Goals:

- Create an agent that can play DOOM at a superhuman level
    - Plays Singleplayer at a superhuman level (speed run?)
    - Plays Multiplayer (vs other agents or humans) at a superhuman level
    - Actually learns the complexities of the game (i.e. getting ammo when out, getting health when low, deciding whether to run or to fight)
    - Agent can learn to use the different weapons in the game and learn what scenarios weapon A is better in than weapon B
- Subgoals that might be fun to explore after/along the way:
    - A general FPS playing agent that isn't a total noob when dropped in an entirely new FPS game given that the relative graphics and rules of the game are the same (i.e. CSGO and Call of Duty, Quake and Doom, etc). Retains the basic understanding of point and shoot semantics

Problems:

- Enemy Identification
    - Arnold claimed to struggle with enemy detection

- The action space
    - Continuous and hierarchical
    - Should be able to move and shoot fluidly (i.e. not stopping and shooting)

- F1 bot claimed to have gotten stuck (i.e. keep moving forward but blocked by an explosive barrel) and specifically designed rules to detect and fix them
- The walls could be differently colored (DoomNet had this problem)

Baseline Model:

- Use frame stacking (n=4) (n x n x 4 -> n x n) using MaxPool?
- Extract visual features from the stacked frames with conv nets (n x n -> h)
- Use LSTMs to keep state
- PPO algorithm to advance policy

Solution :

- Maybe a separate policy for each gun
- DIfferent networks and/or policies for navigation and gun handling?

Reward Shaping:

- Most common rewards from papers:
  - Negative living reward (penalize agent who just lives)
  - Negative health loss reward
  - Negative ammo loss reward
  - Positive item pickup (health, ammo, etc)
  - Distance penalty (penalize agent who stays still)
  - Distance reward (reward agent for moving)

Our Reward Shaping:

- I'm thinking of shaping the item specific rewards based on what the current doom agent has. I.e. instead of just giving positive values for picking up health only give positive reward IF they're health increased from picking up health, same goes for ammo etc. Hopefully doesn't complicate the learning of this mapping though, maybe can do curriculum training and specifically train this relationship

-

Models:
Arnold:

- Paper: https://arxiv.org/pdf/1609.05521.pdf
- Code: https://github.com/glample/Arnold

F1:

- Paper: https://research.fb.com/wp-content/uploads/2017/04/paper_camera_ready_small-1.pdf?
- Code: https://github.com/mihahauke/VDAIC2017/tree/master/f1/F1_track1

IntelAct:

- Paper: https://arxiv.org/pdf/1611.01779.pdf
- Code: https://github.com/mihahauke/VDAIC2017/tree/master/intelact/IntelAct_track2
- Unofficial Implementation: https://github.com/flyyufelix/Direct-Future-Prediction-Keras

TSAIL:

- Paper: https://www.ijcai.org/Proceedings/2019/0482.pdf
- Depth/Detection Code: https://github.com/huangshiyu13/ViZDoom_Collection

DoomNet:

- Code: https://github.com/akolishchak/doom-net-pytorch

Marvin

- Code (maybe?): https://github.com/heronsystems/adeptRL/commit/61c93be67aed52fde457abb05f4c0126dc7cbb36
- Also see Other Code below for 2016/2017 versions

Other Code:

- http://www.cs.put.poznan.pl/mkempka/misc/