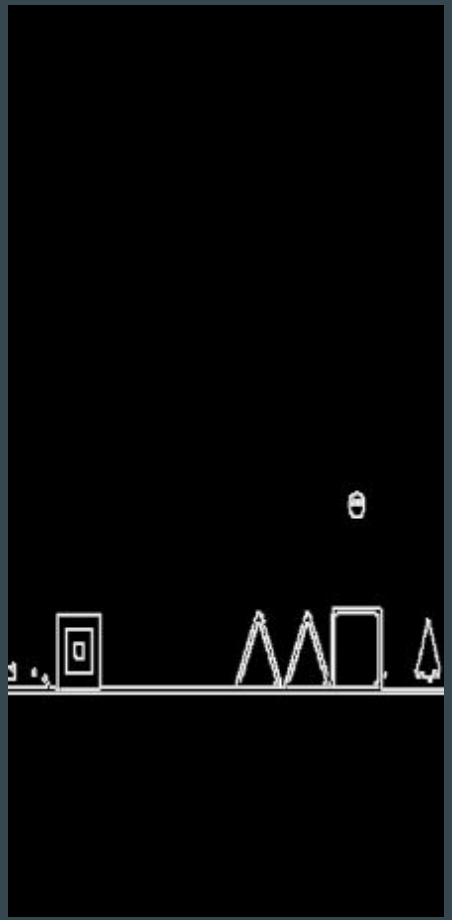
Difficulties and Limitations, Reinforcement Learning Model

- During training the model would often need to be continuously tweaked or changed if it got stuck on a particularly difficult section of the level. We also has to switch from a neural network combination to a neural network double deep q-learning algorithm
- The processing of the frames as screenshots into usable data is extremely difficult, however it's the only way to obtain the observation space from Geometry Dashas we don't have access to the data from the game.
- In order to run completely without slowdown the frame rate would need to be capped on Geometry Dash and other parameters such as graphics as well as window size would need to be adjusted.
- As previously mentioned the model is not generalized and would need to be trained on each level in order to complete it

How we did it? Neuro-evolution Model

- The Neuroevolution model was created using an implementation of NEAT.
- NEAT creates a random population of agents (neural networks) to do actions based on the input image.
- We then evaluate the performance of this population on the level and choose the best agents as the starting point for the next population.
- Over time the population will tend to get better and better at the level.

Input image:



Difficulties and Limitations, Neuroevolution Model

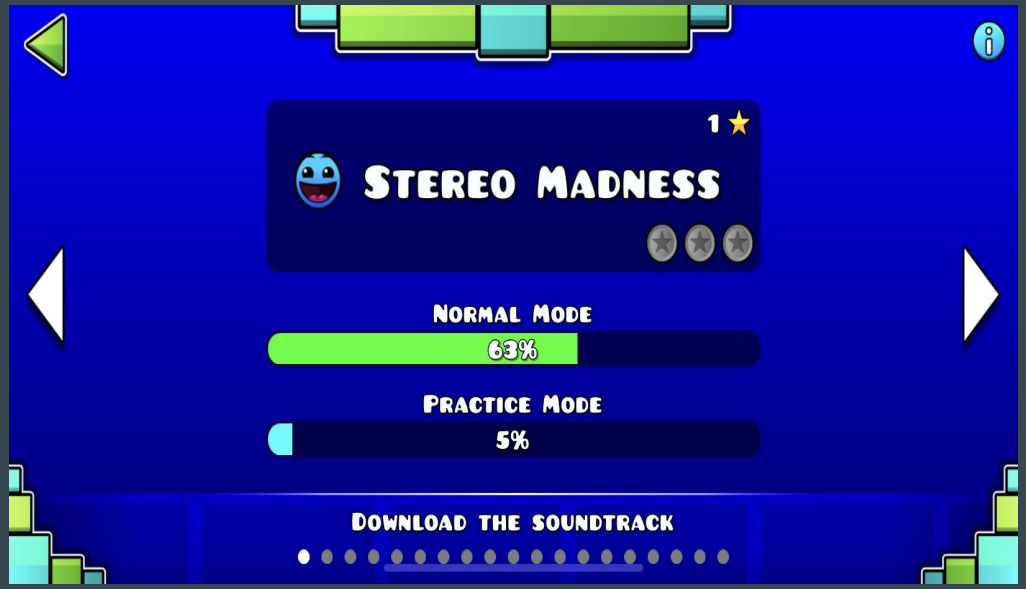
- The neuroevolution model take a large amount of time due to the fact we cannot train agents simultaneously.
- We experimented with different sizes and formats for the input to the model and ultimately went with a resized image of a simplified image.
- The neuroevolution model requires tuning of various model hyperparameters in order to ensure that the model chooses the right agents to mutate after each generation.
- Ideally, this model can generalize but because of how complicated geometry dash is, performance on other levels would not be very good without rerunning the evolution algorithm.

Outcomes and Results

- Both models are able to comfortably get to around 16% on the levels they've been trained on.
- The reinforcement learning agent had a progress of 34% consistently/on average after 1500 iterations and the neuroevolution model had a progress 18% at the highest and 16% on average.
- In total we trained the NEAT and reinforcement models for around 100 hours while tweaking hyperparameters over the semester.
- Also since neither of our models ever truly solved their problem it's hard for us to get an accuracy or z-score reading on the model in its current state
- It seems as though in this case the reinforcement learning agent performs better that the NEAT model though I suspect if we were able to use access the game data through memory that might not be the case

What's next?

- Reading memory of the game
- Using a different game agent
- Implementation of this as a tool to be used by professionals easily
- Successfully creating a generalized models to run on any level



Citations

Chandrakant, W. by: K. (2024, March 18). Reinforcement learning with neural network.

Baeldung on Computer Science. https://www.baeldung.com/cs/reinforcement-learning-neural-network

Chapman, J., & Lechner, M. (2024, March 16). Keras Documentation: Deep Q-learning for atari

breakout. Keras. https://keras.io/examples/rl/deep_q_network_breakout/

Li, T., & Rafferty, S. (2017). Playing geometry dash with Convolutional Neural Networks.

Stanford.CS. https://cs231n.stanford.edu/reports/2017/posters/605.pdf

Nuer, J. (2022, May 20). A comprehensive guide to deep Q-learning. Medium.

https://medium.com/@jereminuerofficial/a-comprehensive-guide-to-deep-q-learning-8aeed632f52f

Citations Cont.

SethBling. (2015, June 13). Mari/O - Machine Learning for Video games [Video]. YouTube.

https://youtu.be/qv6UVOQ0F44?si=mNP6A9265S-ZFnKb

van Hasselt, H., Guez, A., & Silver, D. (2016). Deep Reinforcement Learning with Double

Q-Learning. Proceedings of the AAAI Conference on Artificial Intelligence, 30(1). https://doi.org/10.1609/aaai.v30i1.10295

Image Citations:

Geometry dash. IGN. (n.d.). https://www.ign.com/games/geometry-dash

Aerma. (n.d.). RobTop sneak peek cube Geometry Dash 2.2 Icon. Redbubble. https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.redbubble.com%2Fi%2Fsticker%2FRobTop-sneak-peek-cube-Geometry-Dash-2-2-Icon-by-Aerma%2F158409371.EJUG5&psig=AOvVaw3k-Sbb1U0hGsmUVMluocjf&ust=1715314752023000&source=images&cd=vfe&opi=89978449&ved=0CBIQjRxqFwoTCLD-vMnb_4UDFQAAAAAdAAAAABAQ