Reinforcement Learning of Physics-Based Character Interaction

> Jinda Li, 1034907
> Before this project, I have the experience of using ml-agent of Unity to train a ping-pong game agent in a narrative game project.

Motivations

Inspired by DeepMimic [1], I am trying to discover the possibility of character-environment interaction by reinforcement learning. My motivation is to obtain realistic physical interaction when the character attacks a target with a physical weapon. The method is to train a physics-based agent to imitate a pre-record attacking animation, and at the same time, it should properly control the weapon in the hand to get higher hitting power and keep balance when the hit happens.

Method

I used the marathon-envs [2] framework for setting up. Besides utilizing the style transfer by replacing the target guiding animation, I as well did more attempts.

- Add a stick-like weapon to the ragdoll character. It has weight so it requires the agent to keep balance.
- Add a muscle (Configurable Joint in Unity) to simulate the wrist for the reason of controlling the weapon.
- Create the target with random mass (0.5-3) to improve robustness.
- Modify the reward function by adding a new hit reward component and rescaling all coefficients. The hit reward encourages the agent to hit the target with greater power. This reward is given by

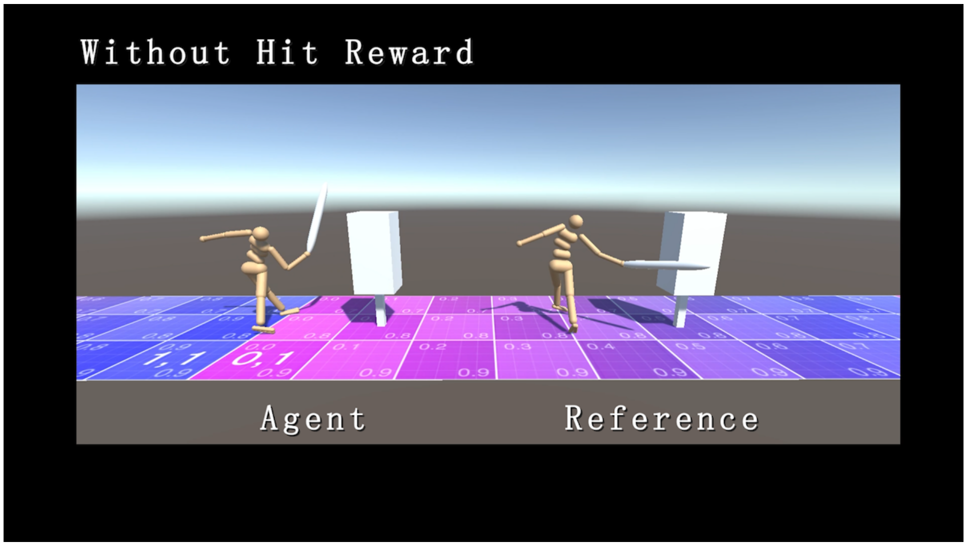
$$r_t = \begin{cases} \min(v/v_{max}, 1\ ), & if\ has\ hit \\ \exp(-4|p_t - p_w|^2, & otherwise \end{cases}$$

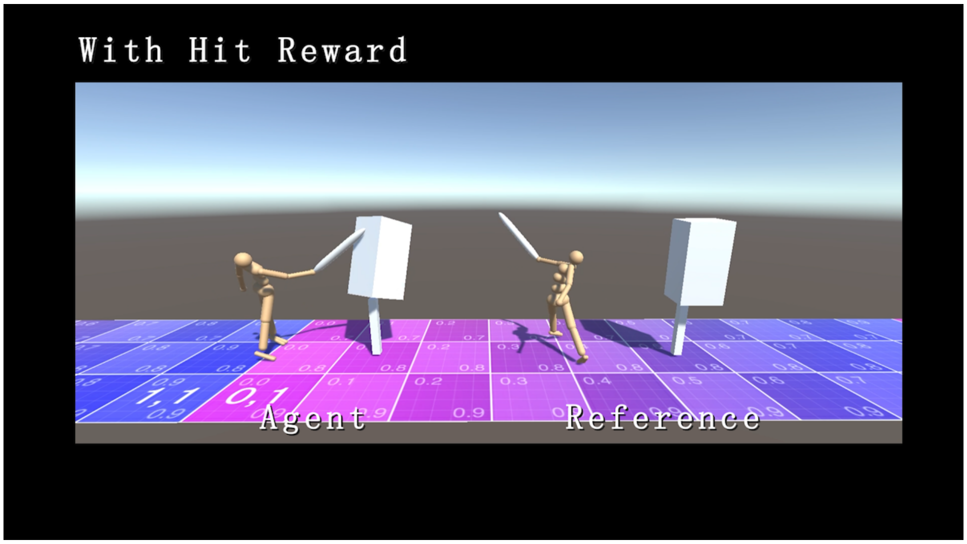
> v is the relative velocity between the stick weapon and the target when the hit happens, v_{max} is the maximum relative velocity. p_t means the position of the target and p_w is the position of the weapon. Other rewards components are described in the reference paper.

I compared the training result of the model with and without hit reward. Without a hit reward, the agent acts more likely to keep balance rather than to hit something because the stick weapon has a certain mass. When the hit reward is included, the agent tries to wave the stick to hit the box.

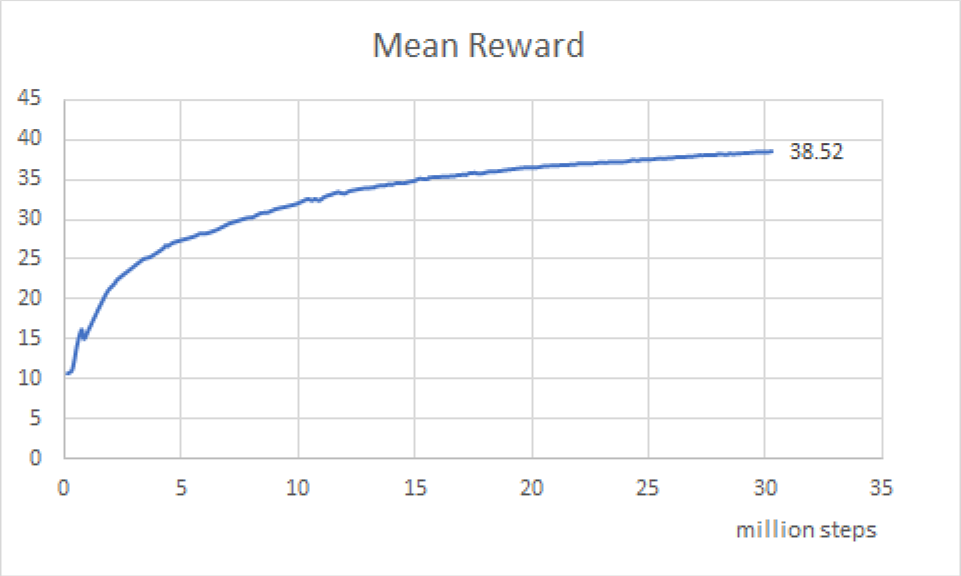
There is a slight pause when the stick collides with the target which represents the realistic effect of interaction force (The target box has a heavier weight in the situation of this picture). When the reference animation ends, the agent is still in the process of hitting. But it seems to re-balance after the influenced by the reaction force.

> Click here to watch the video





The plot below shows the mean reward curve of 30,300,000 steps of training.



I learned the code structure of the marathon-env about physics-based character, which also leads me to the MuJoCo physics engine and motion matching principles. Having read the paper on DeepMimic, DReCon, etc., I learned how to design the reward function to obtain the desired result, and also I got more interest in AI-boosted animations.

In most combat games, animations are played undisturbedly and there are “clipping” or “model glitch” problems. I have an interest now to see if it is possible the physics-based interaction can be applied to games where a spear blocking a sword attack by physics is a unique feature. Motion matching and signal filter methods can be implemented to improve the quality of animations. I am considering extending this topic to my master thesis project.

Reference

[1] *Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. 2018. DeepMimic: example-guided deep reinforcement learning of physics-based character skills. ACM Trans. Graph. 37, 4, Article 143 (August 2018), 14 pages. DOI: https://doi.org/10.1145/3197517.3201311*

[2] *Joe Booth, Vladimir Ivanov. 2020. Realistic Physics-Based Character Controller*, https://doi.org/10.48550/arXiv.2006.07508