Unity를 활용한 드론 배송시스템 구현

2조 : 김현우
김범진
김창기
안종원

1

프로젝트 개요

동기

전 세계적으로 배달, 배송 등 물류 산업에서 드론 활용의 중요성이 점차 증가함에 따라 드론 택배 시장규모 증가 및 해외 관련 기술 개발 증가

프로젝트 구현목표

두개의 물체를 일직선상에서 랜덤하게 생성되는 타겟에 순서대로 배송하는 것을 목표로 한다.

강화학습 기법

PPO(Proximal Policy Optimization)

PPO 알고리즘의 정책 업데이트 방식은 이전의 모델과 현재 모델을 비교해서 더 나은 방향으로 학습이 진행되는 방식이기 때문에 학습 성능의 점진적인 향상을 보장하고, 이로 인해 보상이 점진적으로 증가하므로 타 알고리즘보다 교착상태가 덜 발생한다.

PPO의 batch size를 크게하고, time horizon을 높이면 좀 더 안정적인 배송이 가능했다.

```yaml
behaviors:
  Drone:
    trainer_type: ppo
    hyperparameters:
      batch_size: 64
      buffer_size: 12000
      learning_rate: 0.0003
      beta: 0.001
      epsilon: 0.2
      lambd: 0.99
      num_epoch: 3
      learning_rate_schedule: linear
    network_settings:
      normalize: true
      hidden_units: 128
      num_layers: 2
      vis_encode_type: simple
    reward_signals:
      extrinsic:
        gamma: 0.99
        strength: 1.0
    keep_checkpoints: 5
    max_steps: 10000000
    time_horizon: 1000
    summary_freq: 12000
```

2

프로젝트 구현과정

프로젝트 구현과정

POÈME
City Voxel Pack

PROFESSIONALASSETS
무료 팩 전문 무인 비행기 팩 +...

CROW ART
PBR Cardboard Box

실습에서 사용한 ‘City Voxel pack’과 ‘Drone free pack’을 그대로 사용하고 배송할 물체는 ‘PBR Cardboard Box’ 에셋사용. 배송위치는 직접 구현.

프로젝트 구현과정

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

using Unity.MLAgents;
using Unity.MLAgents.Sensors;
using Unity.MLAgents.Actuators;


0 references
public class PickupObject : MonoBehaviour
```

```
void Start()
{
    pickuptransform = gameObject.transform;
}
```

PickupObject 스크립트를 새로 구성해서 에이전트가 물체를 PickDown 할 수 있도록 구성

```
public void OnCollisionEnter(Collision coll)
{
    if(coll.collider.CompareTag("Pickdown"))
    {
        Debug.Log("Pickdown success");
        pickdownflag = true;
    }
```

Pickdown으로 태그된 오브젝트와 충돌시
pickdownflag를 true로 바꿔 배송을 완료한 상태를 표시

프로젝트 구현과정

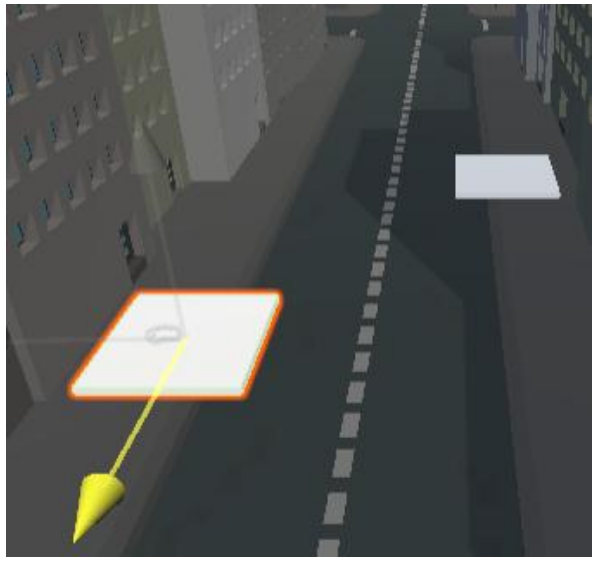
◆ **State :**

1. 드론 에이전트의 위치 좌표

```
private Transform agentTrans;
agentTrans = gameObject.transform;
Vector3.Magnitude(agentTrans.position);
```

2. 랜덤으로 생성되는 배송 목적지의 위치 좌표 (x,y,z)

```
HomeTrans.position = HomeInitPos + new Vector3(Random.Range(-0f, 0f), Random.Range(-0f, 0f), Random.Range(-40f, 0f));
Home_2Trans.position = Home_2InitPos + new Vector3(Random.Range(-0f, 0f), Random.Range(-0f, 0f), Random.Range(-40f, 0f));
```



Z축 방향으로만 -40에서 0사이에서 랜덤으로 보도블럭 위에 배송 목적지가 배치되도록 한다.

프로젝트 구현과정

◆State :

3. 각 오브젝트와 드론 사이의 거리

```
//픽업(큐브1)과 드론사이 거리
float distance_pickup_agent = Vector3.Magnitude(PickupTrans.position - agentTrans.position);
```

```
//픽다운(홈1)과 드론사이 거리
float distance_pickdown_agent = Vector3.Magnitude(PickdownTrans.position - agentTrans.position);
```

```
//픽업2(큐브2)과 드론사이 거리
float distance_pickup_2_agent = Vector3.Magnitude(Pickup_2Trans.position - agentTrans.position);
```

```
//픽다운2(홈2)과 드론사이 거리
float distance_pickdown_2_agent = Vector3.Magnitude(Pickdown_2Trans.position - agentTrans.position);
```

프로젝트 구현과정

◆State :

4. 다섯개의 플래그 변수

```
8 references
public bool flag;
8 references
public bool flag_2;
4 references
public bool pickdownflag;
4 references
public bool pickdownflag_2;
8 references
public bool pickdown_done_flag;
```

초기 x,x/x,x/x

물체1 픽업 o,x/x,x/x

물체1 픽다운 x,o/x,x/o

물체2 픽업 x,o/o,x/o

물체2 픽다운 x,o/x,o/o

초기에는 모든 플래그가 false로 시작하고 두개의 물체의 픽업 여부과 픽다운 여부에 따라 true와 false로 바뀜

프로젝트 구현과정

◆Action:

```
var actions = actionBuffers.ContinuousActions;

float moveX = Mathf.Clamp(actions[0], -1, 1f);
float moveY = Mathf.Clamp(actions[1], -1, 1f);
float moveZ = Mathf.Clamp(actions[2], -1, 1f);
```

Action은 x, y, z축 방향으로의 연속적인 이동
Mathf.Clamp 함수를 통해 입력값을 지정한 최솟값, 최댓값의 범위로 잘라서 사용

프로젝트 구현과정

◆ Reward:

```
public override void OnActionReceived(ActionBuffers actionBuffers)
{
    AddReward(-0.01f);
```

매 스텝마다 -0.01의 reward : 에이전트가 타겟에 빨리 도달했을 때 보상의 총합이 크게한다.

```
var altitude = agentTrans.position.y;
if(altitude >= 25){
    Debug.Log("Please Throttle down");
    SetReward(-3f);
    EndEpisode();
```

일정고도 이상이 되면 -3의 reward, 에피소드 종료

프로젝트 구현과정

```
if(flag != true)
```

에이전트가 물체를 PickUp한 상태가 아닐때

```
if(distance_pickup_agent > 30f)
{
    SetReward(-1f);
    Debug.Log("e1");
    EndEpisode();
}
```

```
if(distance_pickup_2_agent > 100f)
{
    SetReward(-1f);
    Debug.Log("e3");
    EndEpisode();
}
```

- 배송할 물체와의 거리가 일정 거리 이상이 되면 error 알림 출력 후 -1의 reward, 에피소드 종료

```
float reward1 = predistance_pickup_agent - distance_pickup_agent;
AddReward(reward1 * 5f);
predistance_pickup_agent = distance_pickup_agent;
```

물체와의 거리 변화량에 5를 곱한만큼의 reward: 에이전트가 물체와 가까워지도록 한다.

프로젝트 구현과정

```
if(distance_pickdown_agent > 100f)
{
    SetReward(-1f);
    Debug.Log("e2");
    EndEpisode();
}
```

```
if(distance_pickdown_2_agent > 100f)
{
    SetReward(-1f);
    Debug.Log("e4");
    EndEpisode();
}
```

Pickup 후 목적지와의 거리가 너무멀어지면 -1의 reward, 에피소드종료

```
float reward2 = predistance_pickdown_agent - distance_pickdown_agent;
AddReward(reward2 * 5f);
predistance_pickdown_agent = distance_pickdown_agent;
```

목적지와의 거리 변화량에 5를 곱한만큼의 reward: 에이전트가 목적지와 가까워지도록 한다.

프로젝트 구현과정

```
if(coll.collider.CompareTag("Building"))
{
    Debug.Log("Collision with Building");
    SetReward(-10f);
    EndEpisode();
}
```

```
if(coll.collider.CompareTag("Road"))
{
    Debug.Log("Collision with Road");
    SetReward(-30f);
    EndEpisode();
}
```

```
if(coll.collider.CompareTag("Pavement"))
{
    Debug.Log("Collision with Pavement");
    SetReward(-1.0f);
    EndEpisode();
}
```

Agent가 장애물(빌딩, 도로, 보도)와 충돌 시 – reward 값 부여 후 에피소드 종료

```
SetReward(100f);
EndEpisode();
```

목표지점에 object들이 모두 배송 완료 되었을 시 reward 값을 100 부여 후 에피소드 종료

3

프로젝트 구현결과

프로젝트 구현결과

PPO 알고리즘으로 강화학습 수행 결과 시간이 지날수록 드론 에이전트가 물체를 Pickup 하여 목적지에 성공적으로 도달함을 확인할 수 있었다.

프로젝트 구현결과

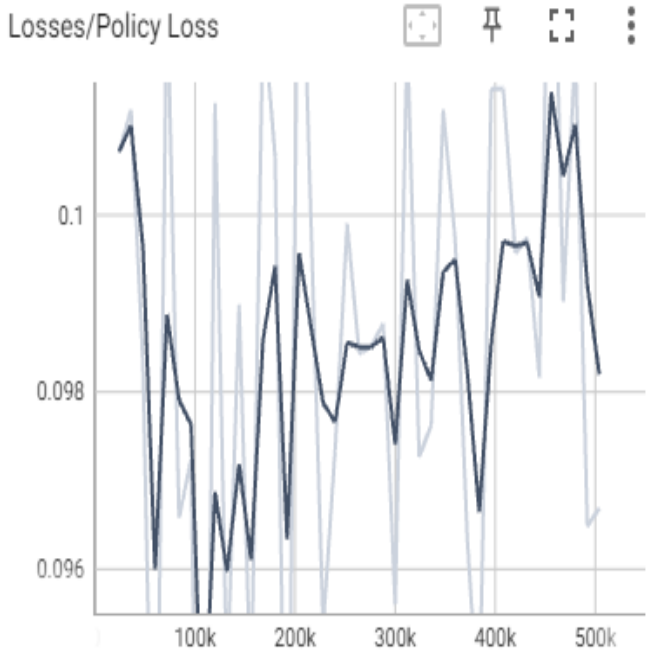
PPO 알고리즘으로 강화학습 수행 결과 시간이 지날수록 드론 에이전트가 물체를 Pickup 하여 목적지에 성공적으로 도달함을 확인할 수 있었다.

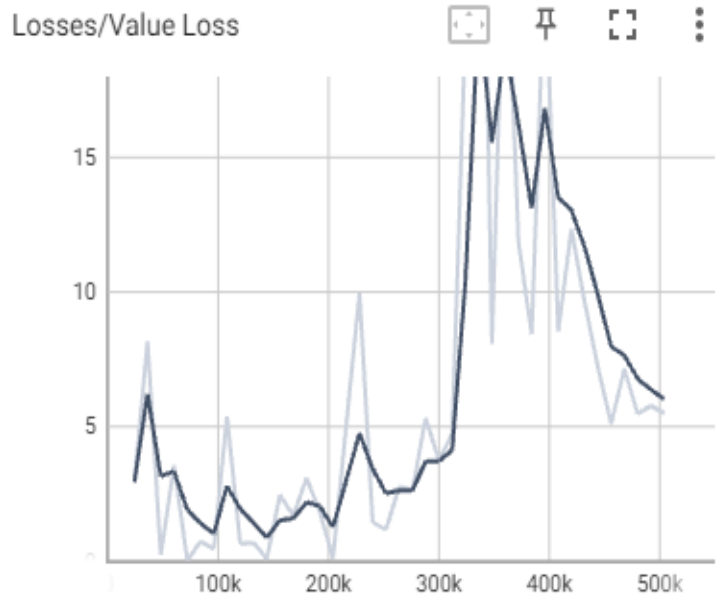
프로젝트 구현결과

```
[INFO] Drone. Step: 156000. Time Elapsed: 1122.282 s. Mean Reward: -22.639. Std of Reward: 43.332. Training.
[INFO] Drone. Step: 168000. Time Elapsed: 1215.137 s. Mean Reward: 117.701. Std of Reward: 89.983. Training.
[INFO] Drone. Step: 180000. Time Elapsed: 1305.106 s. Mean Reward: 206.278. Std of Reward: 48.263. Training.
[INFO] Drone. Step: 192000. Time Elapsed: 1389.244 s. No episode was completed since last summary. Training.
[INFO] Drone. Step: 204000. Time Elapsed: 1482.400 s. No episode was completed since last summary. Training.
[INFO] Drone. Step: 216000. Time Elapsed: 1555.904 s. Mean Reward: -35.994. Std of Reward: 94.741. Training.
[INFO] Drone. Step: 228000. Time Elapsed: 1641.173 s. Mean Reward: 154.032. Std of Reward: 13.487. Training.
[INFO] Drone. Step: 240000. Time Elapsed: 1732.815 s. Mean Reward: 229.769. Std of Reward: 0.000. Training.
[INFO] Drone. Step: 252000. Time Elapsed: 1814.954 s. Mean Reward: 255.371. Std of Reward: 7.015. Training.
[INFO] Drone. Step: 264000. Time Elapsed: 1902.652 s. Mean Reward: 161.107. Std of Reward: 30.070. Training.
[INFO] Drone. Step: 276000. Time Elapsed: 1984.176 s. Mean Reward: 218.040. Std of Reward: 72.680. Training.
[INFO] Drone. Step: 288000. Time Elapsed: 2066.224 s. Mean Reward: 279.270. Std of Reward: 26.950. Training.
[INFO] Drone. Step: 300000. Time Elapsed: 2149.068 s. Mean Reward: 199.629. Std of Reward: 54.628. Training.
[INFO] Drone. Step: 312000. Time Elapsed: 2236.067 s. Mean Reward: 115.810. Std of Reward: 104.474. Training.
[INFO] Drone. Step: 324000. Time Elapsed: 2318.072 s. Mean Reward: 79.411. Std of Reward: 123.692. Training.
[INFO] Drone. Step: 336000. Time Elapsed: 2403.500 s. Mean Reward: 79.399. Std of Reward: 117.714. Training.
[INFO] Drone. Step: 348000. Time Elapsed: 2491.644 s. Mean Reward: 139.022. Std of Reward: 38.903. Training.
[INFO] Drone. Step: 360000. Time Elapsed: 2581.972 s. Mean Reward: 109.211. Std of Reward: 65.952. Training.
[INFO] Drone. Step: 372000. Time Elapsed: 2667.045 s. Mean Reward: 113.298. Std of Reward: 23.764. Training.
[INFO] Drone. Step: 384000. Time Elapsed: 2756.448 s. Mean Reward: 129.730. Std of Reward: 18.158. Training.
[INFO] Drone. Step: 396000. Time Elapsed: 2848.536 s. Mean Reward: 94.336. Std of Reward: 68.949. Training.
[INFO] Drone. Step: 408000. Time Elapsed: 2935.729 s. Mean Reward: 124.681. Std of Reward: 18.992. Training.
[INFO] Drone. Step: 420000. Time Elapsed: 3031.909 s. Mean Reward: 110.669. Std of Reward: 33.697. Training.
[INFO] Drone. Step: 432000. Time Elapsed: 3121.436 s. Mean Reward: 111.996. Std of Reward: 18.869. Training.
[INFO] Drone. Step: 444000. Time Elapsed: 3216.030 s. Mean Reward: 119.080. Std of Reward: 19.233. Training.
[INFO] Drone. Step: 456000. Time Elapsed: 3301.518 s. Mean Reward: 94.214. Std of Reward: 38.061. Training.
[INFO] Drone. Step: 468000. Time Elapsed: 3391.884 s. Mean Reward: 112.164. Std of Reward: 6.456. Training.
[INFO] Drone. Step: 480000. Time Elapsed: 3484.737 s. Mean Reward: 115.408. Std of Reward: 28.791. Training.
[INFO] Drone. Step: 492000. Time Elapsed: 3568.306 s. Mean Reward: 128.789. Std of Reward: 24.561. Training.
```

Step이 지남에 따라 Mean Reward 또한 증가함을 확인.

프로젝트 구현결과





Tensorboard로 확인 결과 loss 값의 변화가 다소 불안정하지만 전반적으로 reward 값은 대체적으로 일정한 값만큼 증가하는 경향을 보이는 것을 확인할 수 있음

감사합니다