

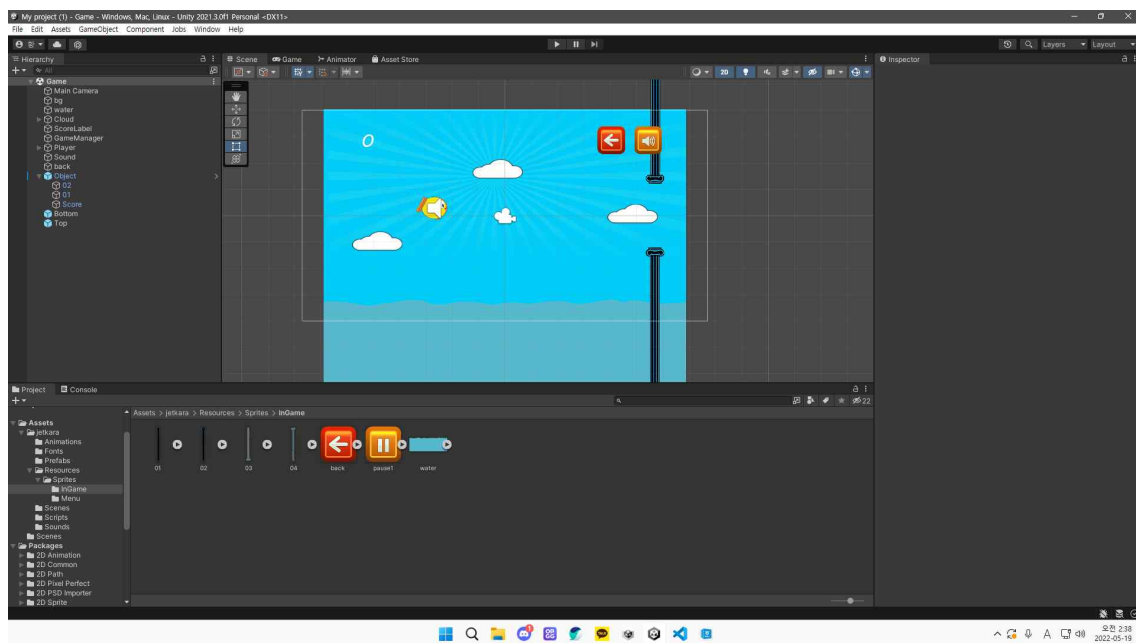
게임프로그래밍 과제4

201903868 황가은

Kara Project에서 변경한 부분은 다음과 같습니다. 추가적인 부분은 아래 구현 과정에 따로 설명해놓았습니다.

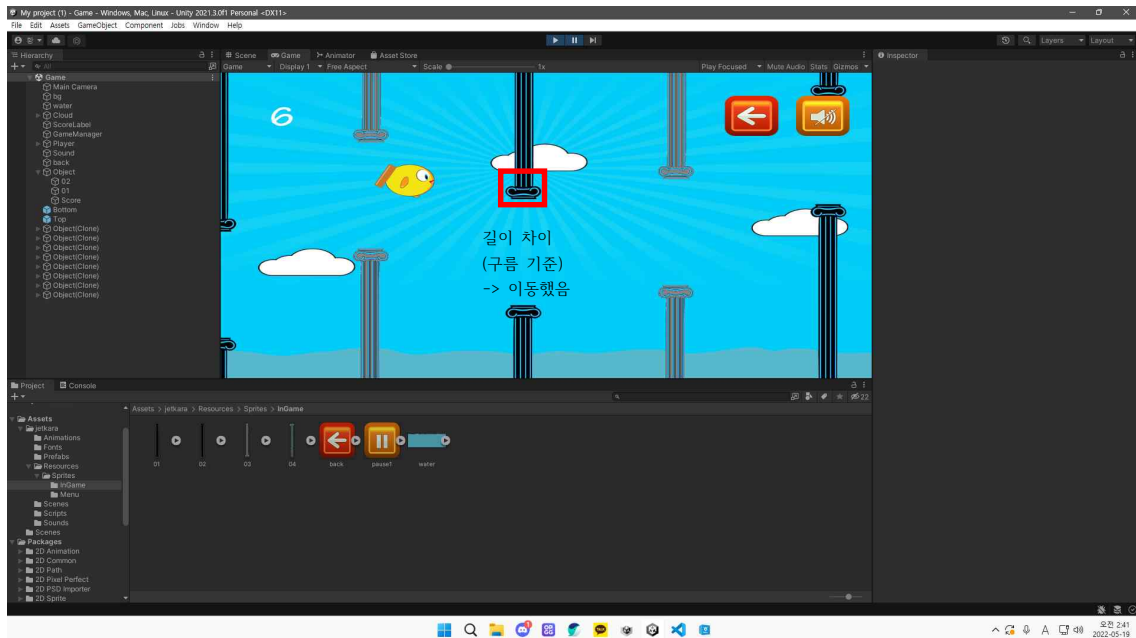
1. 윗기둥을 움직이는 애니메이션 추가
2. 윗기둥이 움직임에 따라 스코어 애니메이션도 그 규격에 맞도록 움직이도록 추가
3. 회색 기둥 Sprite3(윗기둥), Sprite4(아랫기둥)를 추가

<처음 화면>

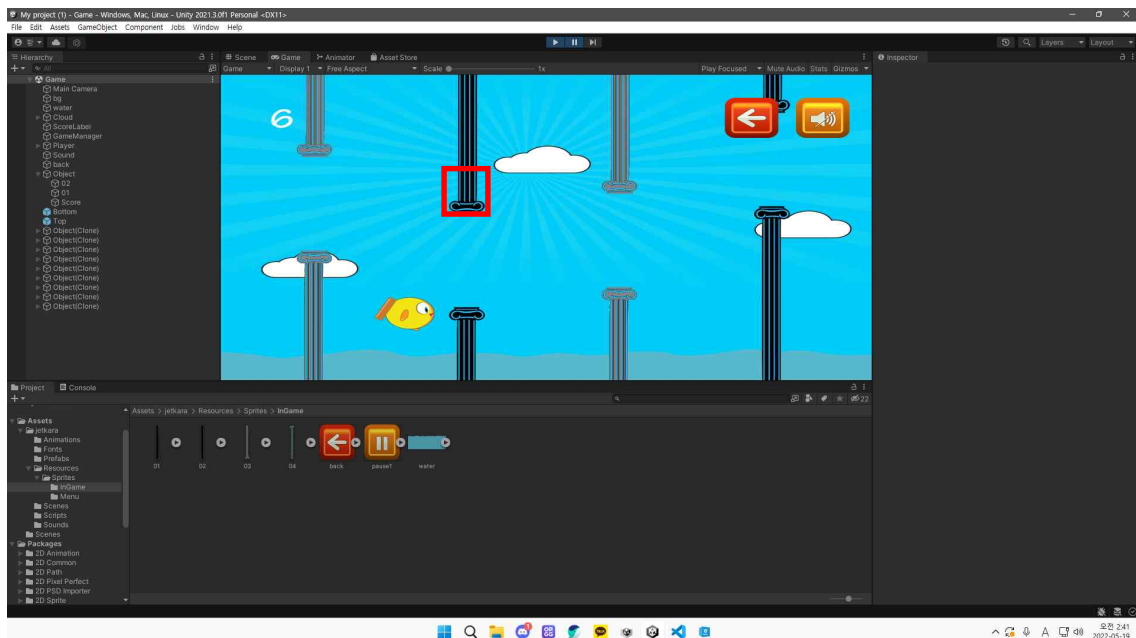


<실행 화면>

<그림 1>



<그림 2>



<그림 1>에서 <그림 2>를 보시면 기둥이 이동했다는 것을 볼 수 있습니다. 구름을 기준으로 빨간색 네 모칸을 보면 길이에 차이가 있습니다. 스코어도 잘 동작합니다.
또한 검은 기둥과 회색 기둥이 번갈아가며 등장하며 윗기둥만 움직이도록 구현했습니다.

<C# 소스코드 - GameManager>

```
using UnityEngine ;
public class GameManager : MonoBehaviour
{
    public GameObject objects;
    SpriteRenderer sr1, sr2;
    Sprite sprite1, sprite2, sprite3, sprite4;
    int count;
    public TextMesh scoreLabel;
    public static int score;
    public int Score
    {
        set
        {
            score =value ;
            scoreLabel .text =Score .ToString ();
        }
        get
        {
            return score ;
        }
    }
    void Start ()
    {
        score =0 ;
        count =0 ;
        sprite1 =Resources .Load <Sprite >("Sprites/InGame/01");
        sprite2 =Resources .Load <Sprite >("Sprites/InGame/02");
        sprite3 =Resources .Load <Sprite >("Sprites/InGame/03");
        sprite4 =Resources .Load <Sprite >("Sprites/InGame/04");
        InvokeRepeating ("CreateObjects", 1 ,2 );
    }
    void CreateObjects ()
    {
        count ++;
        GameObject ob =Instantiate (objects , new Vector3 (7.5f , Random .Range (-2f , 2.1f ) , 0 ) ,
Quaternion .identity );
        GameObject obj1 =ob .transform .GetChild (1 ).gameObject ;
        GameObject obj2 =ob .transform .GetChild (0 ).gameObject ;
        sr1 =obj1 .GetComponent <SpriteRenderer >();
        sr2 =obj2 .GetComponent <SpriteRenderer >();

        if (count %2 !=0 )
        {
            sr1 .sprite =sprite3 ;
            sr2 .sprite =sprite4 ;
        }
        else
        {
            sr1 .sprite =sprite1 ;
            sr2 .sprite =sprite2 ;
        }
    }
}
```

<구현 과정과 문제 해결>

과제 내용:

1. 윗 기둥이 아래위로 움직임(animation 이용)
2. 출현하는 기둥이 검은색과 흰색(또는 임의 색)이 번갈아 가면서 나타난다.

위 2가지 모두 해결했습니다. 해결하는 과정중에 나타났던 문제점들이 무엇이 있었는지와 그 해결방안은 어떻게 하였는지는 다음과 같습니다.

먼저 Object 안의 01 객체 즉, 윗기둥 객체에 애니메이션을 추가했습니다. Apply Root Motion을 적용했습니다. 이 Object를 GameManager에서 instantiate 함수를 통해 복제를 하면서 복제된 객체들은 모두 애니메이션 컴포넌트를 가지고 생성됩니다. 또한 기둥이 움직임에 따라 스코어도 같이 움직일 수 있도록 스코어에도 애니메이션을 추가했습니다. 스코어는 애니메이션에서 scale, position 둘 다 적용하여

알맞게 크기가 늘려지도록 조정했습니다.

그 다음은 기둥의 색을 번갈아가면서 바꿔야했는데, 그것을 기둥 객체의 Sprite Renderer 컴포넌트의 Sprite를 바꿔서 해결했습니다. count 변수를 추가하여 객체가 복제될 때마다 증가하도록 했는데 카운트 값에 따라 sprite를 계속 변경해주었습니다. 예를 들어 홀수 일때만 변경하게 되면 복제되는 오브젝트의 값이 해당 sprite로 고정되어 똑같은 색만 나오게 되어 홀수와 짝수 두 개의 경우 모두 변경해야했습니다. 그리고 원래의 object 객체에서 자식인 기둥 객체를 사용하게 되면, 게임을 시작할 때 첫 번째 기둥에서 색깔이 계속 바뀌는 현상이 있었습니다. 그 현상을 Instantiate 되는 객체를 GameObject obj로 설정하여 해당 객체에서 자식인 기둥 객체를 찾아 다시 GameObject obj1, obj2로 설정했습니다. 그리하여 첫 번째 기둥의 색깔이 계속 바뀌는 것을 해결할 수 있었습니다.

기둥의 색을 변경하기 위해서 Sprite 변경을 한다고 위에서 언급을 했습니다. 그러기 위해서는 새로운 색깔을 가지고 있는 Sprite를 따로 생성해서 사용했습니다.

디테일하게 확인하던 중 게임 스코어가 정상적으로 집계되지 않는 것을 확인했습니다. 복제 대상인 오브젝트의 Score가 Destroy 됨으로써 나타나는 현상이었습니다. 첫 번째 기둥이 사라지고 난 다음에 생성되는 5번째 기둥 객체에선 Score 자식이 보이지 않았습니다. 그리하여 집계가 안되었다고 생각이 들어 PlayerScript.cs에서 OnTriggerEnter2D 함수에서 Destroy(col.gameObject); 부분을 주석처리 했더니 해결되었습니다. 이와 비슷한 현상으로 기둥 객체가 특정 개수 이후로 생성되지 않는 문제가 생겼었는데 그 문제는 ObjectMove.cs에서 Destroy(gameObject)를 public GameObject objects를 새로 만들어 Destroy(objects);로 변경해주었더니 해결되었습니다.

<ObjectMove.cs>

```
using UnityEngine ;

public class ObjectMove : MonoBehaviour
{
    public GameObject objects;
    void FixedUpdate ()
    {
        transform .position =new Vector3 (transform .position .x -0.03f , transform .position .y , 0 );
        if (transform .position .x <=-7.5f )
        {
            Destroy (objects );
        }
    }
}
```