Project Proposal

Goals

* Implement a Deep Learning NN to take input data from Nintendo Entertainment System games and output controller inputs.
* The aim is to primarily use the net to attempt high-scoring play at Arcade-style games (Pac-Man, Galaga, Gradius, Donkey Kong, Tetris, Punch-Out!!)
* If the net performs well enough, we may experiment with more complex games, such as those consisting of linear progressing levels (Mario, Castlevania, Mega Man, Kirby, Ghosts and Goblins) Additionally, it would be interesting to see the net attempt even more difficult exploration-oriented games (Zelda, Metroid, Final Fantasy)

Tools

* FCEUX (NES and Famicom ‘All in One’ emulator)
* Lua (for accessing emulator data and providing controller input)
* Python (for performing the training and classification)
* PyTorch

Project Outline

* Collecting data from the game
  + A Lua script will be used to collect pixel and game state data per frame and will write this data into a file.
  + The python script for the Deep Learning NN will read the data from this file and input it into the net.
* The Neural Network
  + The Neural Network will be created using PyTorch
  + Its input will be the pixel data for the game being played and the game state data
  + We will likely be implementing a MDP utilizing the net for the policy or Q function.
    - Possible states will be based on game state and pixel data
    - Possible actions will be controller inputs
    - Reward will be primarily based on the ingame score
      * Could potentially also be influenced by other factors like remaining time, lives remaining, etc.
  + The output will be a set of controller inputs to be executed in the current frame
* Sending controller inputs to the game
  + Once the net has generated controller inputs for the game, the python script will write these inputs into a separate file.
  + A Lua script will be reading the data from this file and will set its contents as the controller input ingame for the current frame.