Project Title: *Mortal (Q)ombat*

CS 238 Status Update

Anthony Weng, Aviad Lengo

1 Problem Re-introduction

Brawlers are a sub-class of video games in which a player pilots a character (also called a "fighter") against another fighter, controlled by a human player or pre-programmed AI. Brawler games are unified in that, since character actions cause deterministic effects, the player with more "skill" will win a given round, game, etc. Skill in brawler games can be described as a combination of *proactivity* and *reactivity*, which describe a player's ability to predict what an opponent will do and plan accordingly and to make dynamic adjustments when an opponent deviates from said prediction, respectively. For our final project, we define our problem as follows: to utilize reinforcement learning (RL) to train an AI agent to play a brawler game with a better-than-random and (ideally) human-comparable level of skill.

2 Completed Work

1. Specification of the RL problem environment (e.g., state attributes, action, reward, terminal conditions) and agent learning approach. At a high-level, we construct a 4-layer neural network to employ the deep Q-learning approach.
    - Key design choices include: generating **two** sets of neural network weights, training the network/agent once by having it fight itself and once by having it fight a random agent; an epsilon greedy action selection regime; use of training and target networks to encourage learning stability; and use of an experience replay buffer to minimize learned correlation between subsequent game states.
2. Creation of `Game` and `Fighter` classes which represent agent-environment interactions and agent dynamics and attributes, respectively.
3. Creation of `DeepQNN`, `Buffer`, and `Trainer` classes which represent the deep Q-learning network, a replay buffer for deep Q-learning, and neural network training loop where hyperparameters can be easily specified and training initiated via a call to `trainer.train(...)`.
4. Creation of a `demo.py` file where games between some pairing of human, random, and trained agents can be created and interacted with/observed. Will be used for trained agent evaluation.
5. Partial completion of training the deep Q-learning neural network. As of submission, $\approx$ 1500 training games have been conducted, consisting of $\approx$ 15,000,000 training experiences.

Figure 1: A screenshot of the gameplay environment and agent training in action!

3 Revised Timeline to Completion

1. Week 8 (prior week): Train both AI agents (i.e., train deep Q-learning neural network weights), with a goal of achieving $\approx$1,000 training games for each agent and their respective training partner (i.e. deep Q-learning/NN agent, random agent). Begin preliminary work on the final project report (likely opting for the paper option).
2. Week 9: Finish any training of the AI agents and begin evaluations of each AI agent against both random and human agents. Evaluation will likely consist of a fixed number of games with win percentages being reported. Continue work on final project report and create/include key figures like training time, average training loss, and reward over time.
3. Week 10: Continue work on and complete the final report. Ideally, we will be able to package the demo of the game as an executable for anyone to try out!