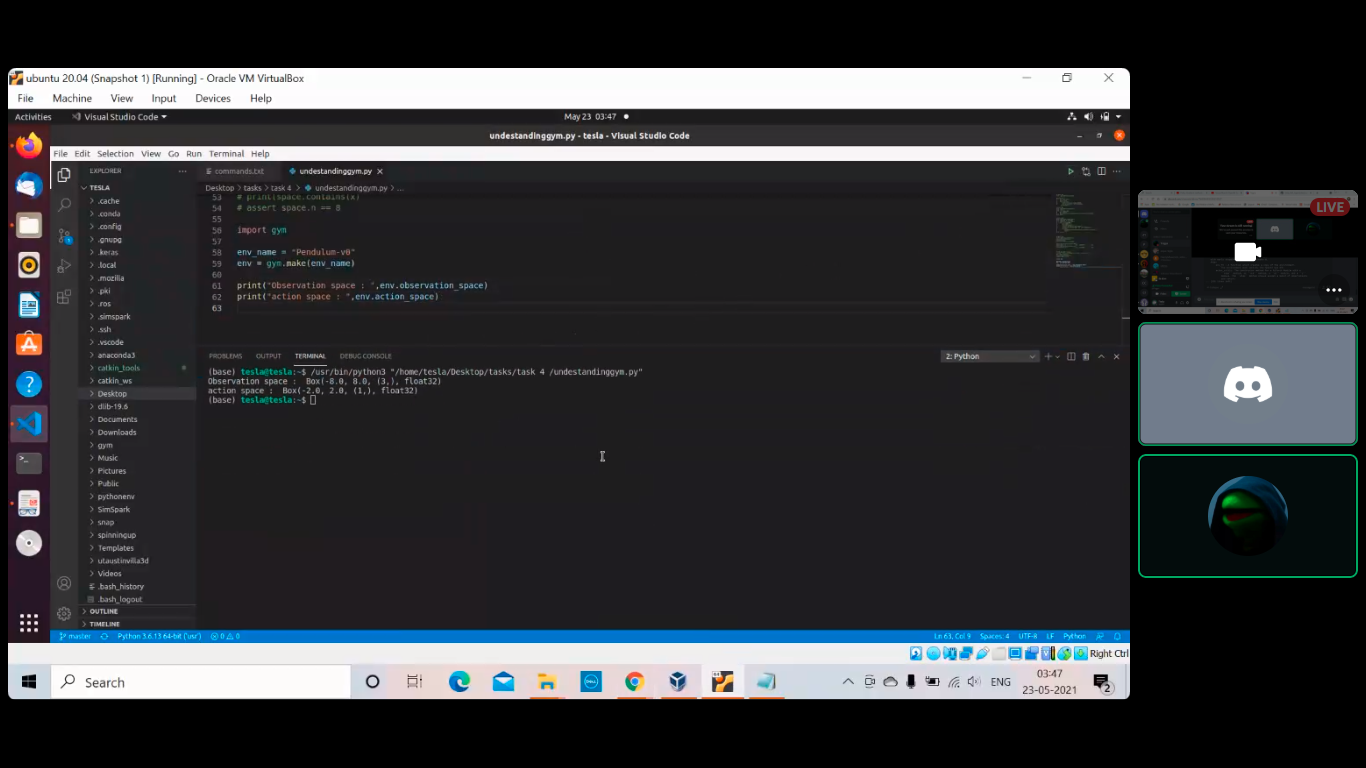
***TASK 4***

**1.You should inspect the observations dimensions and action dimensions by printing env.observation\_space and env.action\_space**

Here**,** action\_space & observation\_space describes what is the valid format for that particular env to work on with, this can be easily seen through following output :



This kind of output shows that the observation and action spaces are continuous. The first two values in action\_space i.e. -2.0 and 2.0 denote the minimum and maximum values respectively. The third value shows the number of parameter present in action\_space like in this case (1,) represents the only parameter Joint Effort (torque) while float32 tells its datatype. Same goes for observation\_space.

**2. Look at the environment definition and understand what the observations mean: gym/gym/envs/classic\_control/pendulum.py**

**○ in particular, look at the step function, line 32 and where the observations come from (get\_obs function, line 57)**

The step function in general returns four values: observation, reward, done, info. In this case, the get\_obs function returns the observation of environment as a numpy array of three elements (cos(theta), sin(theta), theta dot). The step function updates the values of theta and theta dot (arguments of get\_obs). The new theta\_dot is calculated using a simple physics equation i.e. torque = (moment of inertia)\* alpha (where alpha = (new\_theta\_dot-theta\_dot)/dt & torque = (mg\*l)/2 + action)

Further, new\_theta is calculated using appropriate equation involved.

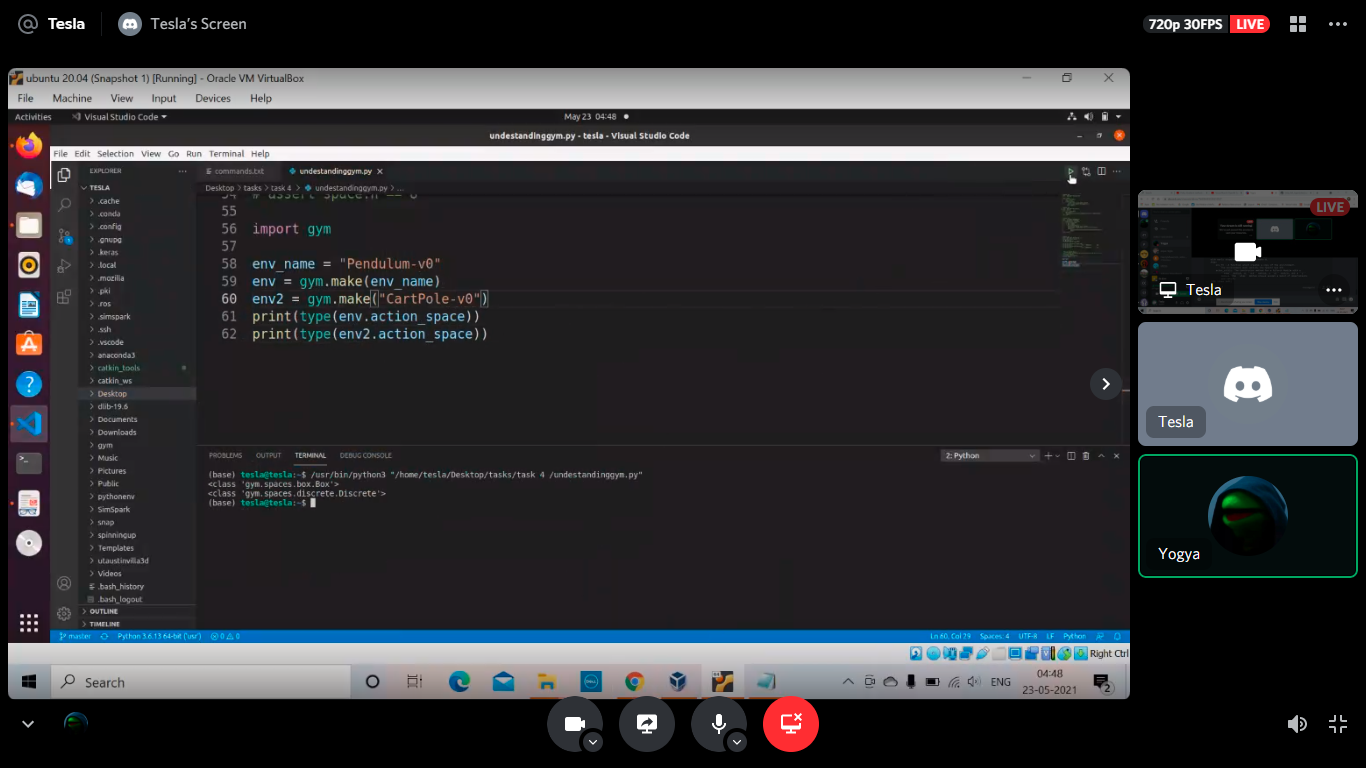
**3. What do the different components in the observations stand for? What’s self.state[1]/thetadot in the environment file, what does it mean?**

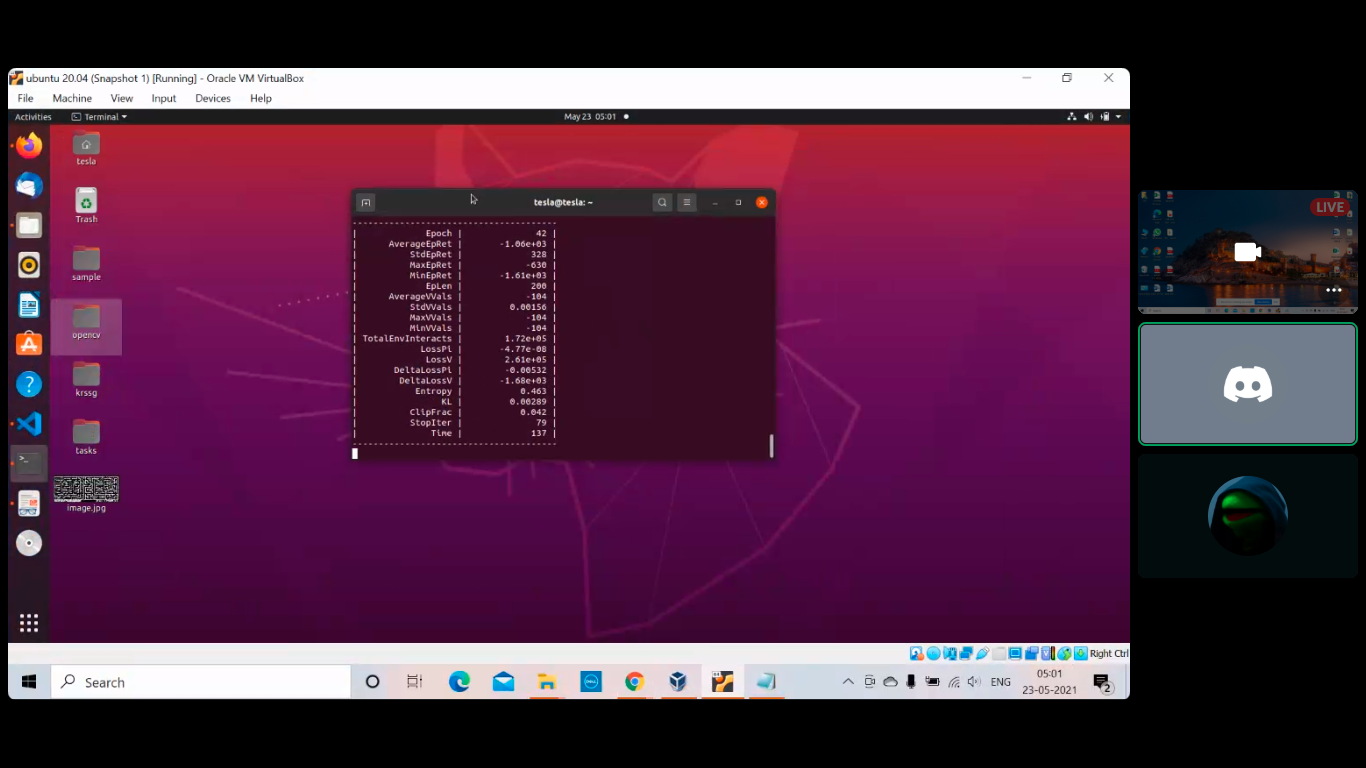
The ‘observations’ are referred to as ‘state’ in python script given which has two components theta and theta\_dot. It is only when get\_obs is called; cosine and sine of theta are calculated and displayed along with theta\_dot

The self.state[1] i.e. theta\_dot is the angular velocity of the pendulum about the pivot.

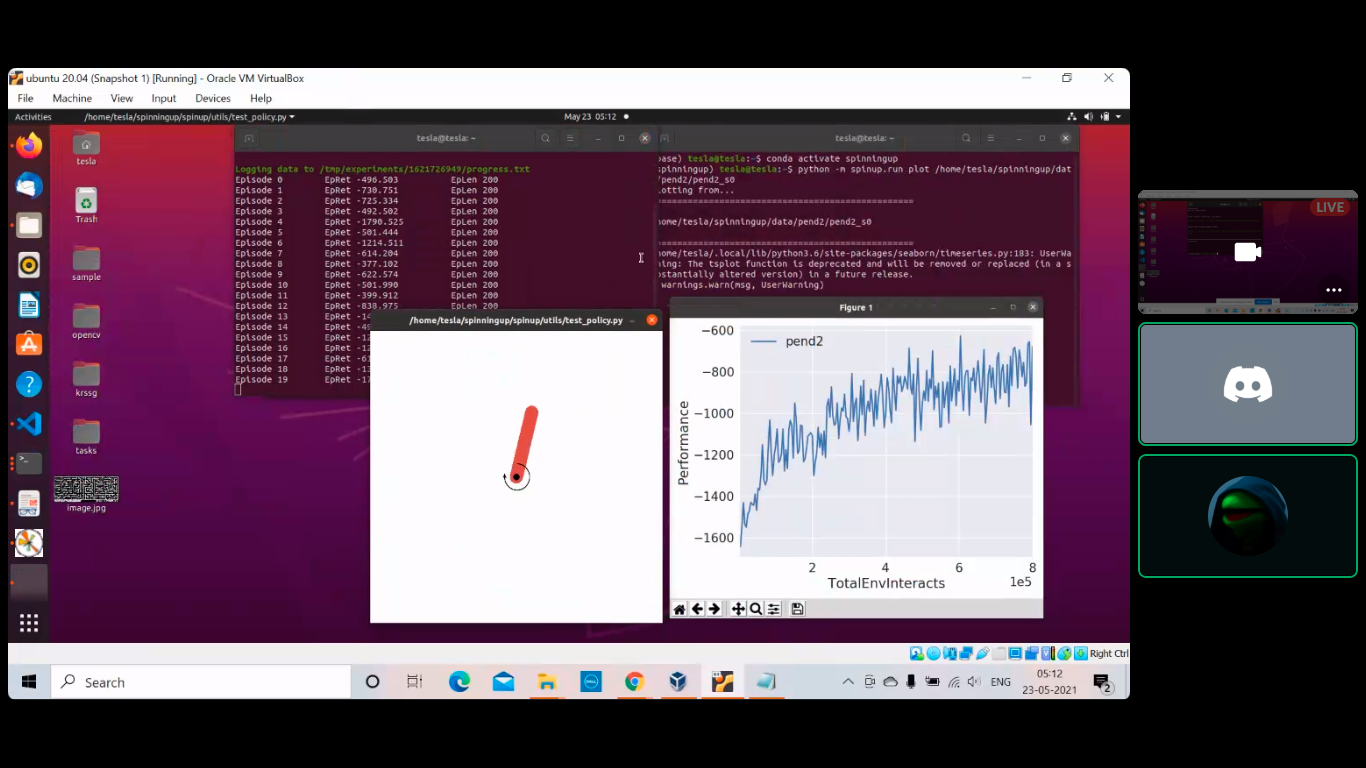
**4. Is this discrete action (“turn left/right”) or continuous action (“turn left/right at [0-1]x speed”)?**

For the Pendulum environment, the action of turning left/right is a continuous action as depicted in the output. For CartPole to turn left/right, it is a discrete action (i.e. 0 for left and 1 for right). For continuous actions, value is stored in a float data type thus, the values lies within a given range and can take infinite values rather than just finite (as in case of discrete).





***INSTANT OF PPO ALGORITH RUNNING AT EPOCH=42***



***Analysis after successfully training the agent using PPO Algorithm (EPOCHS =200)***

**5. The reward calculation (“costs”, line 42), has different components. What do they stand for and how are they weighted?**

In essence, the goal is to aim at angle zero with least rotational velocity and least effort. Thus, the reward calculation contains three components, i.e. theta, theta\_dot, and action. These components are weighted according to their preference i.e. first the theta should approach zero followed by theta\_dot, as for other values of theta, if theta\_dot approaches zero then, it is not desirable. Similar is the case for action (torque) which affects alpha and it is comparatively less weighted.

Reward = -[(theta^2) + 0.1\*(theta\_dot^2) + 0.001\*(action^2)]

As theta belongs from –pi to pi,

The lowest reward is: - (pi^2 + 0.1\**8^2 + 0.001\**2^2) = -16.2736044

The highest reward is : 0