Materials and General Info

Past Attempts -

https://github.com/OpenHoldem/openholdembot

Research -

<https://www.codeproject.com/Articles/783225/An-alternative-introduction-to-Genetic-and-Evoluti>

<https://www.codeproject.com/Articles/792887/Travelling-Salesman-Genetic-Algorithm>

https://www.codeproject.com/Articles/707505/Genetic-Algorithms-Demystified

<http://poker.cs.ualberta.ca/publications.html>

http://www.cs.cmu.edu/~sandholm/

Game Engines -

<https://github.com/phstc/javapokertexasholdem>

https://github.com/fblaha/pokershell

Basic Implementation -

<https://www.codeproject.com/Articles/791914/How-to-train-your-Pokerbot>

<http://cowboyprogramming.com/2007/01/04/programming-poker-ai/>

https://gym.openai.com/docs/

BASIC ASSUMPTIONS -

Texas Hold’em heads up fixed limit (1 on 1 with pot limit)

* http://poker.wikia.com/wiki/Fixed-limit

We never run out of money

TODO List -

Choose engine

Set up game

* Must accept inputs and produce output
* Create basic “AI” that always calls every game
* Must be able to execute an “agent-environment loop” - See OpenAI docs below

Choose framework to work with

**Observations**

If we ever want to do better than take random actions at each step, it'd probably be good to actually know what our actions are doing to the environment.

The environment's step function returns exactly what we need. In fact, step returns four values. These are:

* observation (object): an environment-specific object representing your observation of the environment. For example, pixel data from a camera, joint angles and joint velocities of a robot, or the board state in a board game.
* reward (float): amount of reward achieved by the previous action. The scale varies between environments, but the goal is always to increase your total reward.
* done (boolean): whether it's time to reset the environment again. Most (but not all) tasks are divided up into well-defined episodes, and done being True indicates the episode has terminated. (For example, perhaps the pole tipped too far, or you lost your last life.)
* info (dict): diagnostic information useful for debugging. It can sometimes be useful for learning (for example, it might contain the raw probabilities behind the environment's last state change). However, official evaluations of your agent are not allowed to use this for learning.

This is just an implementation of the classic "agent-environment loop". Each timestep, the agent chooses an action, and the environment returns an observation and a reward.