




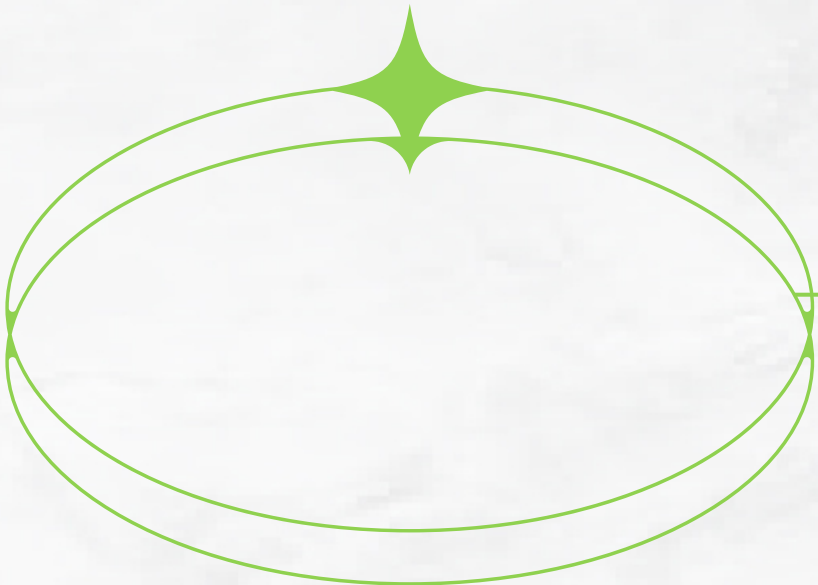
MUJOGO

APRENDIZADO POR REFORÇO

Gustavo Sanches
Lucas Treuke



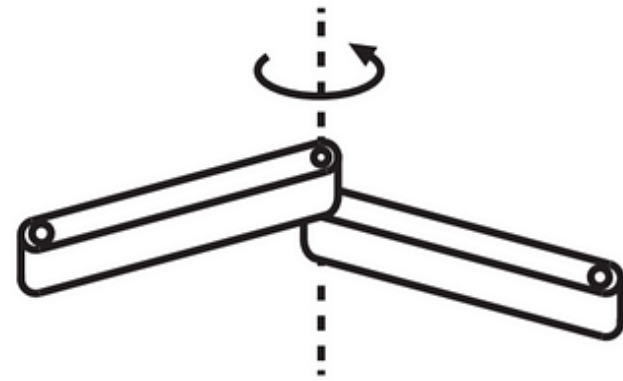
O QUE É O MUJOCO?



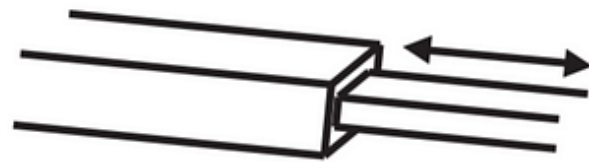
"MuJoCo stands for Multi-Joint dynamics with Contact. It is a general purpose physics engine that aims to facilitate research and development in robotics, biomechanics, graphics and animation"

<https://mujoco.readthedocs.io/en/latest/overview.html>

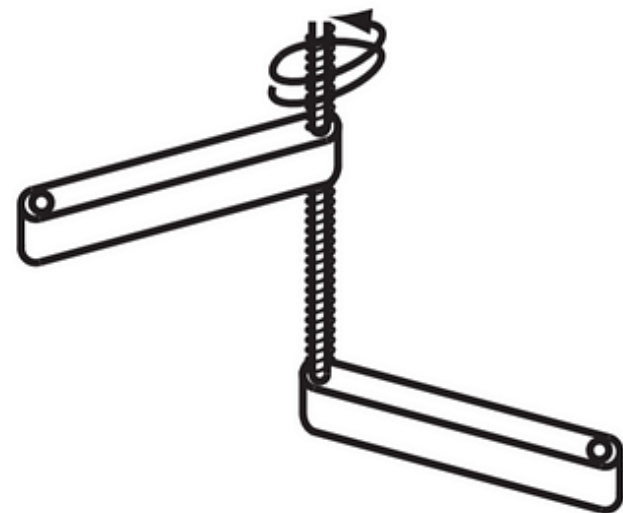
TIPOS DE JUNTAS



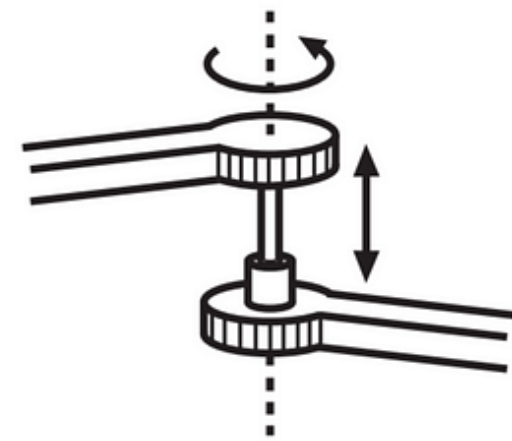
Revolute
(R)



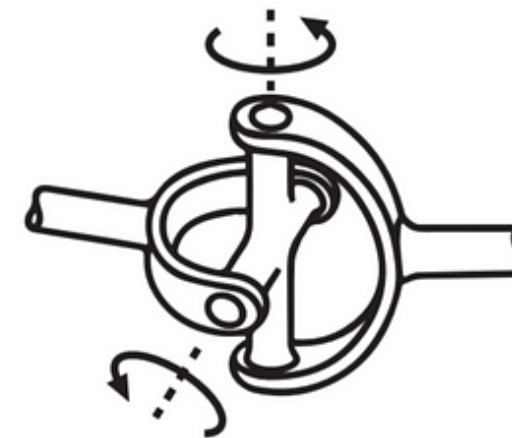
Prismatic
(P)



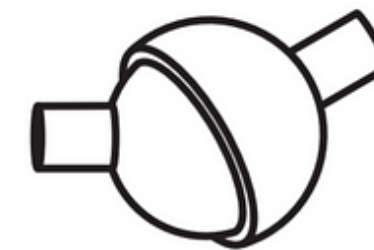
Helical
(H)



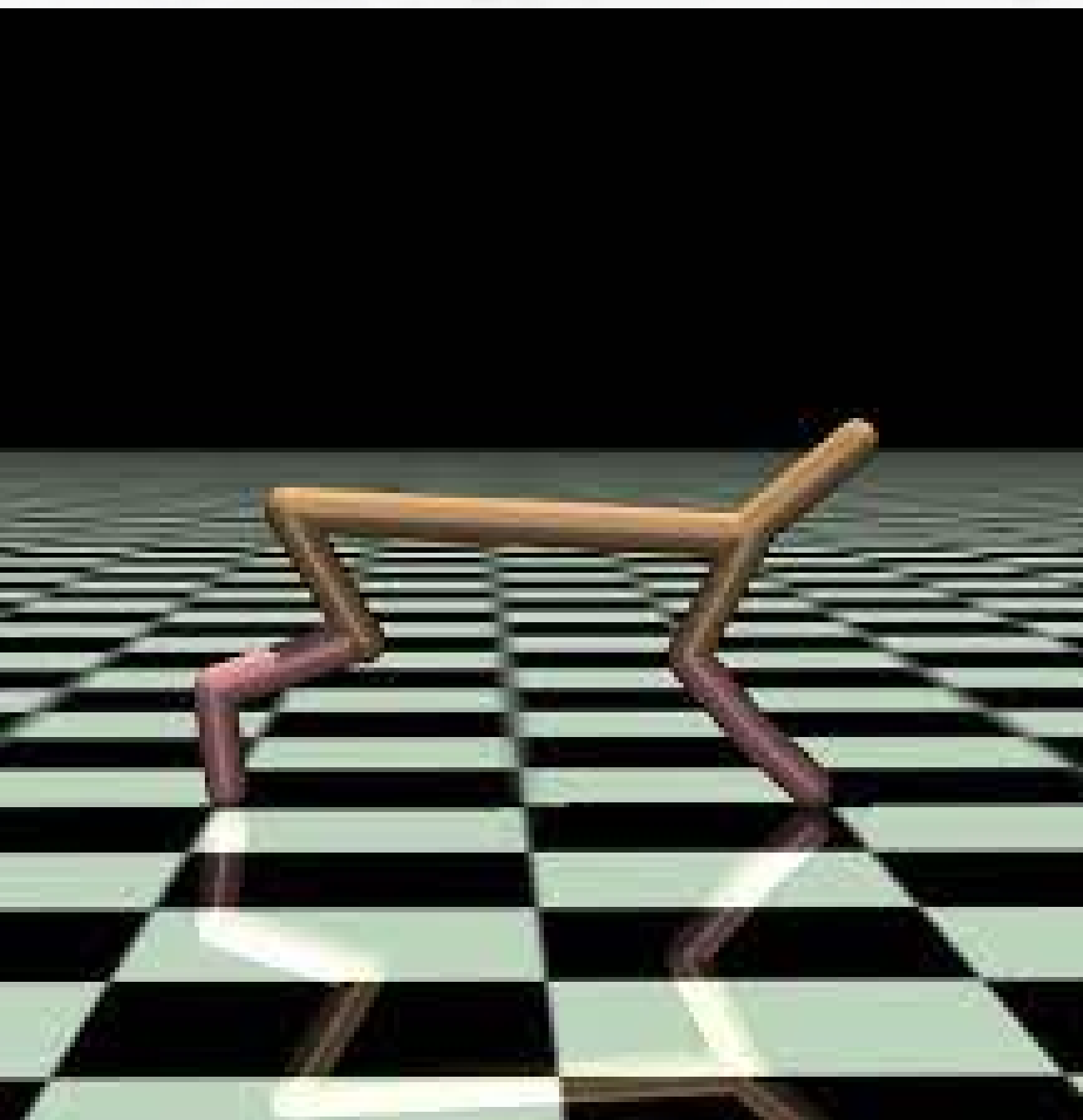
Cylindrical
(C)



Universal
(U)



Spherical
(S)



HALF CHEETAH



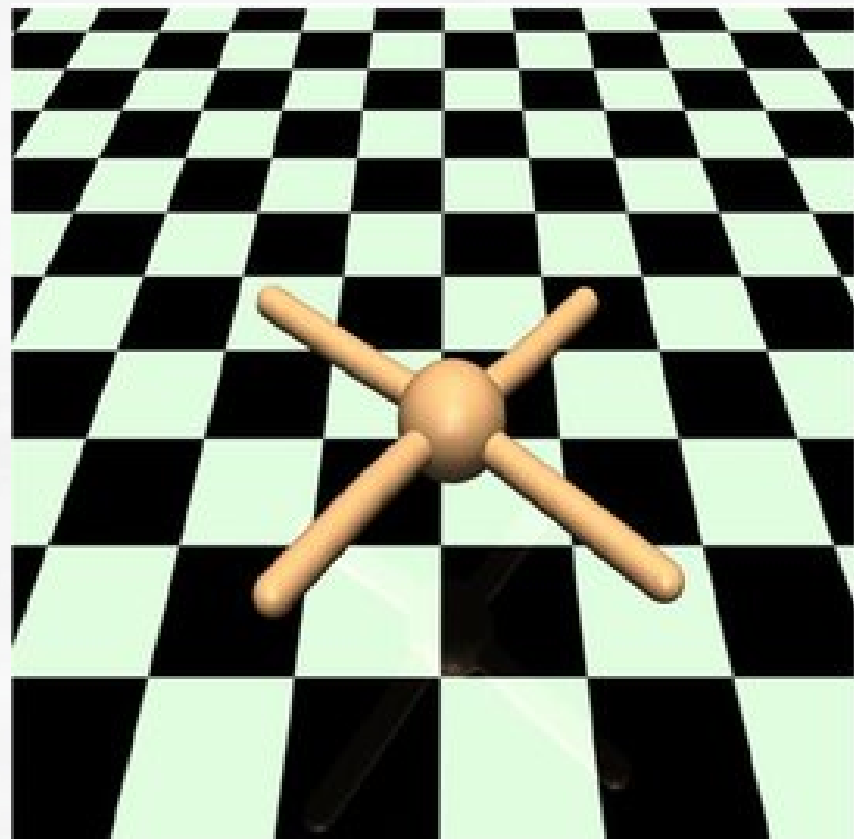
SOBRE

Action Space: $(6,1)$ com valores de -1 a 1

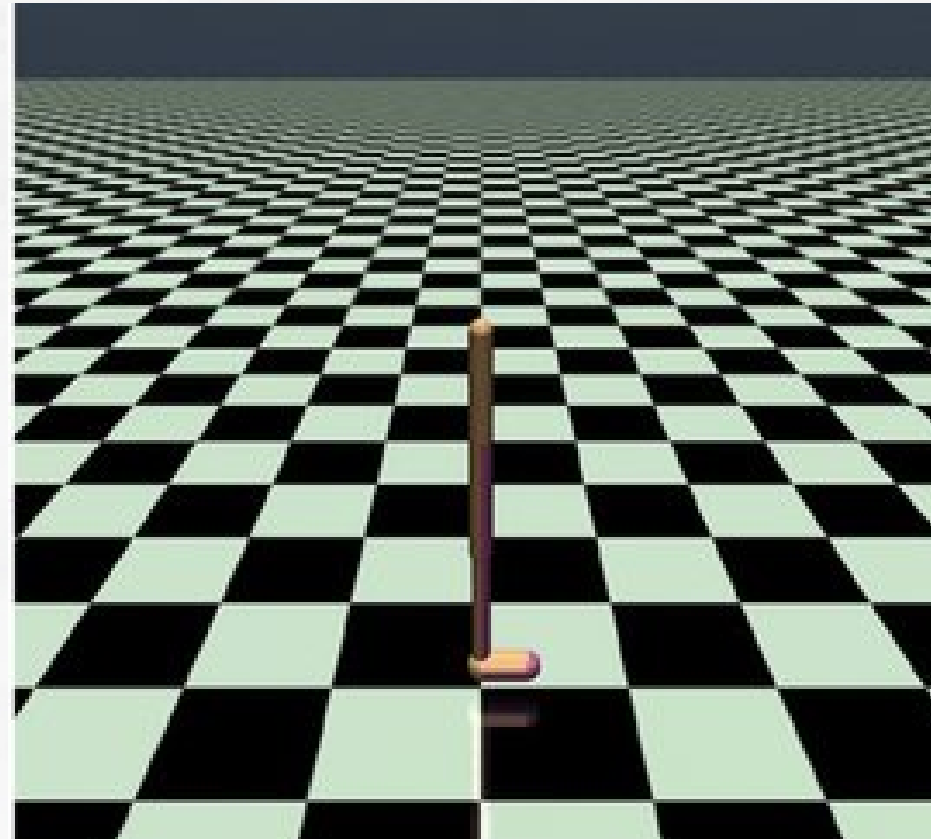
State space: $(17,1)$ com valores de $-\infty$ a ∞

Reward: Distância em x percorrida - tamanho das ações

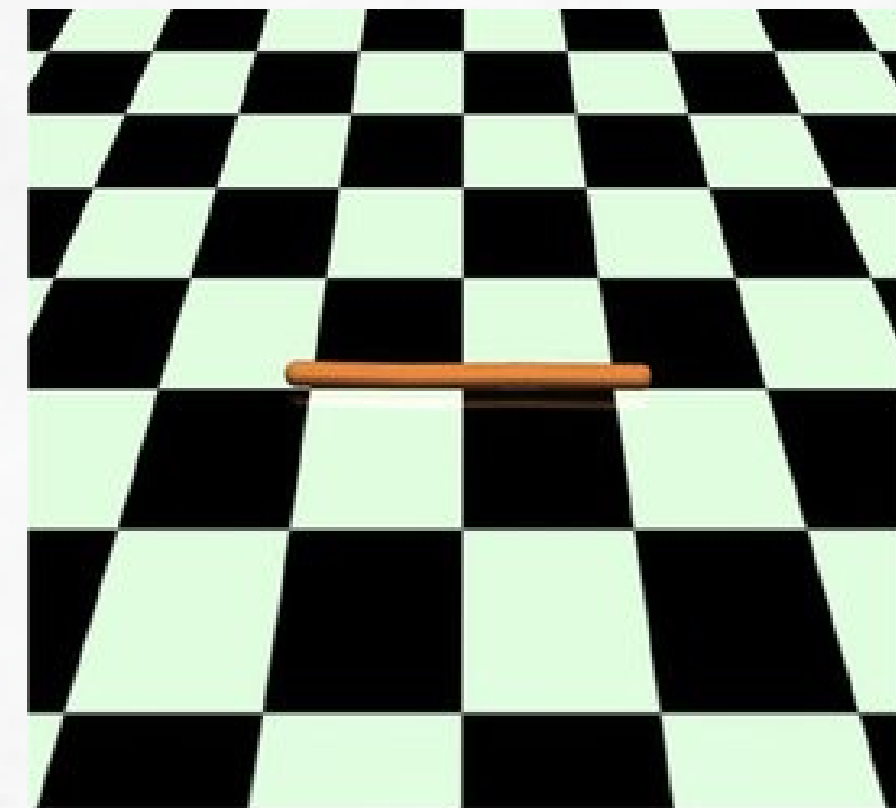
OUTROS MODELOS



Ant



Hopper



Swimmer



ALGORITMO

DQL X Actor-critic



ALTA DIMENSÃO ESPAÇO DE OBSERVAÇÃO

DQL resolve o problema de uma alta dimensão no espaço de observação



ALTA DIMENSÃO ESPAÇO DE AÇÃO

DQL possui dificuldade de lidar com espaços de ação com alta dimensionalidade



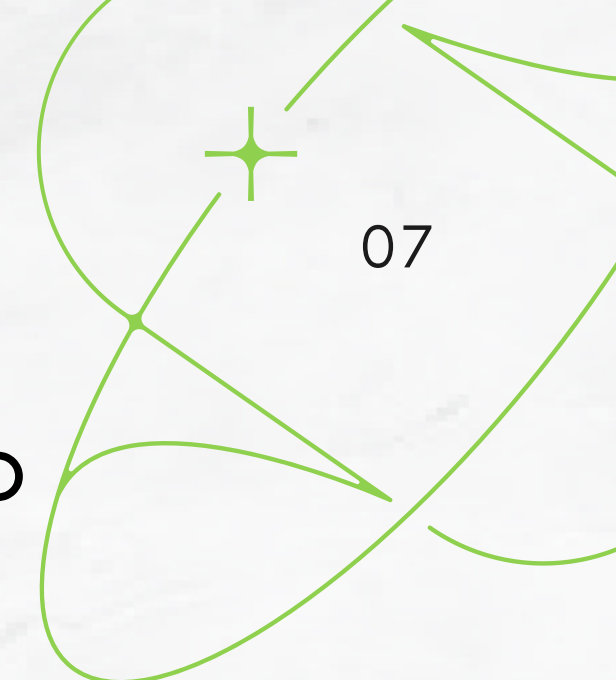
POSSIBILIDADE DE DISCRETIZAÇÃO DO ESPAÇO

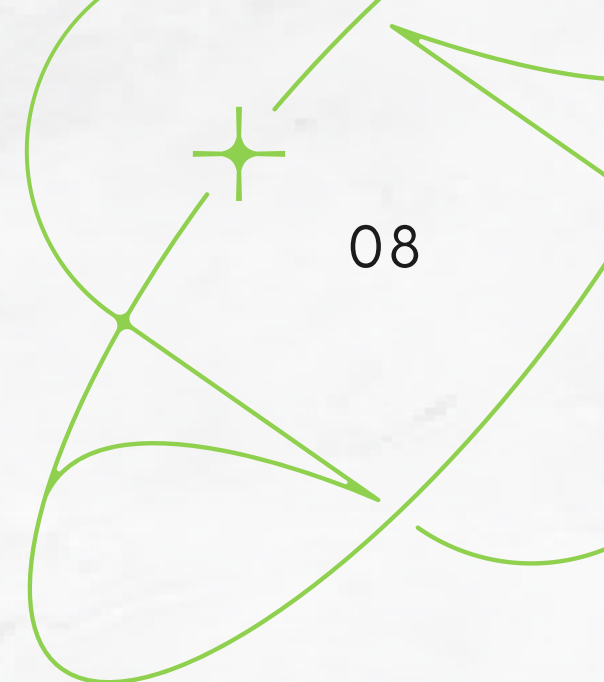
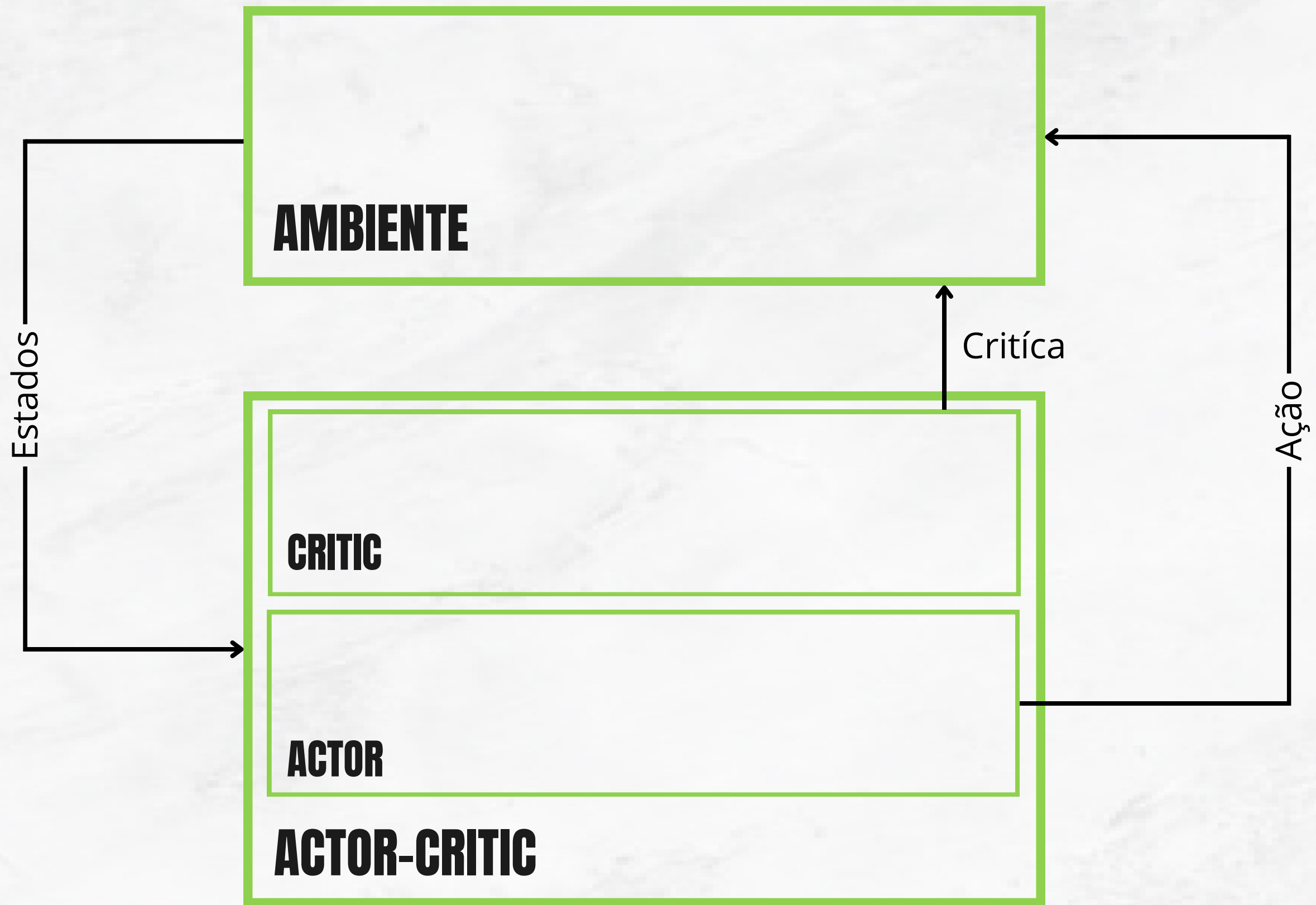
É possível discretizar o espaço de ação para se manter utilizando o DQL



TEMPO DE TREINAMENTO

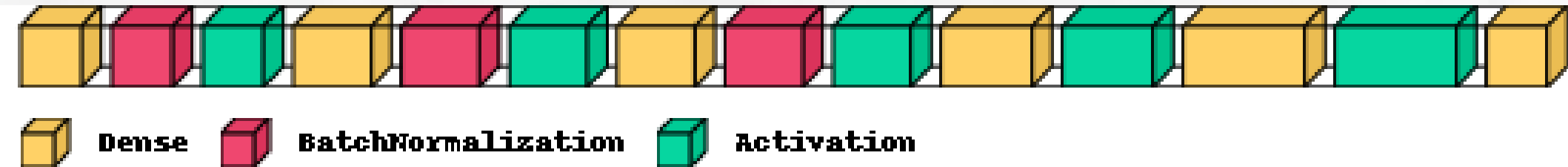
DQL passa a possuir u tempo de treinamento alto com a discretização do espaço



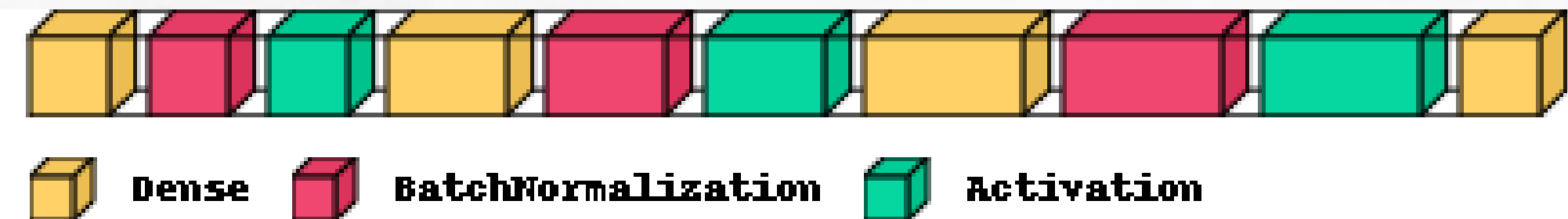


Redes

CRITIC



ACTOR



OUTRAS CONSIDERAÇÕES

◆ ADIÇÃO DE RUÍDO NA EXPLORAÇÃO

estratégias de exploração como epsilon greedy não são utilizadas.

◆ TREINAMENTO MAIS RÁPIDO

Tem a tendência de ser mais rápido que os algoritmos de policy gradient comuns.

◆ DEPENDENTE DO PONTO INICIAL

A seed influencia fortemente o resultado.

Algorithm 1 DDPG algorithm

Randomly initialize critic network $Q(s, a|\theta^Q)$ and actor $\mu(s|\theta^\mu)$ with weights θ^Q and θ^μ .
Initialize target network Q' and μ' with weights $\theta^{Q'} \leftarrow \theta^Q, \theta^{\mu'} \leftarrow \theta^\mu$
Initialize replay buffer R
for episode = 1, M **do**
 Initialize a random process \mathcal{N} for action exploration
 Receive initial observation state s_1
 for $t = 1, T$ **do**
 Select action $a_t = \mu(s_t|\theta^\mu) + \mathcal{N}_t$ according to the current policy and exploration noise
 Execute action a_t and observe reward r_t and observe new state s_{t+1}
 Store transition (s_t, a_t, r_t, s_{t+1}) in R
 Sample a random minibatch of N transitions (s_i, a_i, r_i, s_{i+1}) from R
 Set $y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$
 Update critic by minimizing the loss: $L = \frac{1}{N} \sum_i (y_i - Q(s_i, a_i|\theta^Q))^2$
 Update the actor policy using the sampled policy gradient:

$$\nabla_{\theta^\mu} J \approx \frac{1}{N} \sum_i \nabla_a Q(s, a|\theta^Q)|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu} \mu(s|\theta^\mu)|_{s_i}$$

Update the target networks:

$$\begin{aligned} \theta^{Q'} &\leftarrow \tau \theta^Q + (1 - \tau) \theta^{Q'} \\ \theta^{\mu'} &\leftarrow \tau \theta^\mu + (1 - \tau) \theta^{\mu'} \end{aligned}$$

end for
end for



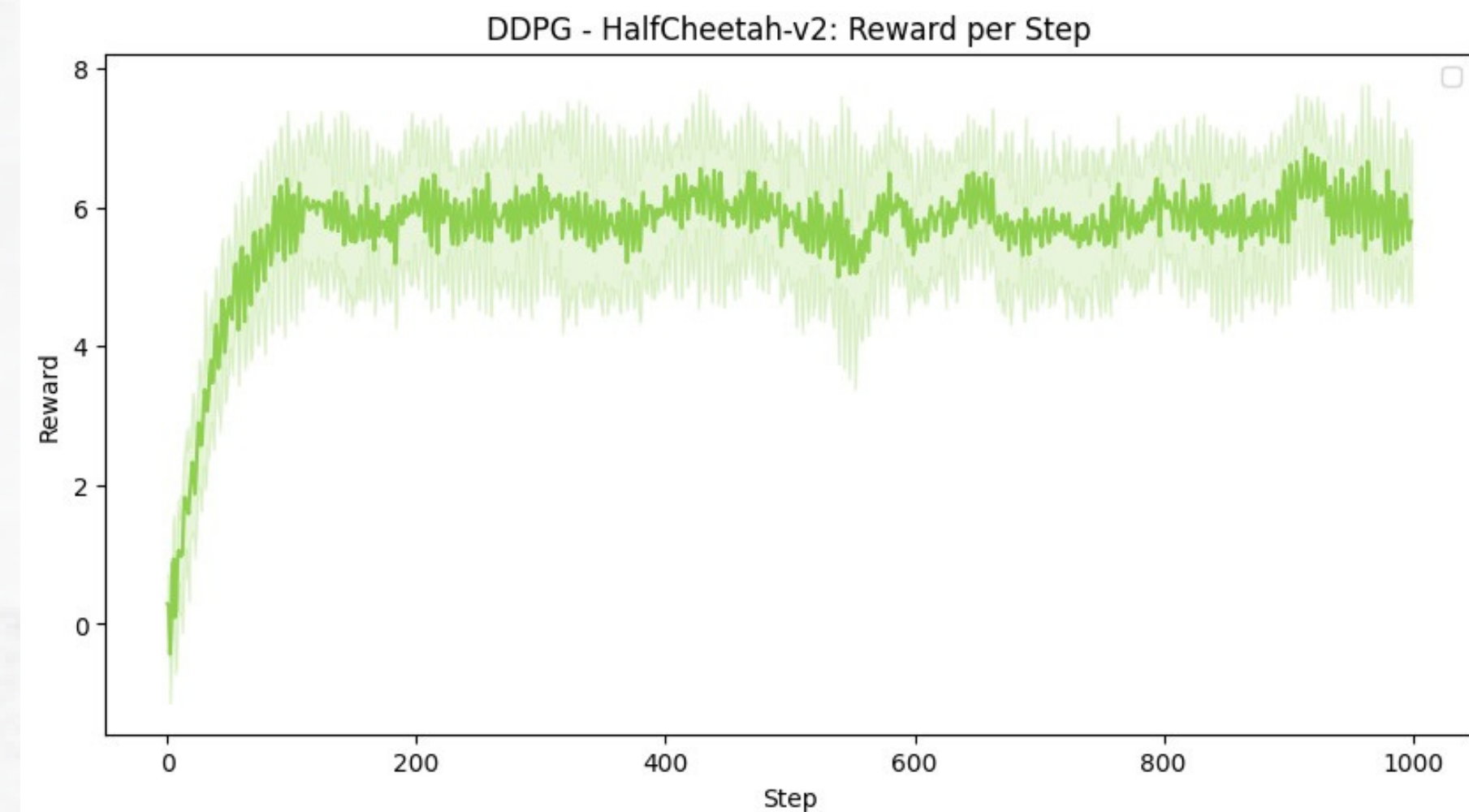
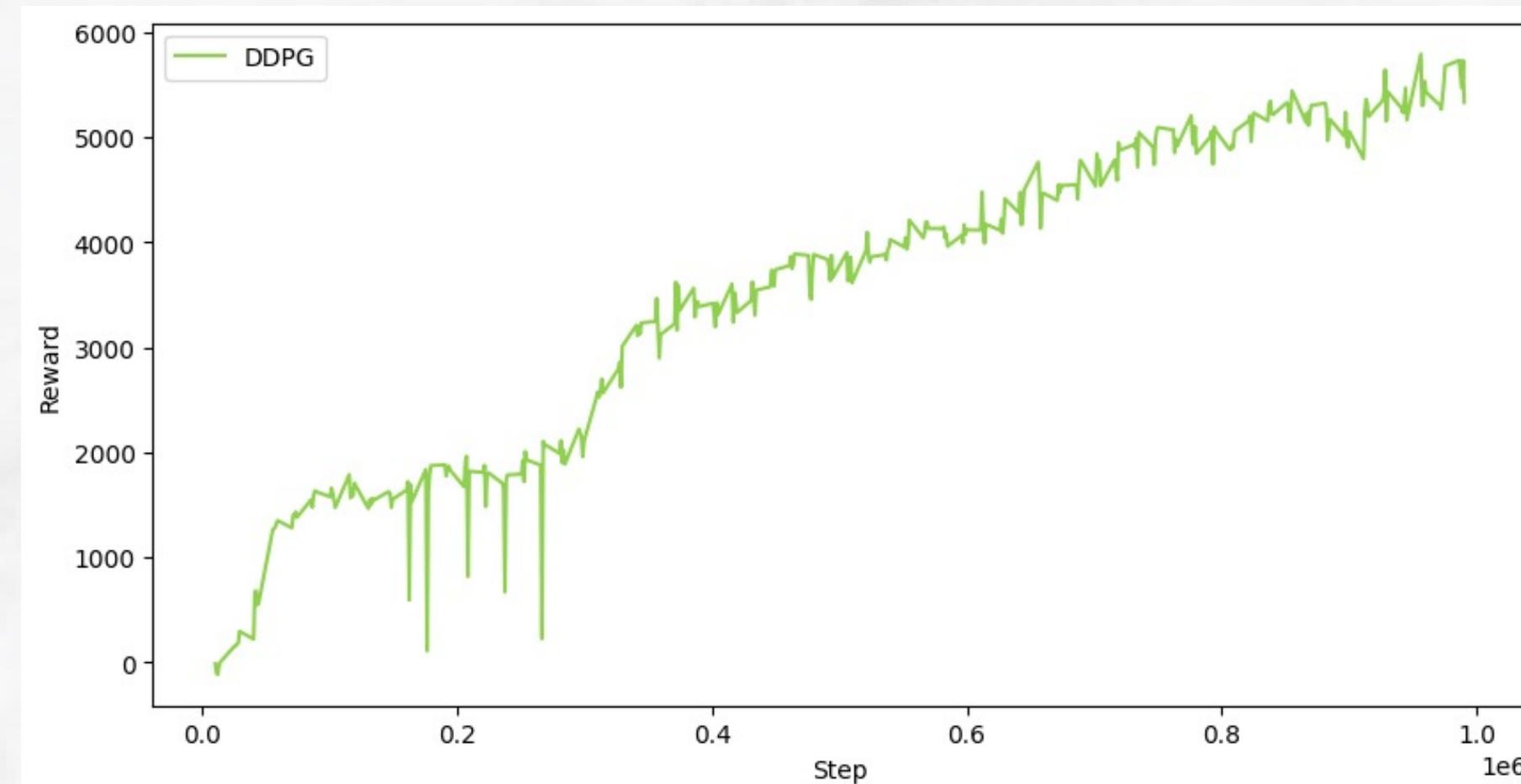
RESULTADOS

RECOMPENSAS

A recompensa média após 1000 steps é de aproximadamente uma distância de 5000 unidades.

Modelo ganha velocidade e depois estabiliza.

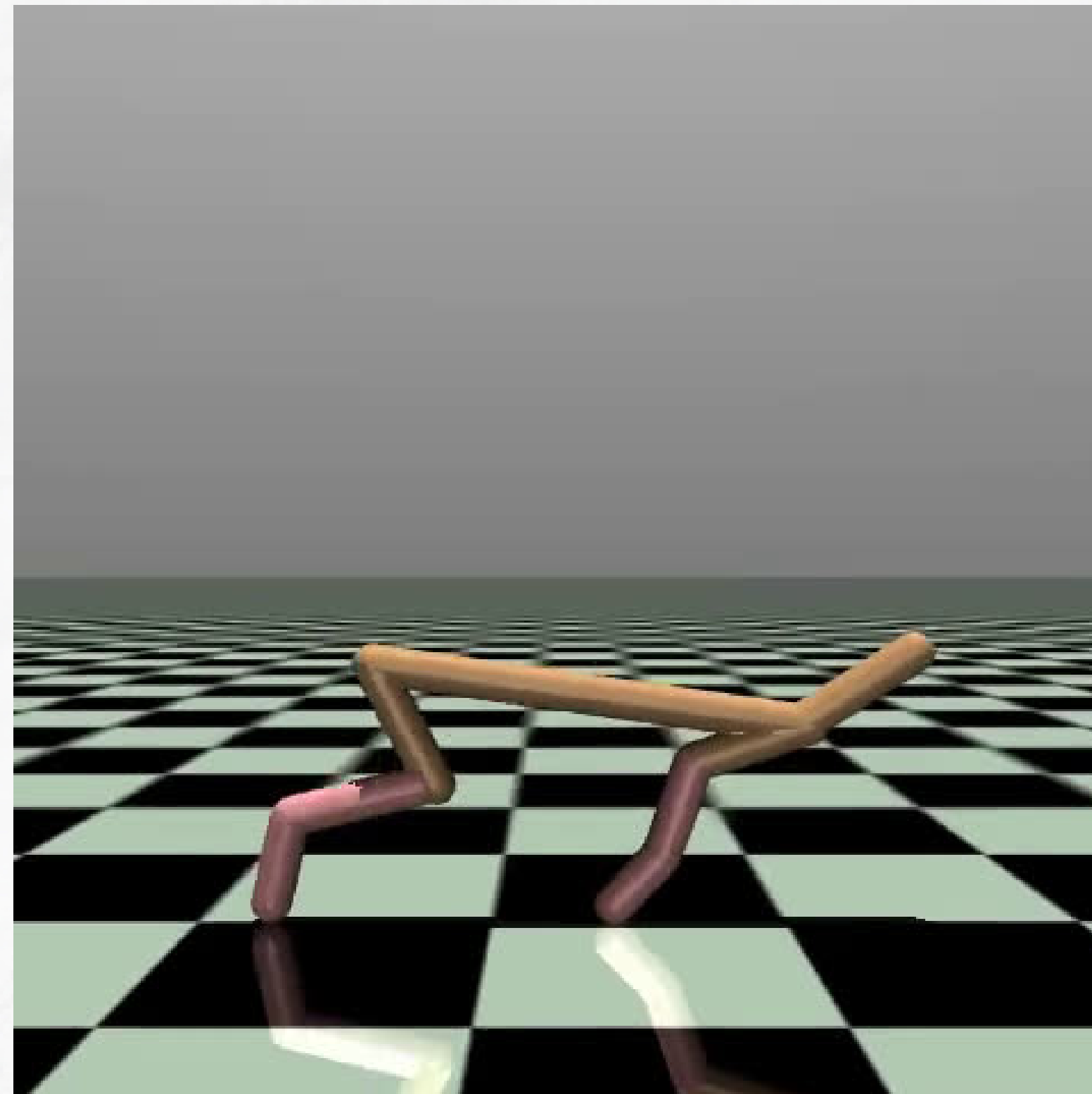
Treinamento durou cerca de 5 horas.



PIOR MODELO

O primeiro teste após uma noite inteira de treinamento e um computador quase queimado resultou nisso....

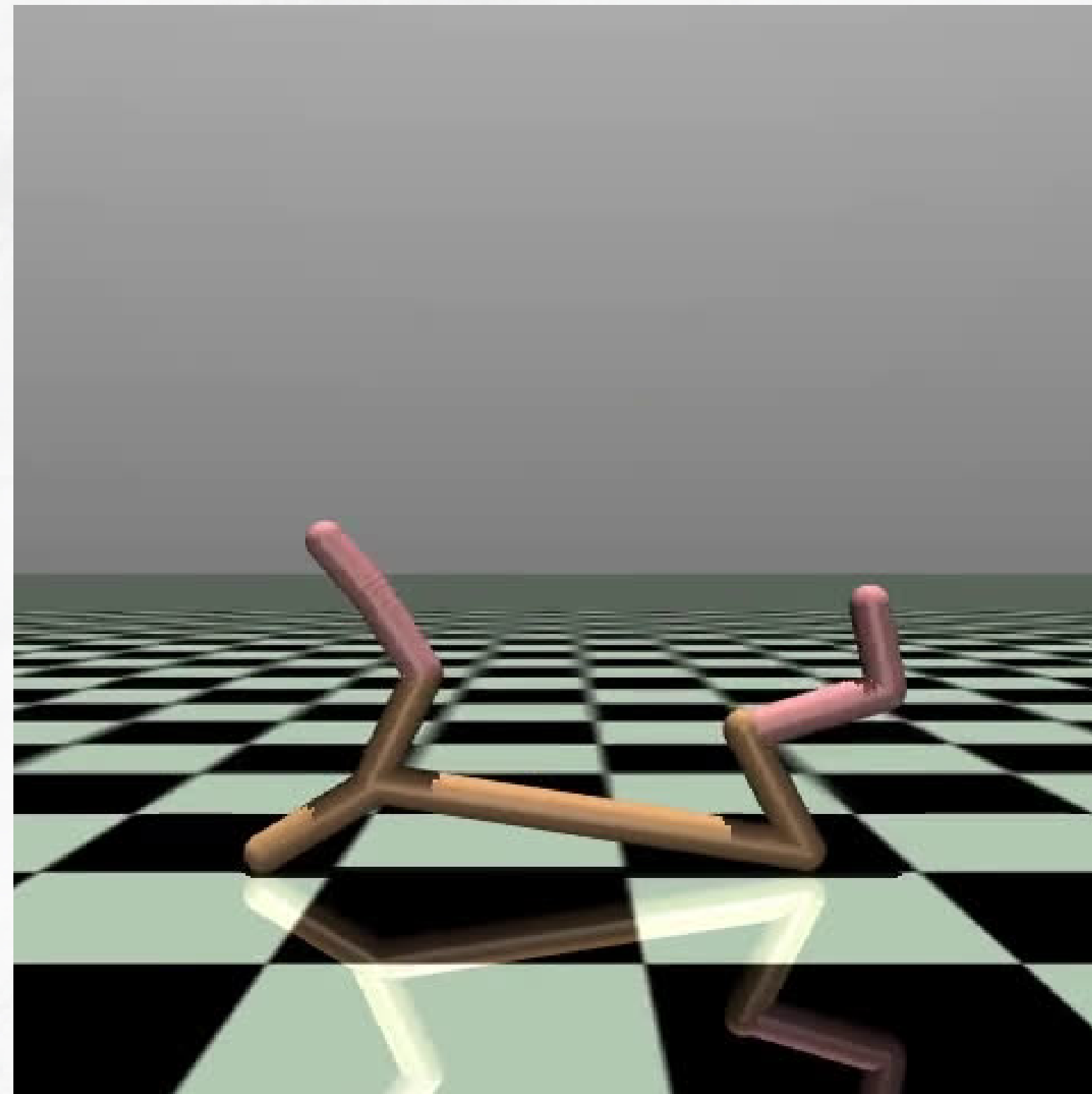
2000 steps por episódio,
200 episódios,
código mal otimizado.

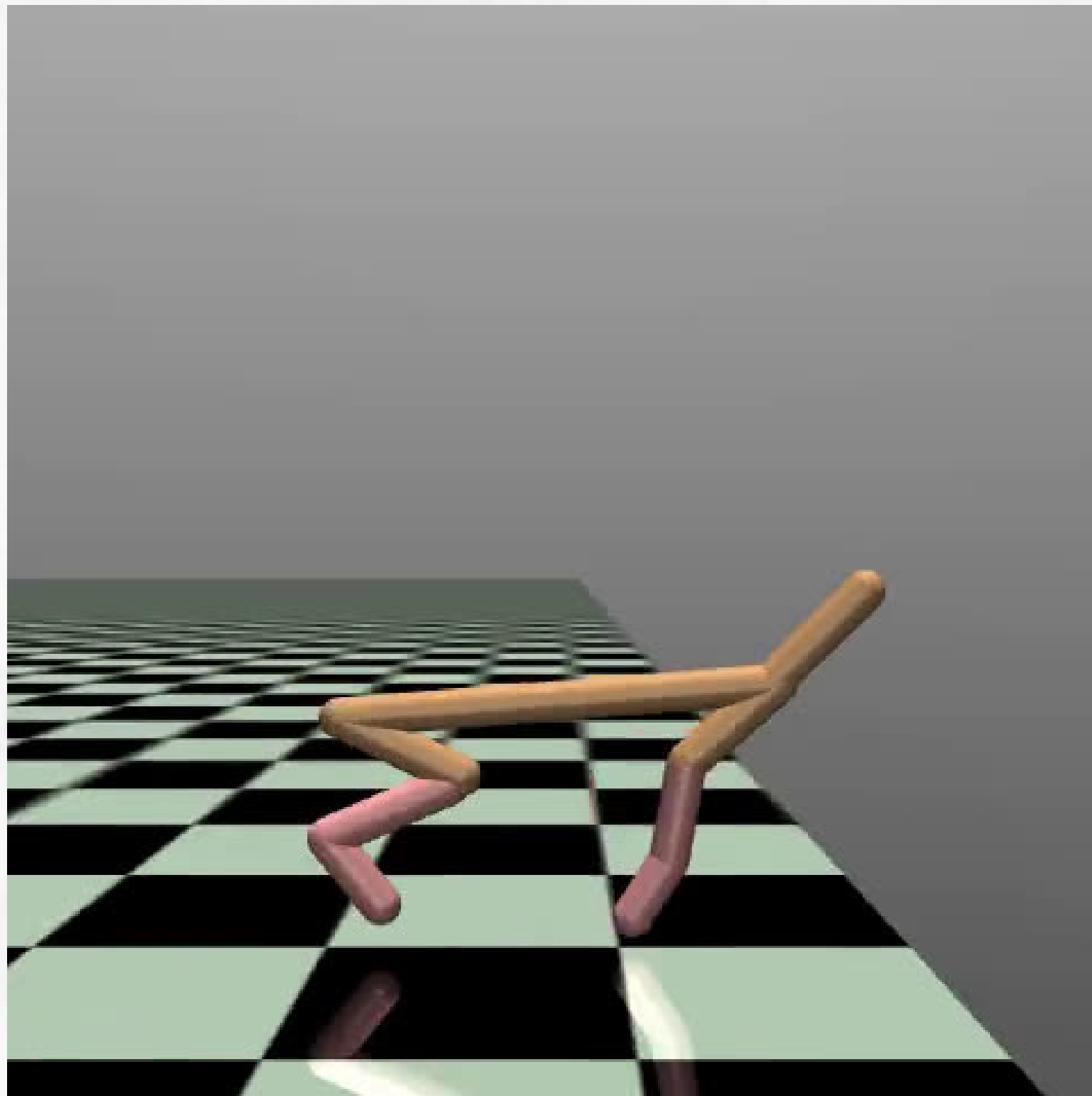


MODELO MÉDIO?

Difícil classificar se bom ou ruim, mas durante as tentativas de treinamento ele também aprendeu a andar assim...

200 steps por episódio,
20 episódios





MELHOR MODELO

Após pouco mais de cinco horas de treinamento e muitas tentativas falhas, alcançamos uma marca de 5662 de recompensa

1000 steps por episódio,
200 episódios

**OBRIGADO
PELA
ATENÇÃO**



Gustavo Sanches
Lucas Treuke

Bibliografia

Scott Fujimoto, Herke van Hoof, and David Meger. **Addressing function approximation error in actor-critic methods**, 2018

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. **Continuous control with deep reinforcement learning**, 2019.

Liang, James. **Training a Cheetah to run with Deep Reinforcement Learning**. Available at <https://towardsdatascience.com/training-a-cheetah-to-run-with-deep-reinforcement-learning-6dca2975443a>.

Silver, David, et al. "**Deterministic policy gradient algorithms**." International conference on machine learning. Pmlr, 2014.

