

# Bipedal Walker

---

This is how to teach a bot to walk on two feet.

Bipedal Walker is an [OpenAI Gym](#) environment where an agent learns to control a bipedal 2D character to reach the end of an obstacle course. What makes this challenging is that the agent only receives limbs coordinates along with the Lidar information. The agent has to learn to balance, walk, run, jump on its own without any human intervention.

There are two versions:

- Normal, with slightly uneven terrain.
- Hardcore, with ladders, stumps, pitfalls.

To solve the normal version, you need to get 300 points in 1600 time steps. To solve the hardcore version, you need 300 points in 2000 time steps.

The code in this repo solves Bipedal Walker V3 in normal mode (`hardcore=False`).

## Prerequisites and Recommended Tools

- [Conda](#) installed on your system
- Visual Studio Code

## Setup Instructions

Follow the steps below to set up your environment and install the necessary dependencies.

### 1. Create a Conda Environment

First, create a new Conda environment with Python 3.11:

```
conda create -n gym-env python=3.11 -y
```

### 2. Activate the newly created environment

```
conda activate gym-env
```

### 3. Install the required packages

```
conda install \
    swig gymnasium-box2d \
```

```
stable-baselines3 tensorboard \  
plotly scikit-learn optuna -y
```

## Project Structure

- `bipedal_train.py`: Train the agent with PPO default parameters.
- `bipedal_eval.py`: Evaluate the agent using a previously saved model.
- `optuna_ppo.py`: Auto-tune PPO using [OPTUNA](#).
- `optuna_ddpg.py`: Auto-tune DDPG using [OPTUNA](#).
- `bipedal_wandb_ppo.py`: Train / evaluate the agent with tuned hyper-parameters + [W&B](#) integration.
- `bipedal_wandb_ddpg.py`: Train / evaluate the agent with tuned hyper-parameters + [W&B](#) integration.
- `record_video.py`: Record a video of the trained agent.

### 4. Train the PPO model using customized environment with new angular velocity configs and new reward scheme

New environment customized using different parameters is saved in `/custom_env/bipedal_walker.py`.

For new angular velocity at joints and hip, changes were made at line 34 and 35

```
SPEED_HIP = 8 # default value: 4  
SPEED_KNEE = 12 # default value: 6
```

For new reward scheme, new equation was added 575

```
# reward will be given to velocity  
if vel.x > 5.0:  
    # normalize it to scale[-1, 1]  
    reward += 0.08 * 0.3 * vel.x * (VIEWPORT_W / SCALE) / FPS
```

```
python3 bipedal_custom.py
```

After the training, to evaluate model's performance run:

```
python3 bipedal_eval_custom.py
```

## Tensorboard Visualization

```
tensorboard --logdir=./training/logs
```

