RV.Lab
Robot & Virtual Reality

KONKUK University
Mechanical Engineering
Autonomous Robotics

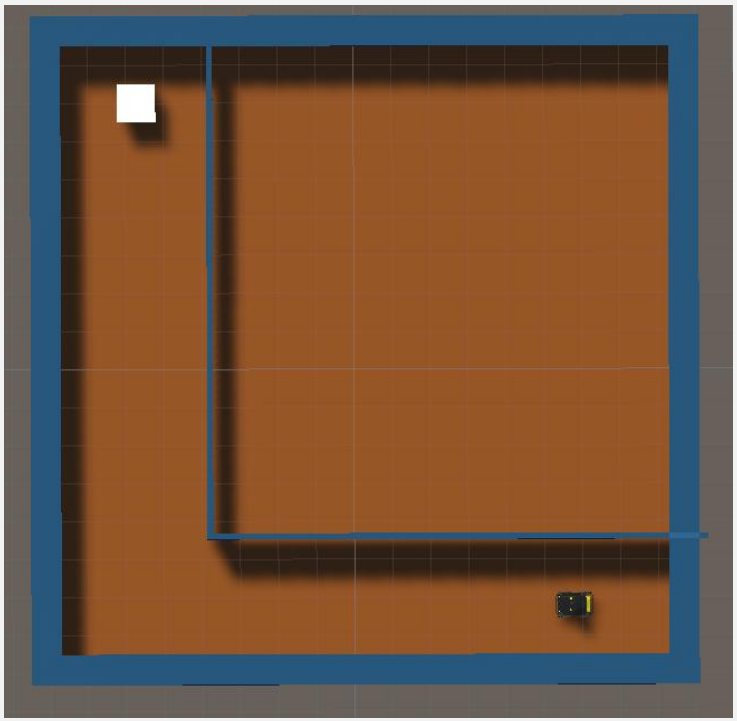
연구 진행사항(husky)

전민경

KU 건국대학교 KONKUK UNIV.

학습 환경 – simple_maze

환경 구성

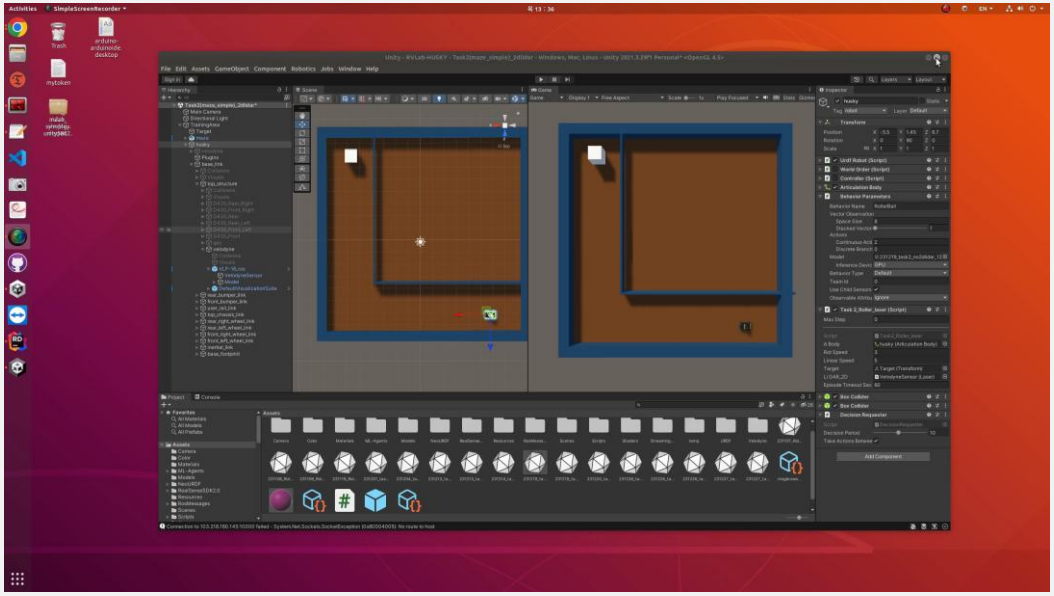


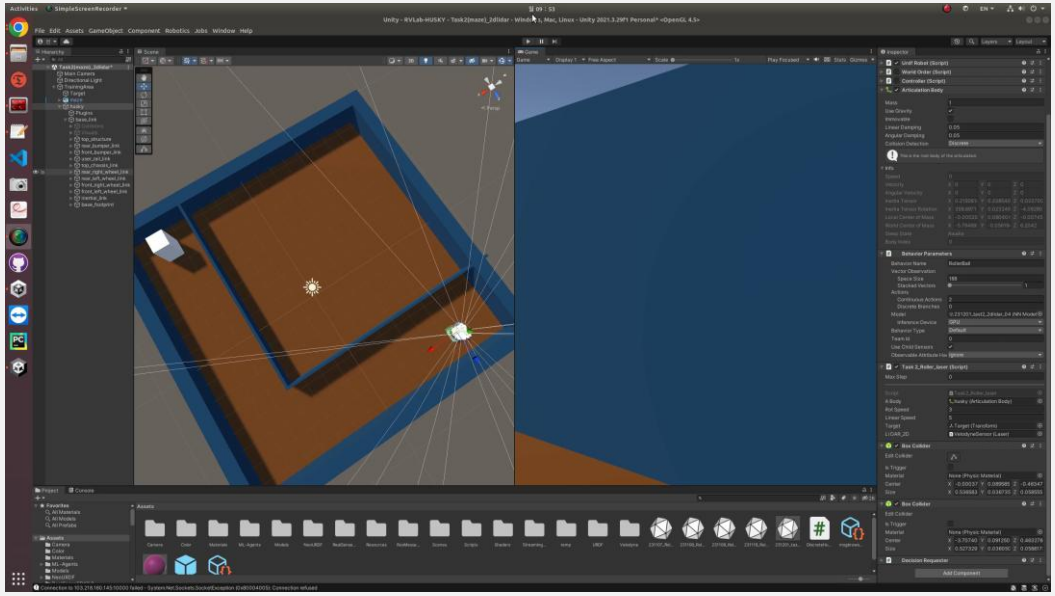
<simple_maze>

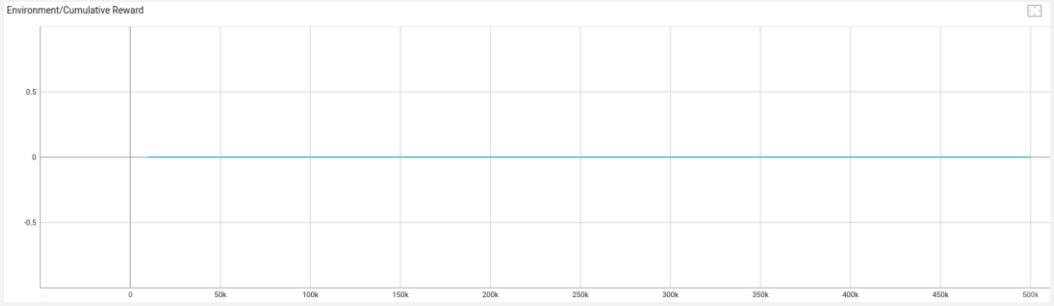
기본 설정	
Step	500,000
reward	• Target과 agent의 거리 < 1.42: +1 • Wall, Partition 충돌: 에피소드 종료
Observation	• Target과 agent의 위치(x, y, z 좌표) • Agent의 x, z축 속도 → 총 8개의 관측 벡터

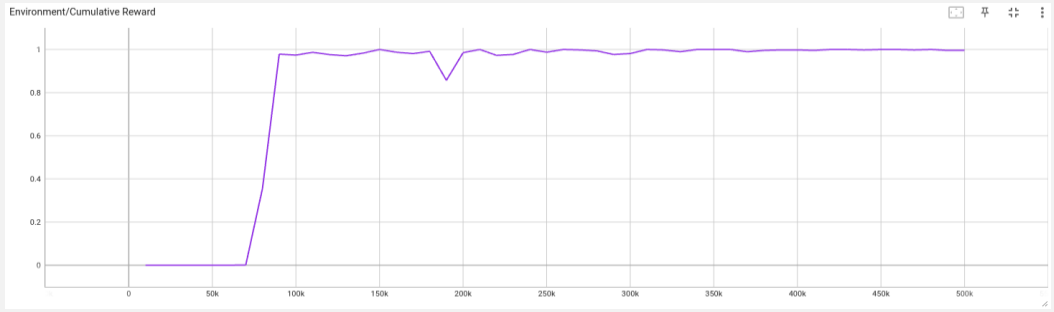
- Simple_maze의 기본 학습 설정은 Roller Ball 예제와 같음.
- LiDAR의 사용 여부(관측 정보), 거리와 시간에 따른 보상, 장애물(음의 보상 추가) 등을 바꿔서 학습함.

LiDAR의 사용 여부









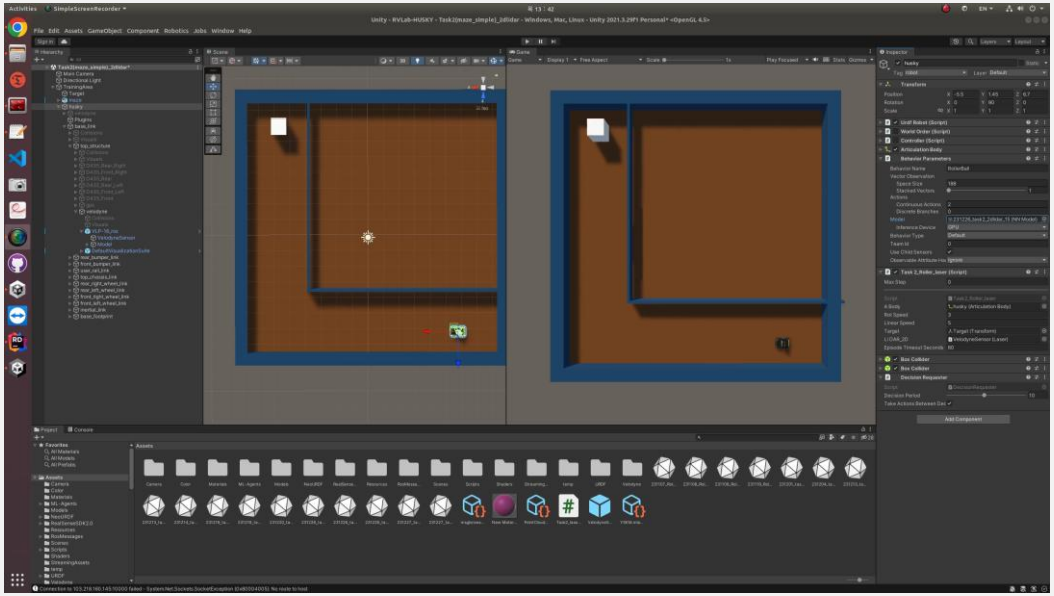
<2D LiDAR 사용 X>

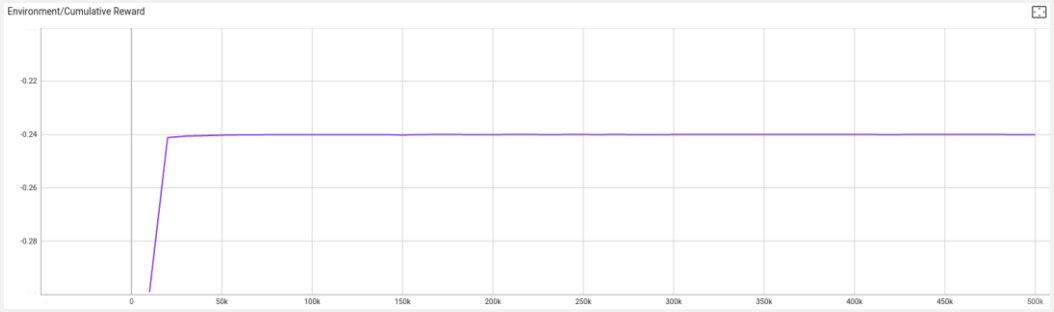
<2D LiDAR 사용 O>

- **차이점: 2D LiDAR 센서를 사용했는가?**
- 사용 X: 벽과 파티션 충돌이 계속 일어나며 500,000 step 이내에 목적지 도착 미완료. (보상: 0, 시간: 2.832hr)
- 사용 O: 코너 부분에서 인코스를 타며 500,000 step 이내에 목적지 도착 완료. (보상: 1에 수렴, 시간: 6.155hr)

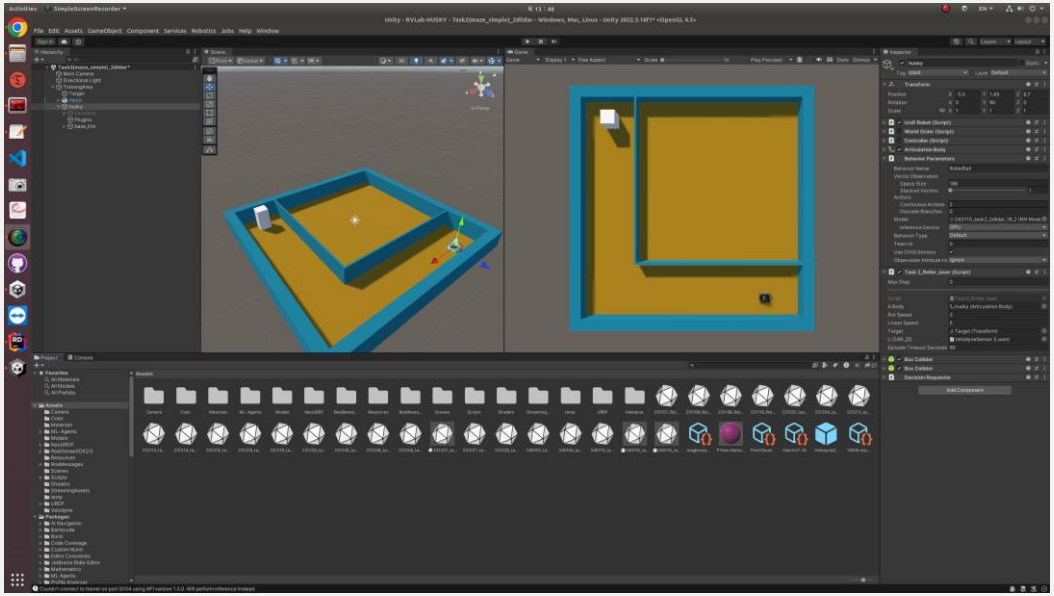
RV.Lab
Robot & Virtual Reality

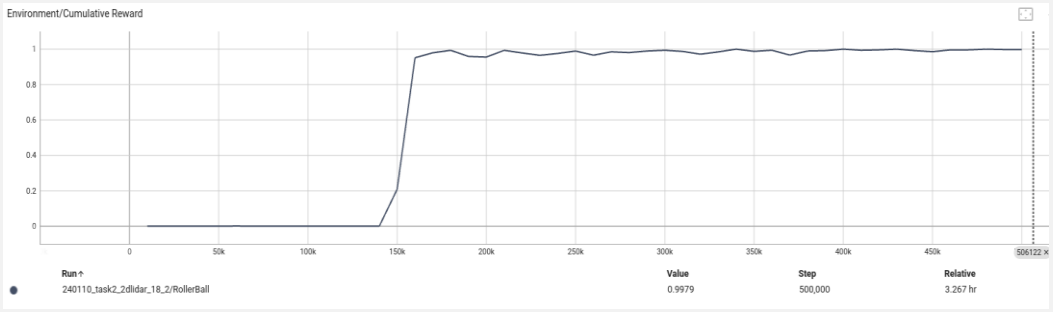
시간에 따른 보상





<A: 매 step 마다 –0.01 보상>





<B: 60초 시간 제한>

- **차이점: 시간에 따른 보상**
- A: 로봇이 에피소드를 빨리 끝내려고 장애물에 고의로 충돌함. 학습 미완료. (보상: -0.24에 수렴, 시간: 3.207hr)
- B: 60초 시간 제한을 두니, 학습 시간이 단축됨. (보상: 1에 수렴, 시간: 3.267hr)

거리에 따른 보상

거리 보상 수정

```
if (distanceToTarget > 14.5f && distanceToTarget <= 17.0f)
{
    // Debug.Log("stage 1");
    AddReward( increment: 0.001f);
}
else if (distanceToTarget > 13.3f && distanceToTarget <= 14.5f)
{
    // Debug.Log("stage 2");
    AddReward( increment: 0.002f);
}
else if (distanceToTarget > 10.8f && distanceToTarget <= 13.3f)
{
    // Debug.Log("stage 3");
    AddReward( increment: 0.003f);
}
else if (distanceToTarget > 5.5f && distanceToTarget <= 10.8f)
{
    // Debug.Log("stage 4");
    AddReward( increment: 0.004f);
}
else if (distanceToTarget > 1.42f && distanceToTarget <= 5.5f)
{
    // Debug.Log("stage 5");
    AddReward( increment: 0.005f);
}
else if (distanceToTarget <= 1.42f)
{
    // Debug.Log("Finish");
    SetReward(10.0f);
    EndEpisode();
}

if (Time.time - episodeStartTime >= episodeTimeoutSeconds)
{
    EndEpisode();
}
```

<1. 구역 분할>

```
if (distance <= 1.42f)
{
    SetReward(1.0f);
    EndEpisode();
}
else
{
    float reward = preDist - distance;
    // Debug.Log($"reward: {preDist}, {distance}, {reward}");
    SetReward(reward);
    preDist = distance;
}
```

<2. 거리 변화량에 따른 보상>

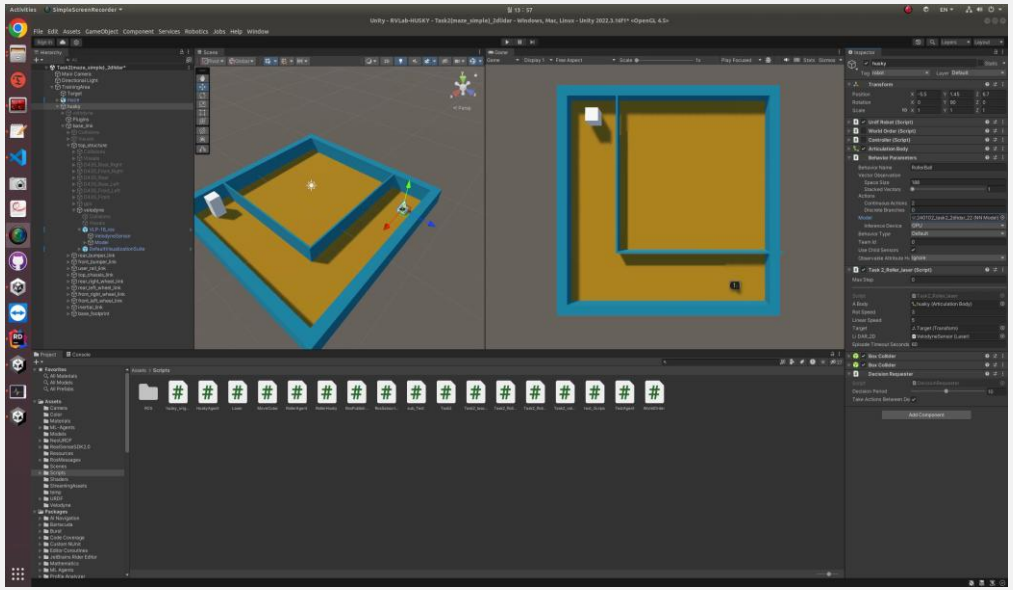
```
if (distance <= 1.42f)
{
    SetReward(100.0f);
    EndEpisode();
}
else
{
    float reward = preDist - distance;
    // Debug.Log($"reward: {preDist}, {distance}, {reward}");
    SetReward(reward);
    preDist = distance;
}
```

<3. 거리 변화량에 따른 보상(수정)>

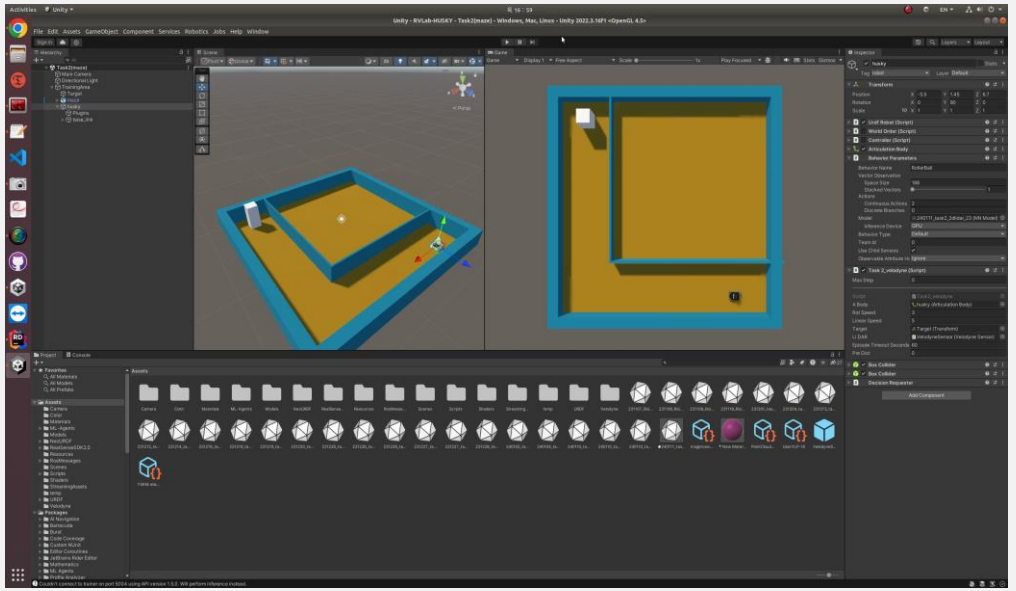
- **차이점: 거리에 따른 보상(구역 분할 or 거리 변화량에 따른 보상)**
- 구역 분할: 목적지와 로봇의 거리가 가까워질수록 큰 보상을 주되, 목적지 도착의 보상보다 작아야 학습됨.
- 거리 변화량: 목적지 도착의 보상이 거리 변화량에 따른 보상보다 커야 한다. 다만, 너무 큰 숫자는 수렴하려면 학습 step 수가 길어지고 이는 학습 시간 증가를 초래함.

거리에 따른 보상

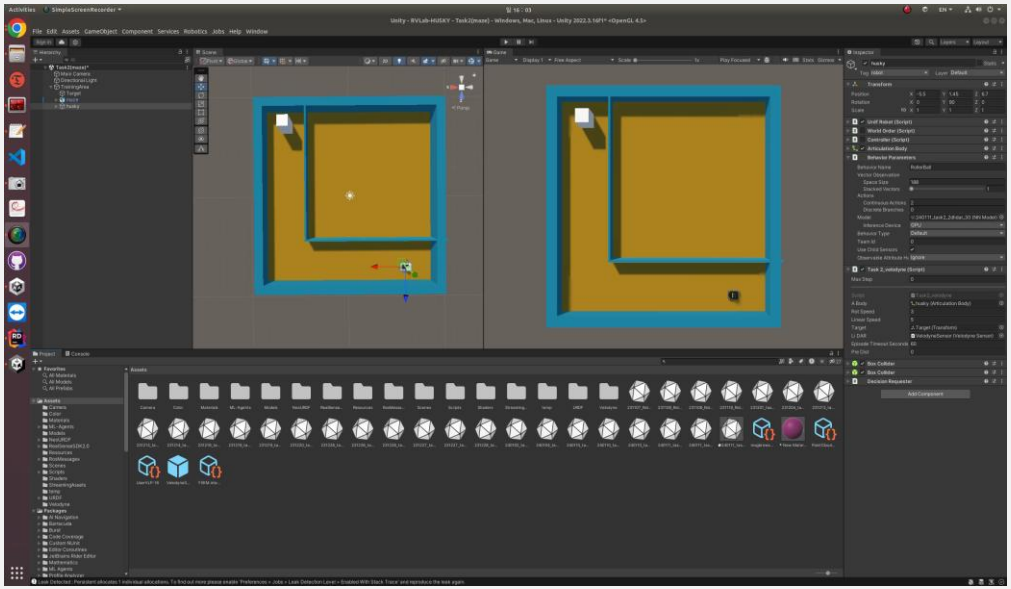
거리에 따른 보상



<1. 구역 분할>



<2. 거리 변화량에 따른 보상>

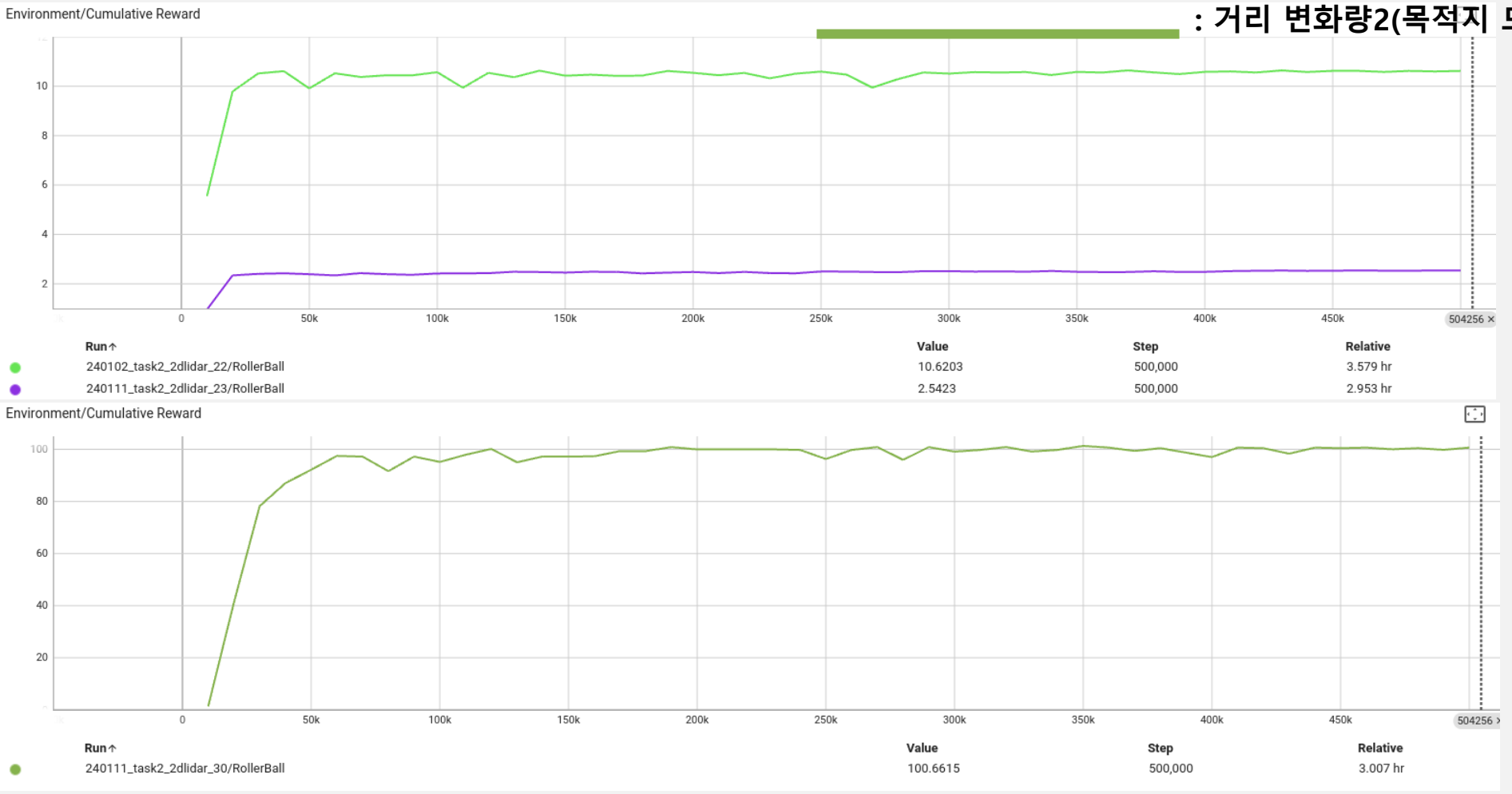


<3. 거리 변화량에 따른 보상(수정)>

거리에 따른 보상

RV.Lab
Robot & Virtual Reality

: 구역 분할
: 거리 변화량1(목적지 도착: +1.0)
: 거리 변화량2(목적지 도착: +100.0)



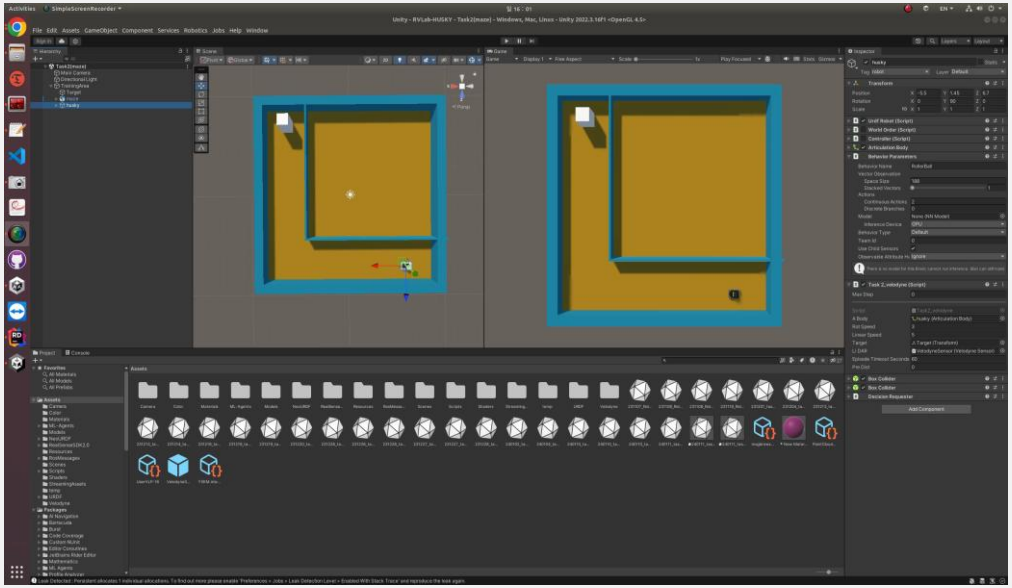
<reward>

- 학습시간: 3.579hr & 2.953hr & 3.007hr
- 보상: 10.6, 2.54, 100.66에 수렴함.
- Step: 500,000

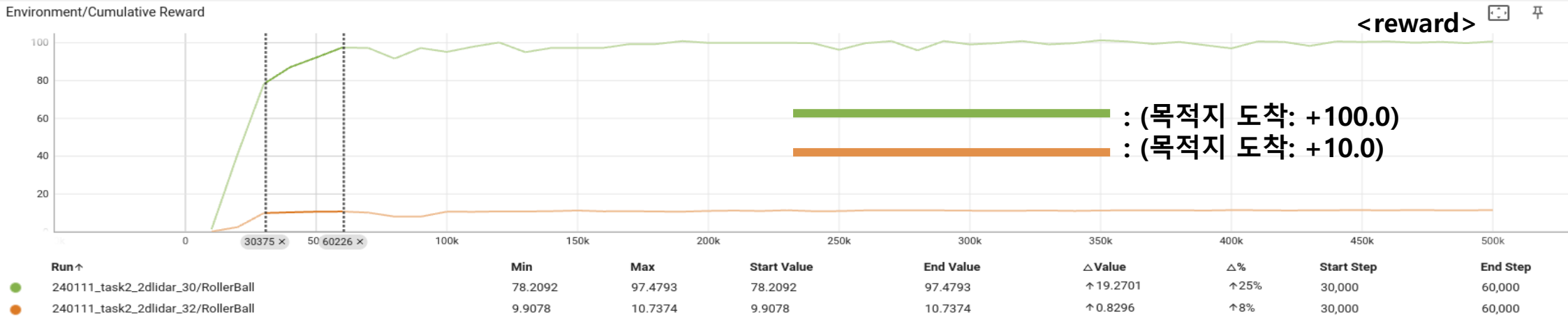
→ 목적지 도착(+1.0) 보상만 있을 때보다 약 3시간 정도 학습 시간 단축.

거리에 따른 보상

거리에 따른 보상



<거리 변화량에 따른 보상(목적지 도착: +10.0)>

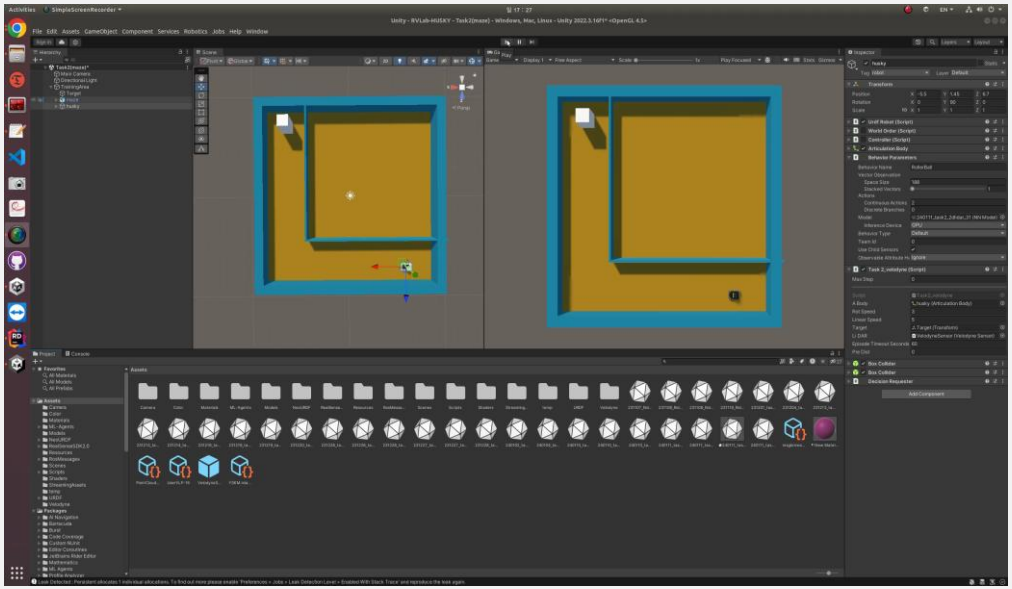


Run	Min	Max	Start Value	End Value	△Value	△%	Start Step	End Step
240111_task2_2dlidar_30/RollerBall	78.2092	97.4793	78.2092	97.4793	↑19.2701	↑25%	30,000	60,000
240111_task2_2dlidar_32/RollerBall	9.9078	10.7374	9.9078	10.7374	↑0.8296	↑8%	30,000	60,000

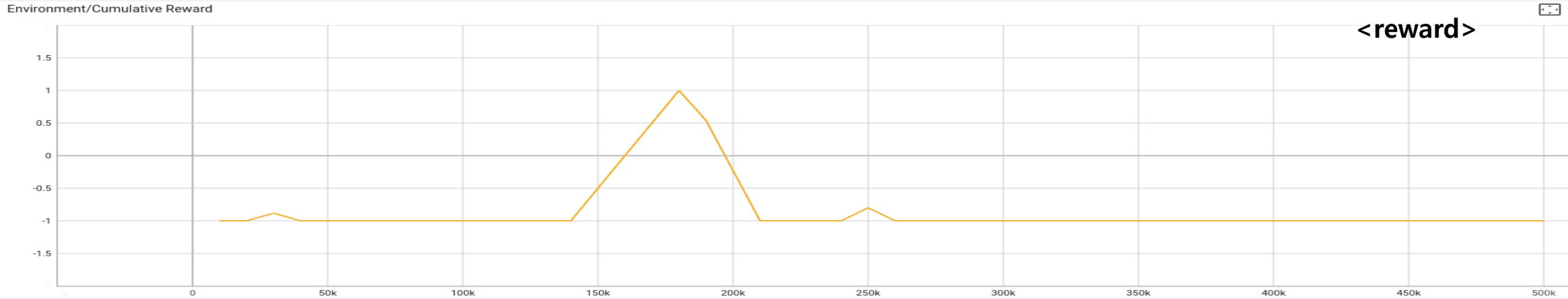
- 목적지 도착을 +10.0으로 한 것이 +100.0으로 한 것보다 30,000 step 정도 더 빨리 수렴함.
- 따라서 보상 값은 너무 큰 것보다 적절한 크기로 설정하는 것이 좋음.

장애물(음의 보상 추가)

장애물(음의 보상 추가)



<장애물(음의 보상 추가)>



<reward>

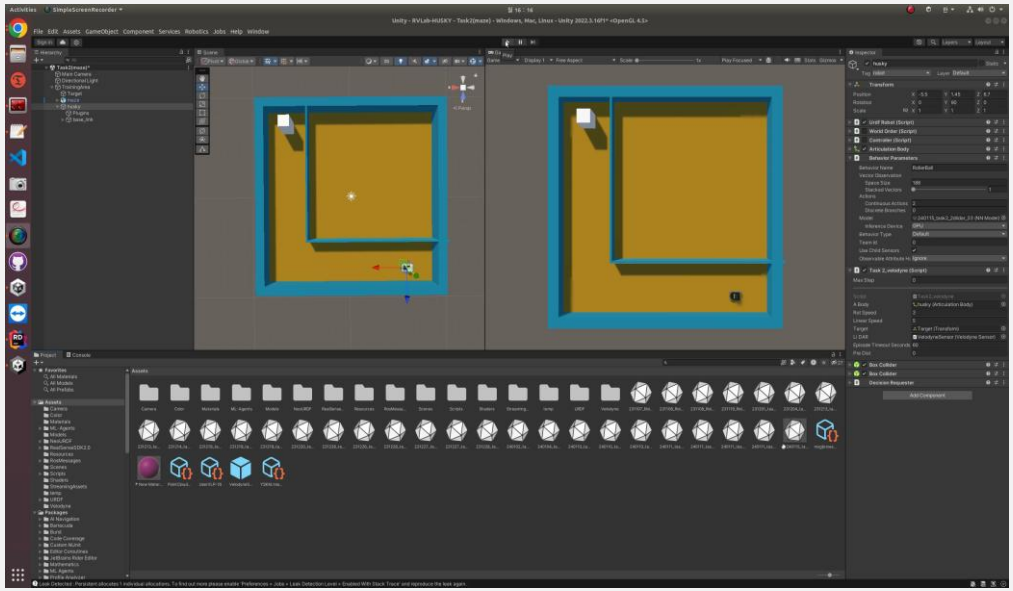
- 벽, 파티션 충돌의 경우, 에피소드 종료 뿐만 아니라 음의 보상 추가(-1.0)

→ 학습 도중 모습:장애물과 충돌하지 않으려고 에이전트의 행동이 조심스러움. 에피소드가 끝나지 않는 현상 발생함. (학습시간: 2.950hr, step: 500,000, 보상: -1에 수렴)

→ 따라서 60초 시간 제한(강제 에피소드 종료)이 필수.

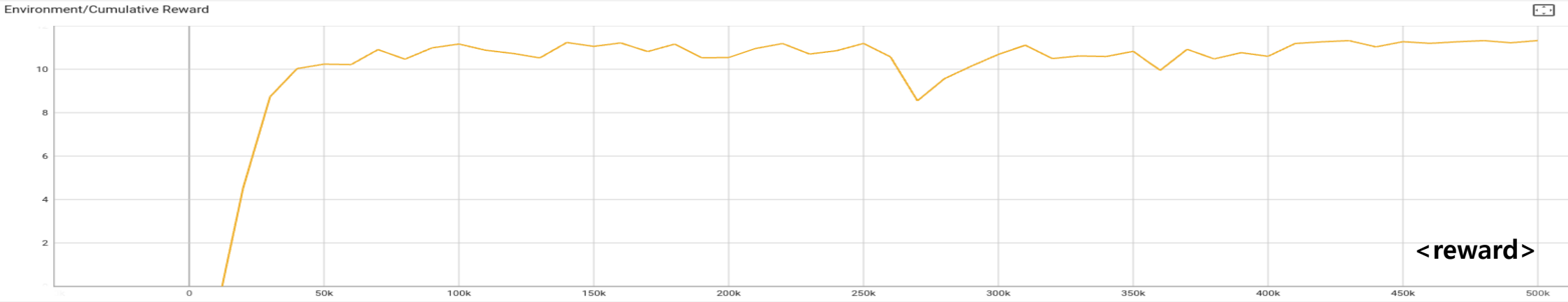
복합적인 보상

복합적인 보상



<복합적인 보상>

Reward	
목적지 도착	+10.0(거리 1.42m 이내면 도착으로 간주) & 종료
장애물 충돌	-1.0 & 종료(Wall, Partition)
거리 변화량	(이전과 목적지 거리 – 현재와 목적지 거리)
매 step	-0.01
그 외, 60초 시간 제한(에피소드의 시간)	



<reward>

- 학습시간: 2.974hr, step: 500,000, 보상: 11에 수렴

→ 양의 보상의 합이 음의 보상의 합보다 커야 매 step 마다 더하는 -0.01이 고의 충돌을 안함.

→ 거리, 시간에 따른 보상까지 다 추가해도 ml-agent 상에서의 간단한 미로 환경 학습 시간 단축은 약 3hr 한계임.