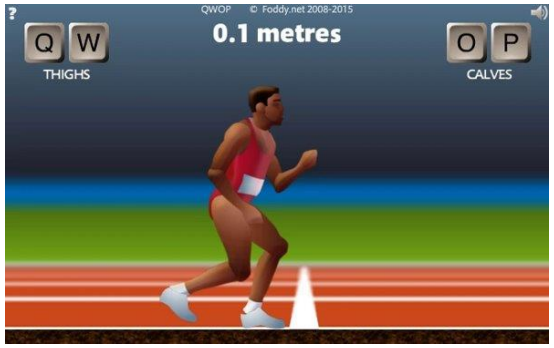
Model-based reinforcement learning for browser game QWOP using DreamerV2

David Guevara

Background



64x64 pixels

- Reward
- Additional info:
  - Position X, Y
  - Torso angle
  - Head angle
  - Thighs angles
  - Calves angles
  - Terminal (T/F)

DreamerV2 -> Master Atari games

Tested against Walker2D (MuJoCo)

Most games are not 8-bit games

Let's test how it does with a mix of both Atari and Walker2D

Make it get to the goal

Algorithms

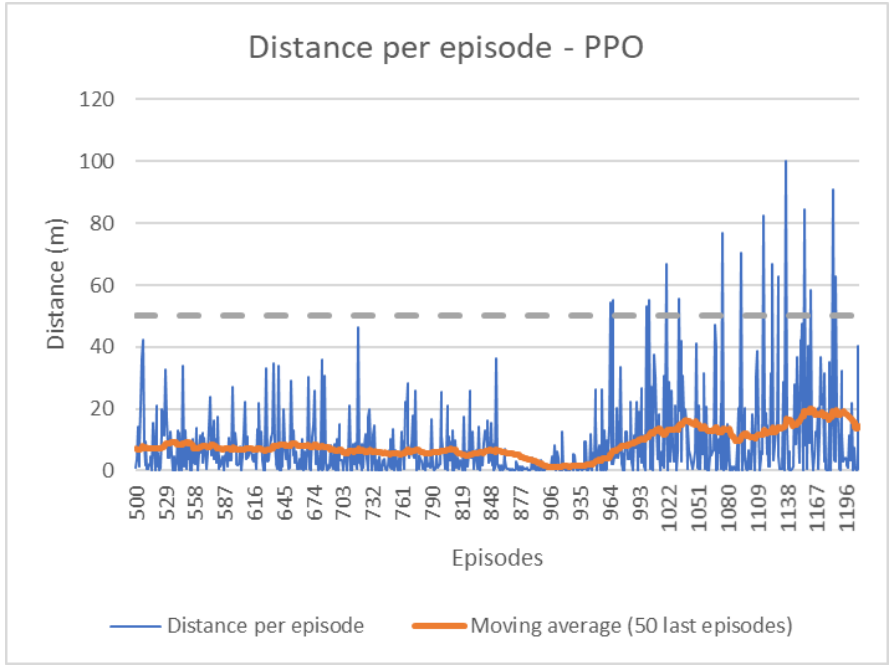
- DreamerV2
  - Run prefill episodes
  - Understand how it works
  - Run training inside its "mind"
  - Test and adjust inner environment
- Proximal Policy Optimization (PPO)
  - Model-free
  - Small steps within trust region
  - This makes it very stable

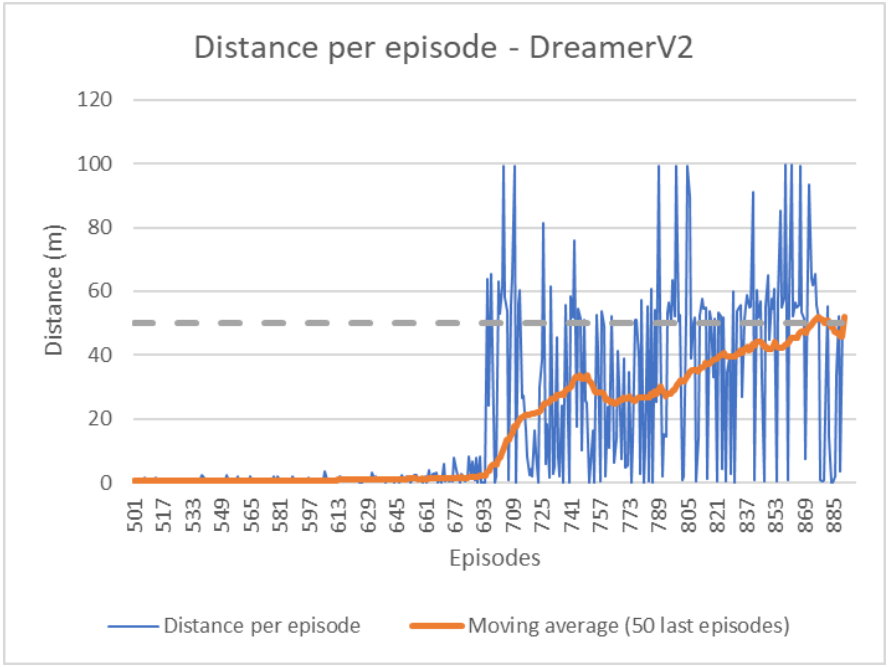
Methods

- Environment – Calculate different rewards and return the current distance
- DreamerV2 – Accept the state, reward, remember max distance per episode and log per episode
- Proximal Policy Optimization – Remember max distance per episode and log per episode
- Tested different rewards
  - **Velocity**
  - Torso angle * velocity
  - Velocity * distance
  - Distance

Findings

Algorithm	Max. avg. distance (m) (50 last episodes)	No. of victories in 1200 episodes	No. episodes for first victory	Max. avg. Reward (50 last episodes)
DreamerV2	**52.078**	**7**	**704**	0.015
PPO	19.915	1	1208	**0.154**

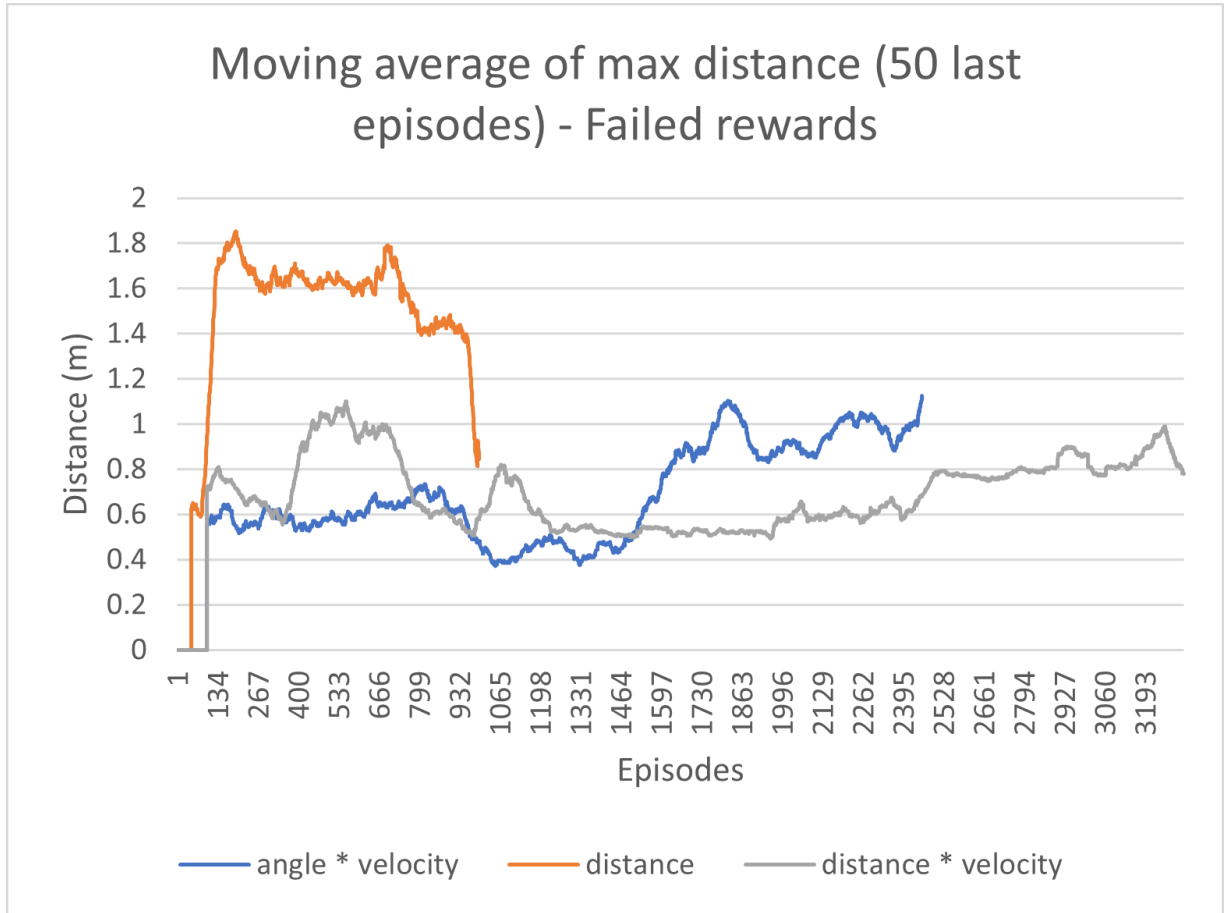




In brief…

- Summary
  - Trained Dreamer and PPO for QWOP
  - Everything was already built-> Put everything together

- Conclusions
  - Velocity seems to be the only effective reward
  - 50-meter mark -> checkpoint
  - DreamerV2 learned quicker
  - PPO prioritized mean velocity
  - DreamerV2 prioritized not having negative velocity


Moving average of max distance (50 last episodes) - Failed rewards
Distance (m)
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
1
134
267
400
533
666
799
932
1065
1198
1331
1464
1597
1730
1863
1996
2129
2262
2395
2528
2661
2794
2927
3060
3193
Episodes
angle * velocity
distance
distance * velocity