

2D Game Basic

마우스 처리

박성준

(sjpark.jesus@gmail.com)

학습 목표

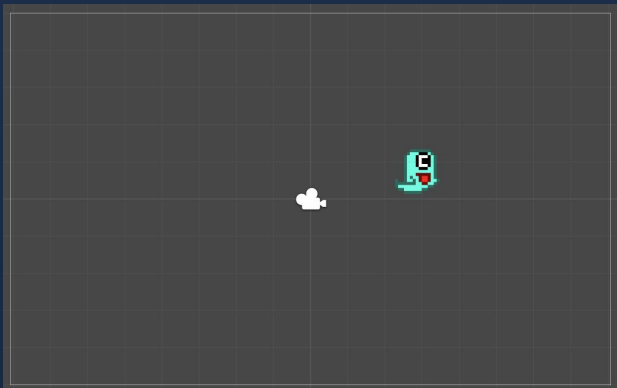
- 총돌 판정으로 마우스의 터치를 조사한다.
- 터치하면 사라진다
- 터치하면 계속 회전한다.
- 터치하면 룰렛처럼 회전하고 멈춘다.
- 오브젝트의 중첩 순서
- 장치 그림책

마우스 터치 및 중요 함수

- 게임 오브젝트가 마우스 터치가 되었는가를 조사함
 - **void OnMouseDown()** 을 사용
 - 게임 오브젝트에 Box Collider 2D 컴포넌트가 있어야 함
 - 응용
 - 마우스로 터치하면 사라짐
 - 마우스로 터치하면 회전
 - 마우스로 터치하면 룰렛 회전
- 게임오브젝트가 사라지게 하는 방법
 - **게임오브젝트.SetActive(false);**
- 자신의 게임오브젝트가 사라지게 하는 방법
 - **this.gameObject.SetActive(false);**

1.마우스 클릭 : 터치하면 사라진다.

- 충돌판정으로 마우스의 터치를 조사한다.
 - 이미지 : 유령 (obake_0)
 - 콤포넌트 : Box Collider 2D
 - 새로운 스크립트 작성 : OnMouseDown_Hide



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

public class OnMouseDown_Hide : MonoBehaviour
{
    void OnMouseDown() //터치하면
    {
        this.gameObject.SetActive(false); // 지운다
    }
}
```

2.마우스 클릭 : 터치하면 계속 회전한다.

- 충돌판정으로 마우스의 터치를 조사한다.
 - 이미지 : 해바라기 (flower_1)
 - 콤포넌트 : Box Collider 2D
 - 새로운 스크립트 작성 :
 - OnMouseDown_Rotate



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

public class OnMouseDown_Rotate : MonoBehaviour
{
    public float angle = 360;
    public float rotateAngle = 0;

    void OnMouseDown() //터치하면
    {
        rotateAngle = angle; //회전 각도를 지정
    }

    void OnMouseUp() // 터치를 그만하면
    {
        rotateAngle = 0;
    }

    void FixedUpdate() // 계속 호출한다 (일정 시간동)
    {
        this.transform.Rotate(0, 0, rotateAngle / 50);
    }
}
```

2.마우스 클릭 : 터치하면 계속 회전한다.

- 점점 줄어드는 방식

```
Speed = Speed * (float) 0.97;  
// speed = 1;  
// speed = speed * 0.98 -> 0.98 (0.98)  
// speed = speed * 0.98 -> 0.9604 (0.96)  
// speed = speed * 0.98 -> 0.941192 (0.94)  
// speed = speed * 0.98 -> 0.92236816 (0.92)  
this.transform.Rotate(xAngle: 0, yAngle: 0, zAngle: Speed);
```

마우스 클릭 : 터치하면 룰렛처럼 회전하고 멈춘다.

- 충돌판정으로 마우스의 터치를 조사한다.
 - 이미지 : 해바라기 (flower_1)
 - 컴포넌트 : Box Collider 2D
 - 새로운 스크립트 작성 :
 - OnMouseDown_Roulette



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

public class OnMouseDown_Roulette : MonoBehaviour
{
    public float maxSpeed = 0; //최대속도 지정
    float Speed = 0;

    void OnMouseDown()
    {
        Speed = maxSpeed; // 최대 속도를 낸다
    }

    void FixedUpdate()
    {
        Speed = Speed * (float)0.97; // 스피드를 점점 줄여서
        this.transform.Rotate(0, 0, Speed);
    }
}
```

생각 해 볼 문제

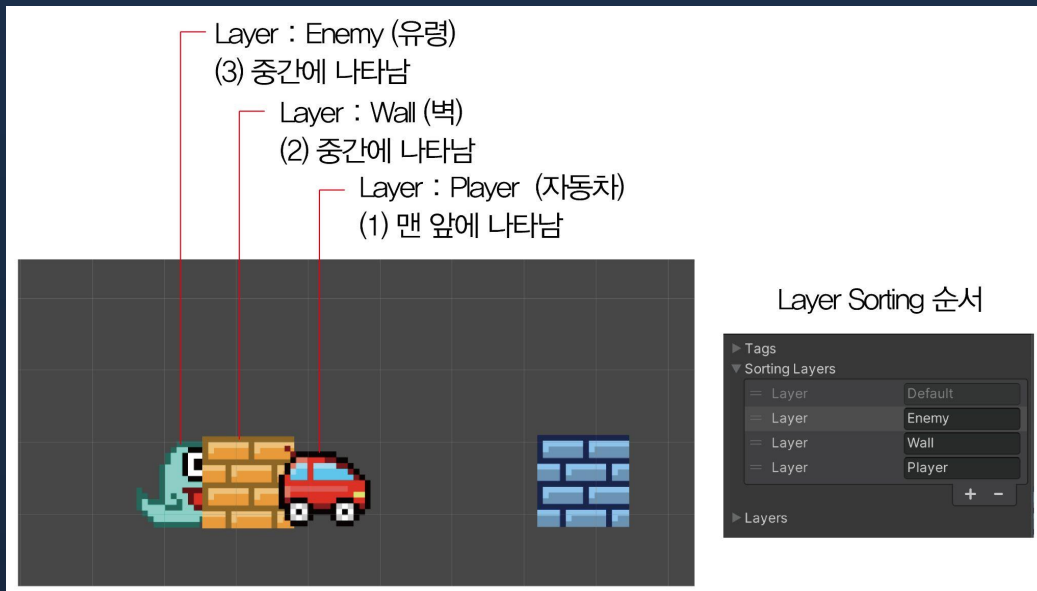
- 마우스를 터치 할 때 한번은 오른쪽으로 한번은 왼쪽으로 회전하는 해바라기를 구현해 보세요.
- 터치할 때 크기를 커지고 회전할때 원상태의 크기로 돌아오려면?

오브젝트의 중첩 순서

- 하나의 씬에 여러개의 게임 오브젝트를 배치하면 겹치는 경우가 있다. 게임 오브젝트가 겹치면 한쪽이 앞쪽에, 다른 한쪽이 안쪽에 나타나게 되는데 이 중첩 순서를 바꿀 수 있다.
- 중첩 순서를 바꾸는 방법
 - **Sorting Layer** : 가장 우선순위가 높은 방법
 - **Order in Layer**
 - **Position Z** : 가장 약한 지정 방법

오브젝트의 중첩 순서

- **Sorting Layer :**
 - 겹치는 레이어를 여러개 준비하고 어느 레이어에 속하는지에 따라 깊이를 지정합니다.
 - 레이어에 이름을 붙여서 깊이를 지정할 수 있으므로 의미를 쉽게 알수 있는 지정 방법
- 유령 오브젝트를 선택 - **Sprite Render** 아래 - **Additional Setting** - **Sorting Layer**에서 지정
 - **Order in Layer** 는 신경쓰지 말자 (**Layer**가 우선 순위가 높음)
- 새로운 레이어를 만든다.



오브젝트의 중첩 순서

- Order in Layer :
 - 중첩 레이어 내에서 번호로 깊이를 지정하는 방법
 - 값이 클수록 맨 앞에 위치하고 가장 적을 수록 맨 뒤에 위치한다.



▼ Additional Settings

Sorting Layer Default
Order in Layer 0

Layer : Default
유령 : 0

▼ Additional Settings

Sorting Layer Default
Order in Layer 1

Layer : Default
벽 : 1

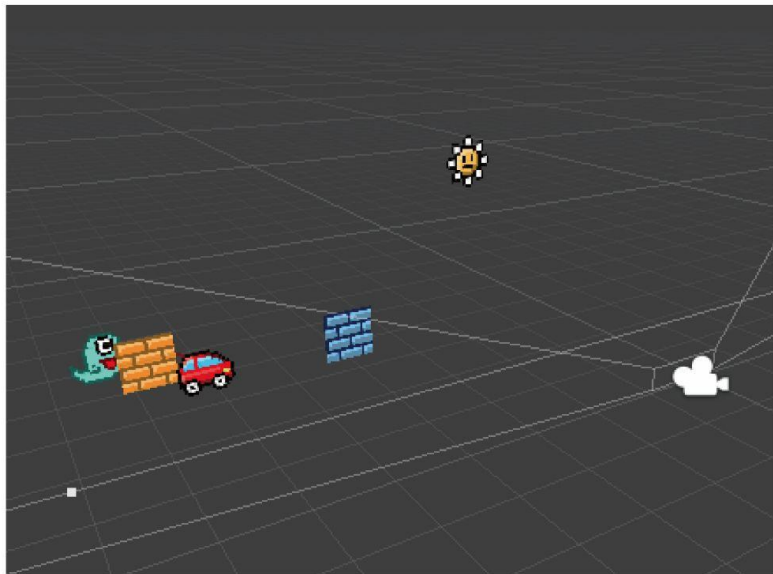
▼ Additional Settings

Sorting Layer Default
Order in Layer 2

Layer : Default
자동차 : 2

오브젝트의 중첩 순서

- PositionZ :
 - 같은 중첩 레이어 내에서 심지어 Order in Layer도 같은 값일때에 지정하는 방법
 - 카메라로부터 거리가 가까울 수록 맨 앞쪽에 표시 (왼손 좌표계)
 - Transform 의 z 값이 적을 수록 쪽에 배치된다.



Transform						
Position	X	-6.75	Y	-2.55	Z	2
Rotation	X	0	Y	0	Z	0
Scale	X	1	Y	1	Z	1

Position Z : 2
유령

Transform						
Position	X	-5.83	Y	-2.56	Z	1
Rotation	X	0	Y	0	Z	0
Scale	X	1	Y	1	Z	1

Position Z : 1
벽

Transform						
Position	X	-4.77	Y	-2.53	Z	0
Rotation	X	0	Y	0	Z	0
Scale	X	1	Y	1	Z	1

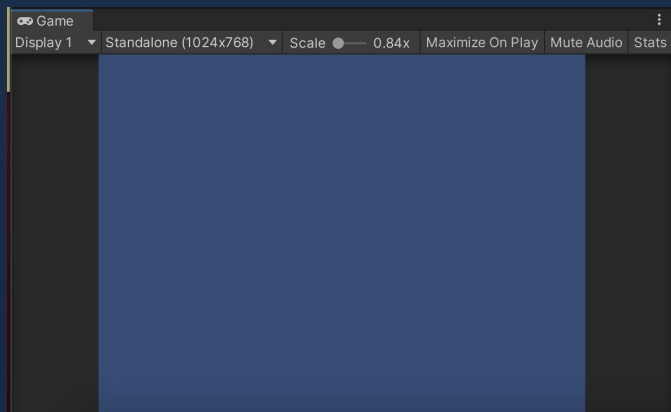
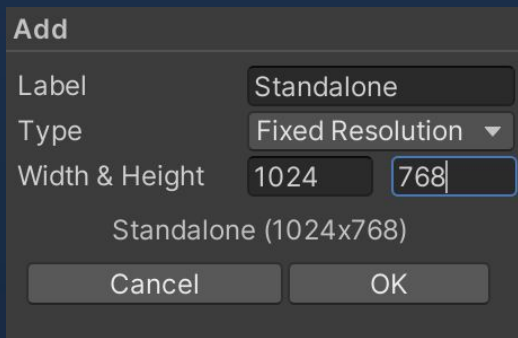
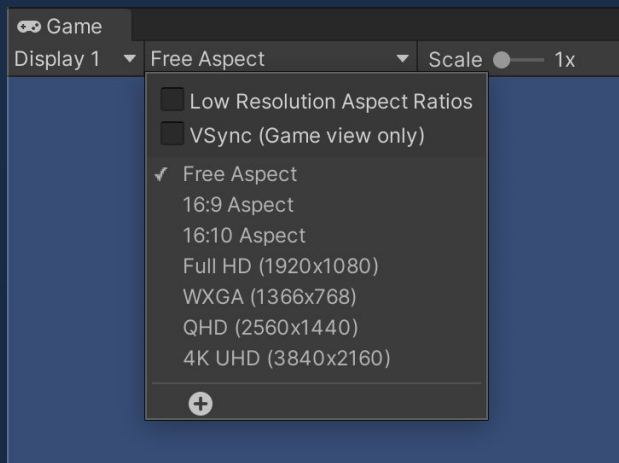
Position Z : 0
자동차

(샘플어플) 장치 그림책 만들기

- 게임의 규칙
 - 유령을 터치하면 사라집니다.
 - 태양을 터치하면 회전합니다
 - 물음표 블록을 터치하면 안에서 룰렛이 나옵니다.
 - 룰렛을 터치하면 잠시 회전하다가 멈춥니다.
- 필요한 이미지
 - 유령 : Obake_0
 - 태양 : flower_0
 - 룰렛 : flower_2
 - 물음표 블록 : block_08
- 필요한 스크립트
 - 터치하면 지운다 : OnMouseDown_Hide
 - 터치하면 회전한다 : OnMouseDown_Rotate
 - 터치하면 룰렛처럼 회전하다가 멈춘다. : OnMouseDown_Roulette

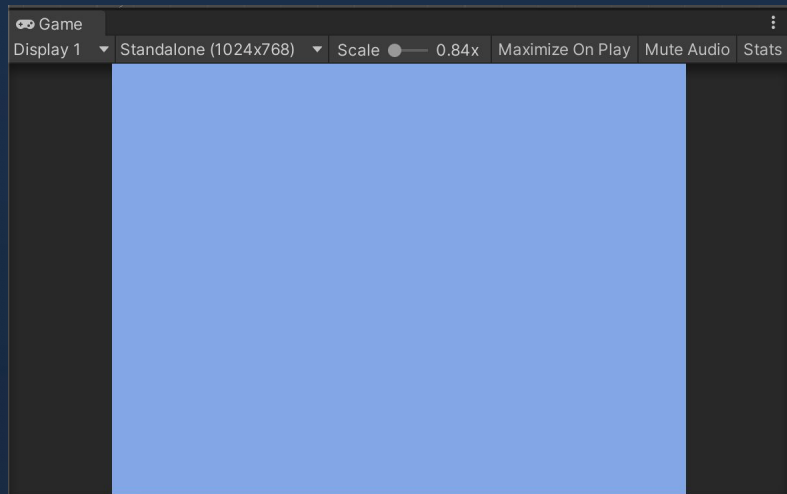
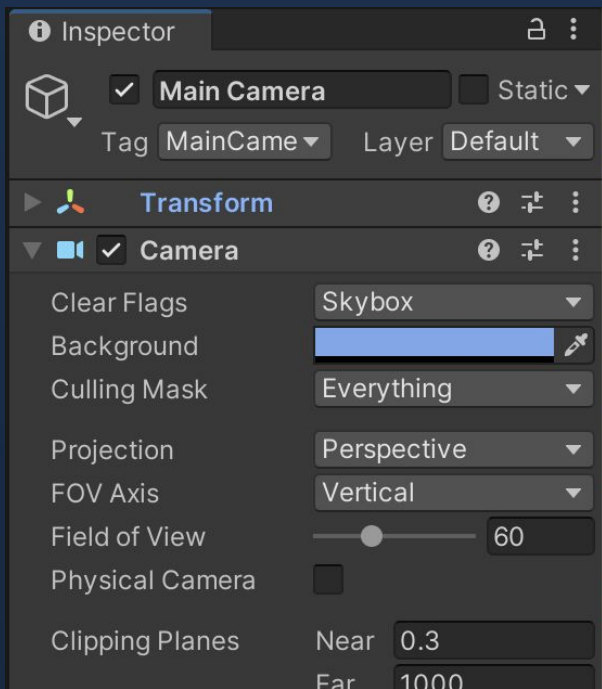
(샘플어플) 장치 그림책 만들기

- (1) 게임 탭에서 [Standalone(1024 X 768)] 을 선택한다.
 - Display Mode가 없으면 추가(+) 버튼을 클릭한 다음 새롭게 모드를 추가한다.



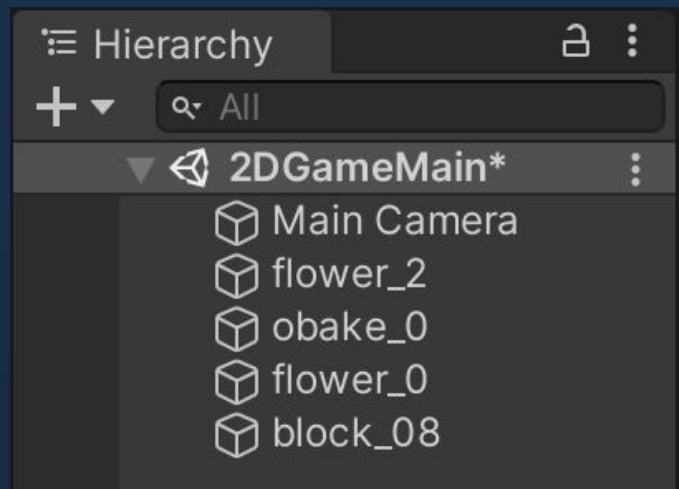
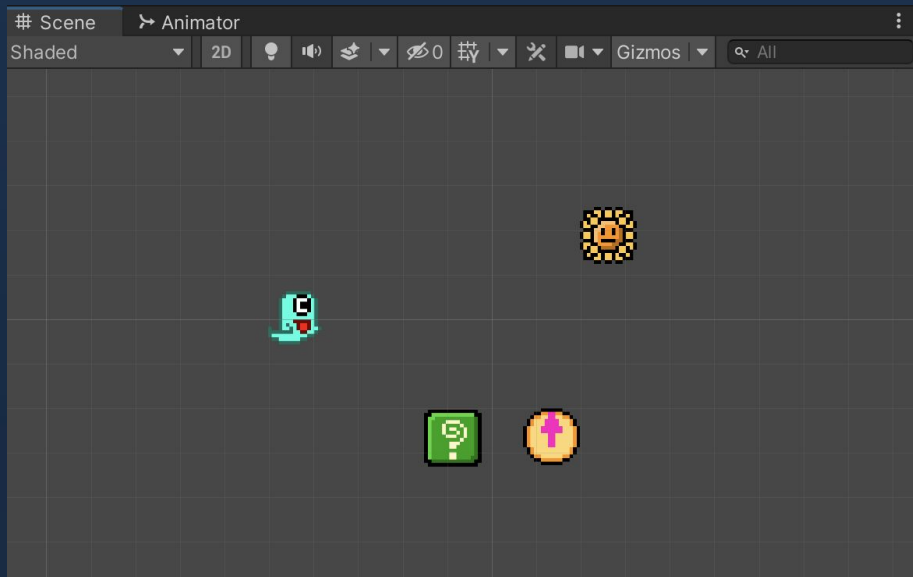
(샘플어플) 장치 그림책 만들기

- (2) Main Camera - Background를 선택한 다음 배경 색을 바꾸어 보도록 한다.



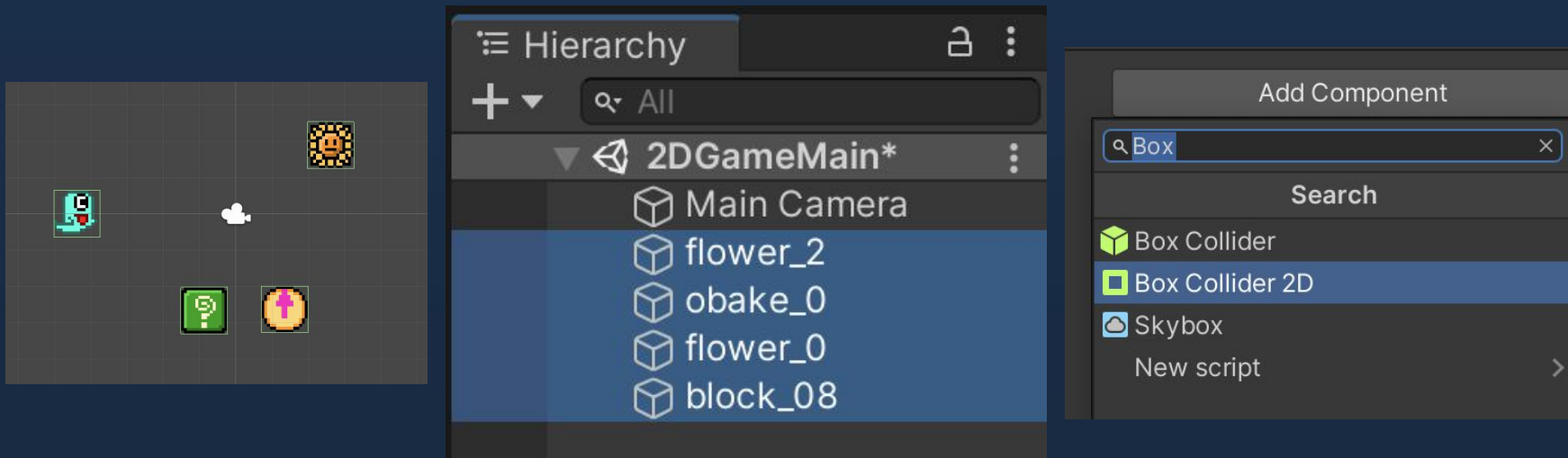
(샘플어플) 장치 그림책 만들기

- (3) 4개의 게임 오브젝트를 배치한다



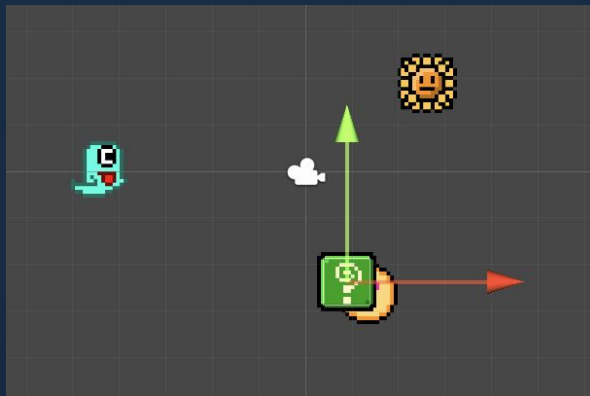
(샘플어플) 장치 그림책 만들기

- (4) 4개의 게임 오브젝트에 Box Collider 컴포넌트를 추가한다.
 - Shift 키를 누른 상태에서 4개의 게임 오브젝트를 선택 – Add Component – Box Collider2D 검색 추가



(샘플어플) 장치 그림책 만들기

- (5) 스크립트를 추가한다.
 - 유령에 : OnMouseDown_Hide
 - 태양 : OnMouseDown_Rotate
 - 룰렛 : OnMouseDown_Roulette
 - 물음표 블록 : OnMouseDown_Hide
- (6) 물음표 블록의 Transform - Position Z 를 -1로 지정하고 룰렛을 겹치게 하여 룰렛을 감추도록 합니다.



Transform				
Position	X	0.89	Y	-2.37
			Z	-1
Rotation	X	0	Y	0
			Z	0
Scale	X	1	Y	1
			Z	1

(샘플어플) 장치 그림책 만들기

- 결과 화면

