# RL study with Mujoco-Humanoid

251210 Term-Proj

Tom Juyeon Kim

# Contents

# 1. Operate Env.

- 운영체제 : Ubuntu 24.04.3 LTS
- GPU : RTX-4090
- Python –version : 3.11.14
- Requirements.txt 에 라이브러리 기재
- GitHub 리포에 프로젝트 업로드

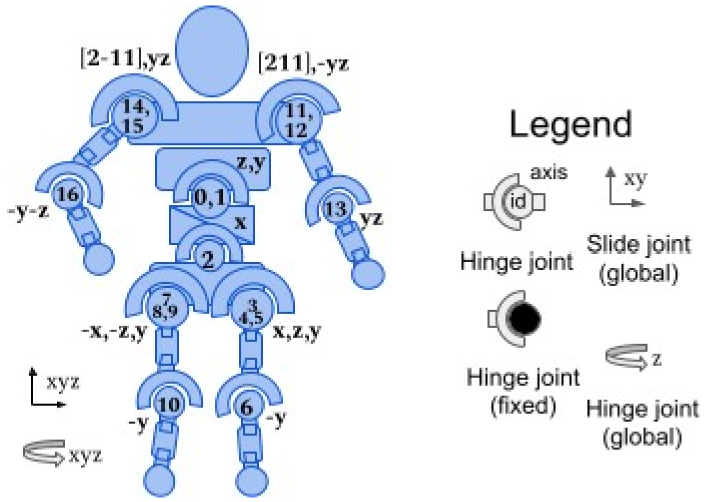
```
kimm-linux@kimm-linux-System-Product-Name:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 24.04.3 LTS
Release:        24.04
Codename:       noble
```

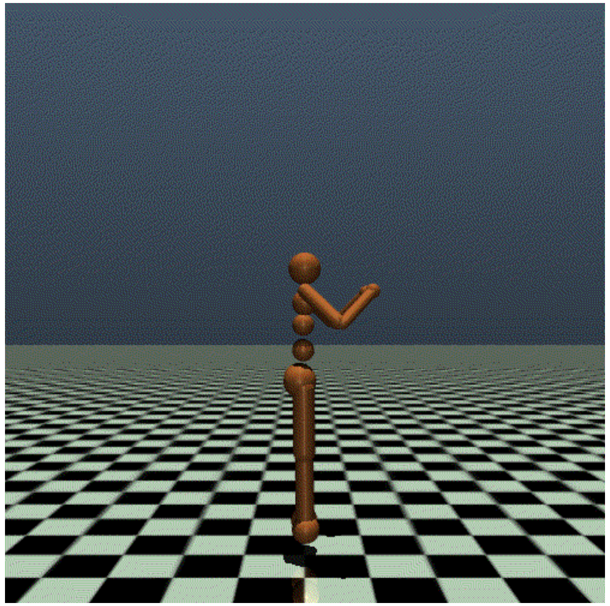
```
kimm-linux@kimm-linux-System-Product-Name:~$ nvidia-smi
Tue Dec  9 15:29:40 2025
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 580.95.05              Driver Version: 580.95.05      CUDA Version: 13.0     |
+-----------------------------------------+------------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id          Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |           Memory-Usage | GPU-Util  Compute M. |
|                                         |                        |               MIG M. |
|=========================================+========================+======================|
|   0  NVIDIA GeForce RTX 4090        Off |   00000000:41:00.0  On |                  Off |
|  0%   57C    P2             86W /  480W |    1334MiB /  24564MiB |      4%      Default |
|                                         |                        |                  N/A |
+-----------------------------------------+------------------------+----------------------+

+-----------------------------------------------------------------------------------------+
| Processes:                                                                              |
|  GPU   GI   CI              PID   Type   Process name                        GPU Memory |
|        ID   ID                                                               Usage      |
|=========================================================================================|
|    0   N/A  N/A          372276      G   /usr/lib/xorg/Xorg                       296MiB |
|    0   N/A  N/A          373858      G   /usr/bin/gnome-shell                      34MiB |
|    0   N/A  N/A          374641      G   /usr/share/code/code                      68MiB |
|    0   N/A  N/A          377821      G   ...rack-uuid=3190708988185955192          88MiB |
|    0   N/A  N/A          508339      C   ...edRL/termPro/.venv/bin/python         792MiB |
+-----------------------------------------------------------------------------------------+
```

# 2. Mujoco-Humanoid

- **Action Space : 제어출력 . 17 개의 조인트의 토크값**
- **Observation Space: 13 개의 파츠에 대한 속성값 (348)**
  **( 위치 , 각도 , 속도 , 각속도 )**

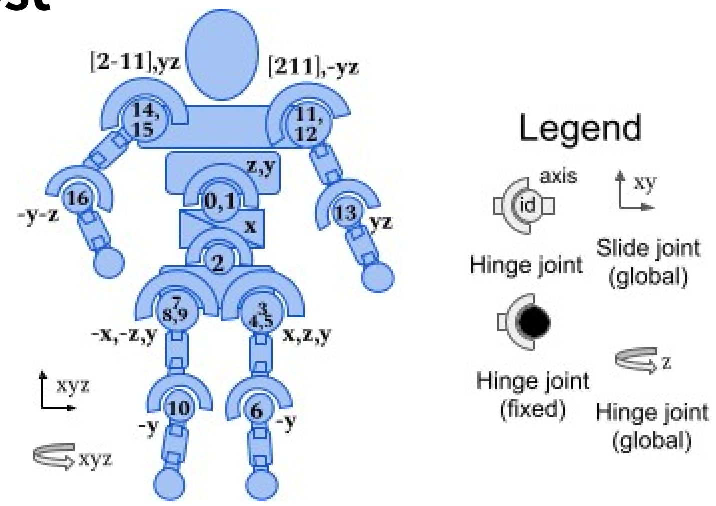


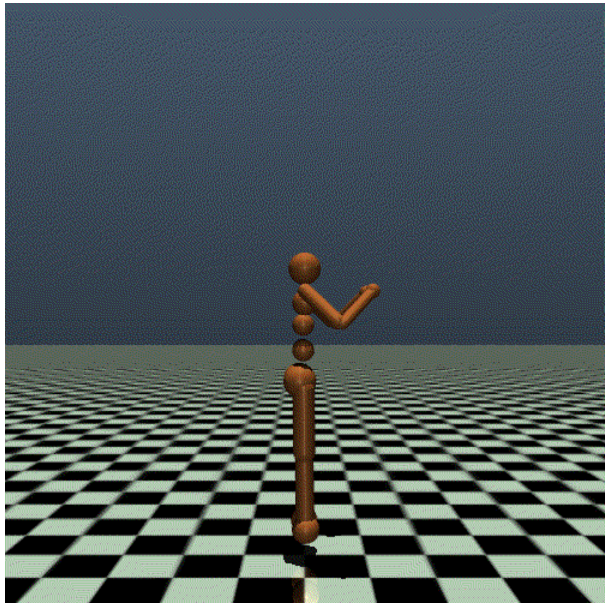



| | |
|---|---|
| Action Space | Box(-.4 .4 (17,), float32) |
| Observation Space | Box(-inf, inf, (348,), ) |
| Reward | Healthy + forward – ctrl - contact |

# 2. Mujoco-Humanoid

- **Reward: healthy_r + forward_r – ctrl_cost – contact_cost**
  - **healthy_reward : 매 스텝 서있으면 받는 보상**
  - **forward_reward : 앞으로 나아가면 받는 보상**
  - **ctrl_cost : 제어출력이 너무 강하면 받는 비용**
  - **contact_cost : 지면에 부드럽게 발을 딛도록 하는 비용**

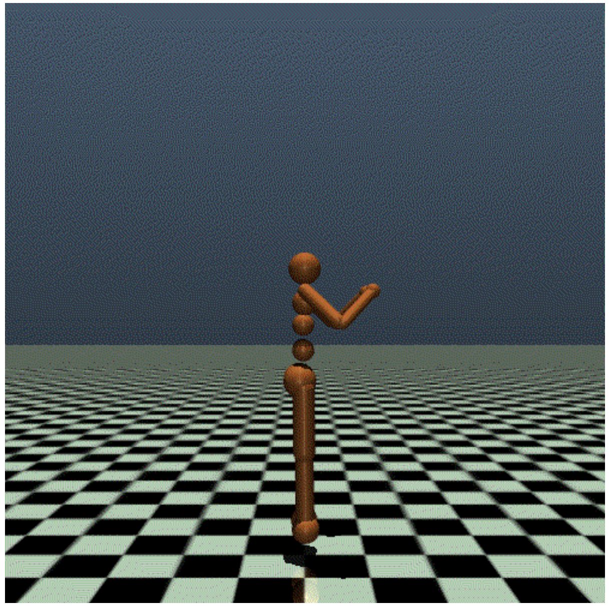


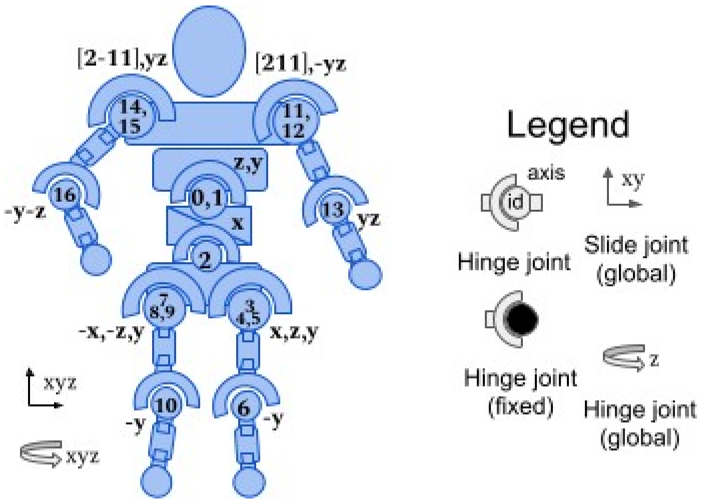



| | |
|---|---|
| Action Space | Box(-.4 .4 (17,), float32) |
| Observation Space | Box(-inf, inf, (348,), ) |
| Reward | Healthy + forward – ctrl - contact |

# 2. Mujoco-Humanoid

- **Episode End**
  - **Termination : 휴머노이드가 쓰러지면 중단**
  - **Truncation : 휴머노이드가 1000 timestep 을 버티면 중단**



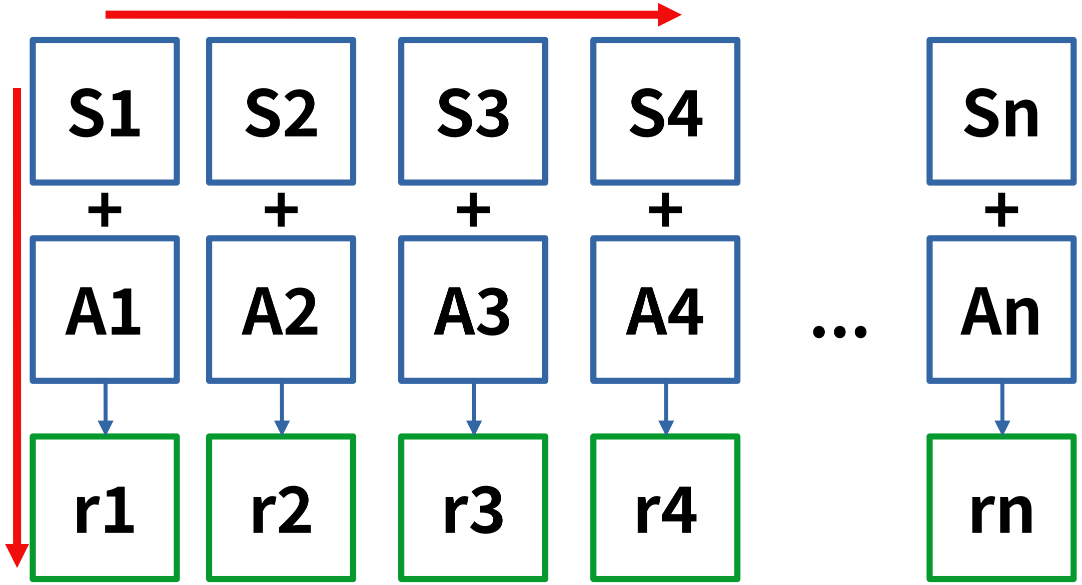



| | |
|---|---|
| Action Space | Box(-.4 .4 (17,), float32) |
| Observation Space | Box(-inf, inf, (348,), ) |
| Reward | Healthy + forward – ctrl - contact |

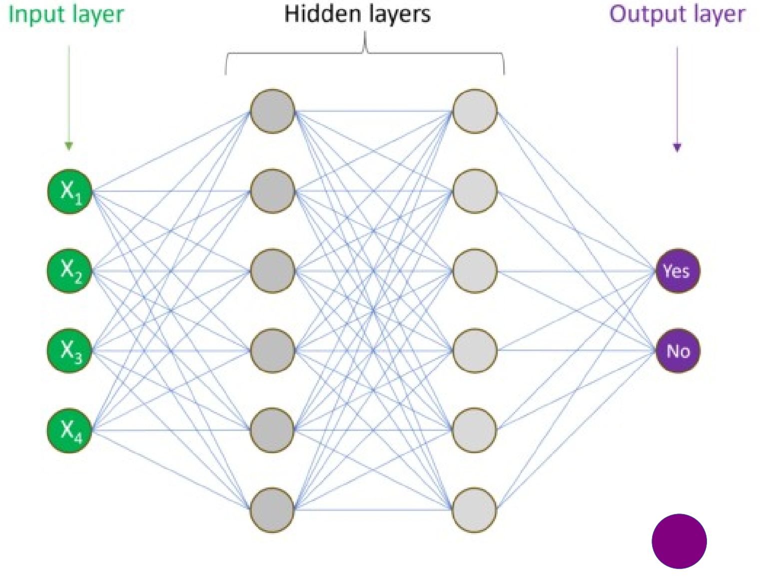
# 3. PPO Algorithm

- 입력을 상태값 , 출력을 A[17,], V[1,] 으로 하는 네트워크
- SB3.PPO 라이브러리는 [, 64, 64,] default
- V[1,] Critic 이 예측한 기대 보상

| | |
|---|---|
| Action Space | Box(-.4 .4 (17,), float32) |
| Observation Space | Box(-inf, inf, (348,), ) |
| Reward | Healthy + forward – ctrl - contact |




S[348,]




A[17,]

V[1,]

# 3. PPO Algorithm

- **PPO 에서는 환경이 준 보상 (reward) 로  return 을 만들고 , 그  return 으로  Advantage 를 계산해서 학습**
- **순간적인 보상 (reward) 를 모아 에피소드 전체 간에 누적 보상인  return 을 활용해 학습**
- **Advantage : Act 가 기댓값보다 얼마나 잘했는지 / 못했는지**

$$A_t = R_t - V_\phi(s_t)$$

| | | | | | |
|---|---|---|---|---|---|
| **r1** | **r2** | **r3** | **r4** | **…** | **rn** |
| **R1** | **R2** | **R3** | **R4** | **…** | **Rn** |

$$R_t = r_t + \gamma R_{t+1}$$

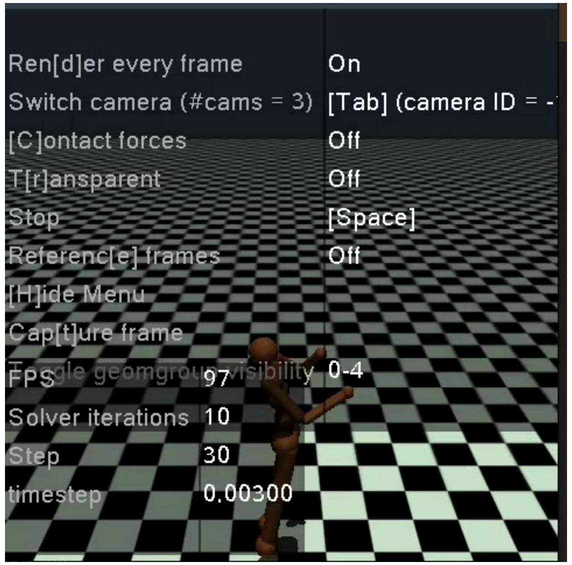
| | |
|---|---|
| States | s1, s2, s3, s4, … , sn |
| Actions | a1, a2, a3, a4, … , an |
| Rewards | r1, r2, r3, r4, … , rn |
| Discounted Rewards | R1, R2, R3, R4, …, Rn |
| Values | V(s1), V(s2), V(s3), …, V(sn) |
| Advantage | A1, A2, A3, A4, …, An |

# 4. Train

- 파라미터가 DL 과 상이하게 사용됨에 유의
  - n_envs : 병렬 환경 개수
  - n_steps : rollout 길이 ( 얼마나 모으고 학습시작할지 )
  - n_epochs : rollout 데이터를 가지고 몇 번 SGD 반복할지
  - batch_size : 1 iter(rollout) 데이터의 양
    - 이 데이터를 가지고 (n_epochs) 번 학습
  - total_timesteps : 이를 5M step 반복
  - net_arch : 은닉층의 노드 갯수
- 10M step 학습해도 잘 서있지 못함

```
n_envs = 32
n_steps = 1024
n_epochs = 10
batch_size = n_envs * n_steps
total_timesteps = 5_000_000
net_arch=[256, 256]
```

hyper_param




10Mstep_


5Mstep_

# 4. Train

- 목표 : ep_len_mean=1000 이상 서있기
- 학습 파라미터 수정
  1) Gamma= 0.9 → 0.9999 로 수정 ( 미래보상에 집중 )
  2) 네트워크 크기 [64,64] → [256, 256] ( 정보 손실 방지 )
  3) log_std_init = -2.0 → -1.0 ( 탐색을 늘림 )
  4) 보상함수의 healthy_reward 올림 ( 서있는것에 보상 )

```
n_envs = 32
n_steps = 2048
n_epochs = 10
batch_size = n_envs * n_steps
total_timesteps = 5_000_000
net_arch=[256, 256]

log_std_init=-0.5
gamma=0.9999
target_kl=0.01
LR = 5e-4
learning_rate=cosine_schedule(LR)
healthy_reward=5.0
```

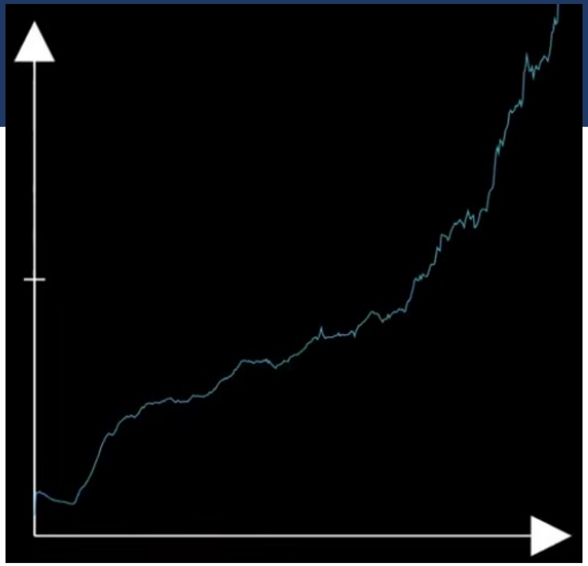
hyper_param

```
Timesteps: 3648000 | Best ep_len_mean: 85.48 | Current ep_len_mean: 82.20
----------------------------------------------
| rollout/                |                  |
|    ep_len_mean          | 82               |
|    ep_rew_mean          | 420              |
| time/                   |                  |
|    fps                  | 2214             |
|    iterations           | 95               |
|    time_elapsed         | 1405             |
|    total_timesteps      | 3112960          |
| train/                  |                  |
|    approx_kl            | 0.039049152      |
|    clip_fraction        | 0.226            |
|    clip_range           | 0.2              |
|    entropy_loss         | -24.8            |
|    explained_variance   | 0.912            |
|    learning_rate        | 0.001            |
|    loss                 | 4.56             |
|    n_updates            | 2470             |
|    policy_gradient_loss | -0.0232          |
|    std                  | 0.134            |
|    value_loss           | 10.1             |
----------------------------------------------
```

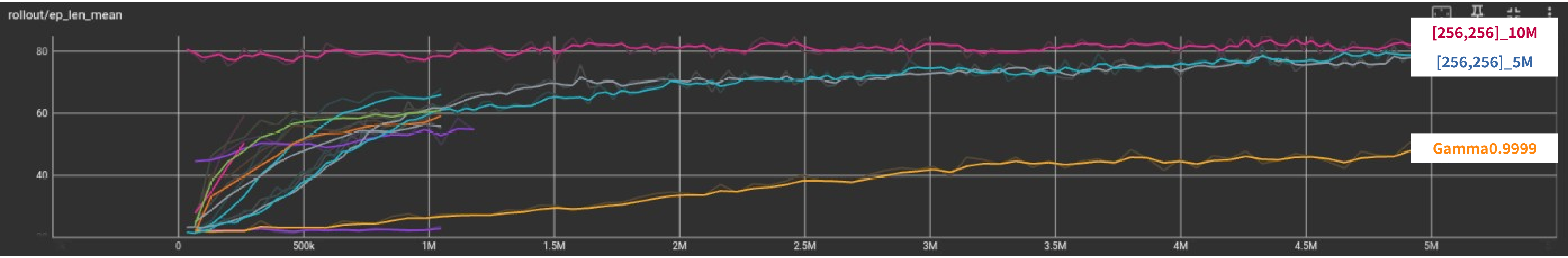
logs

# 4. Train

- 학습결과 : 모두 얼마 지나지 않아 넘어짐
- 두 가지 가정을 하게됨
  - 하이퍼파라미터 설정이 잘못됨
  - 학습과정세팅 자체가 잘못됨



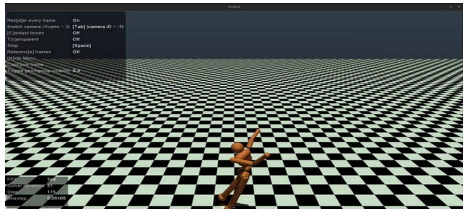
Desired (2Mstep)




My experiment

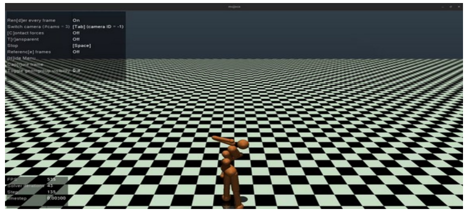
# 5. inference

- **계속 시도중이나 가만히 서있질 못함 ...**

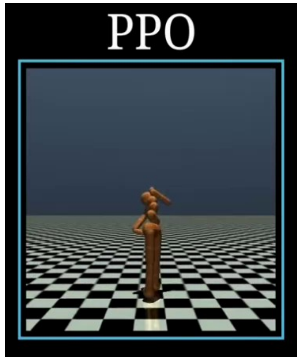


[256,256]_cossche_0.0005LR_

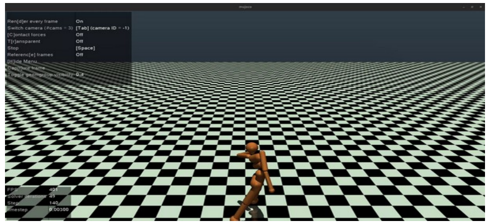
0.9999gamma_5Mstep



ep_len focused model




**Desired**



[256,256]_5Mstep

[256,256]_5Mstep + -3.0exp_1M

# 6. Ref

- 교수님 Base code (pendulum_ppo.py)
- Rainforcement Learning behind Humanoid Robot Explained (https://www.youtube.com/watch?v=QwJcF08hfs8&t=29s)
- Gym docu – Humanoid (https://gymnasium.farama.org/v0.27.0/environments/mujoco/humanoid/)