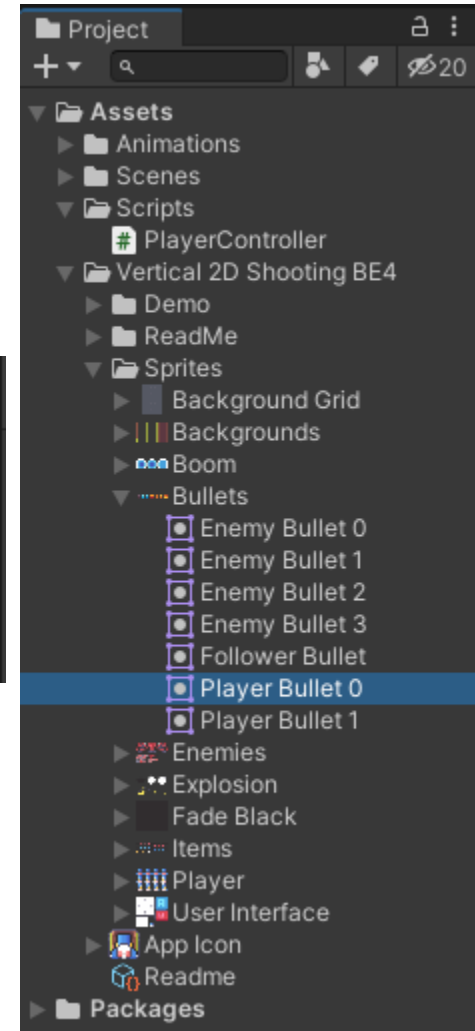
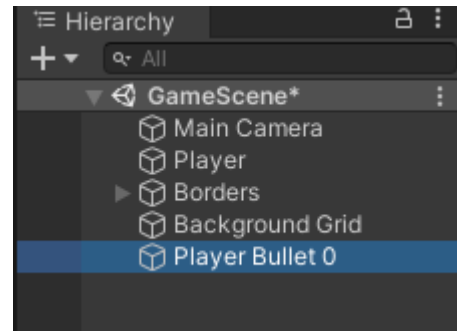
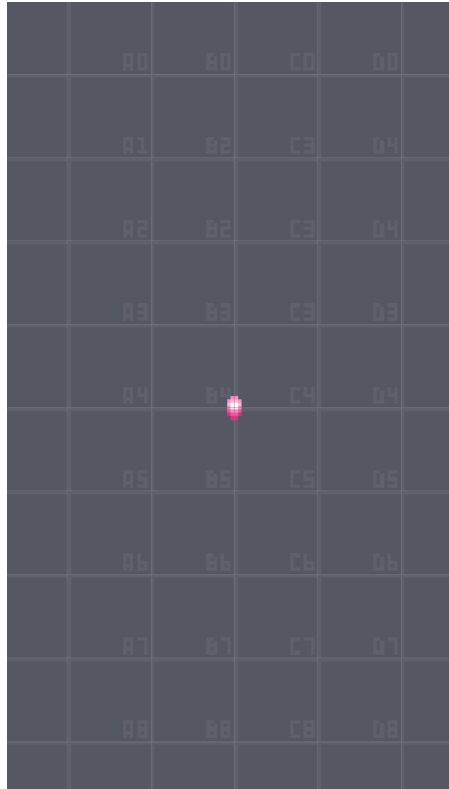


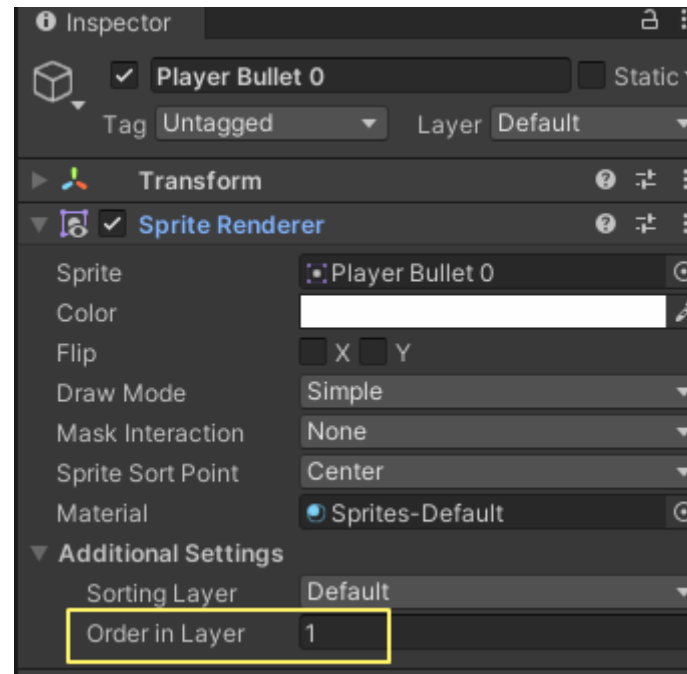
SpaceShooter2D_02

(총알 발사)

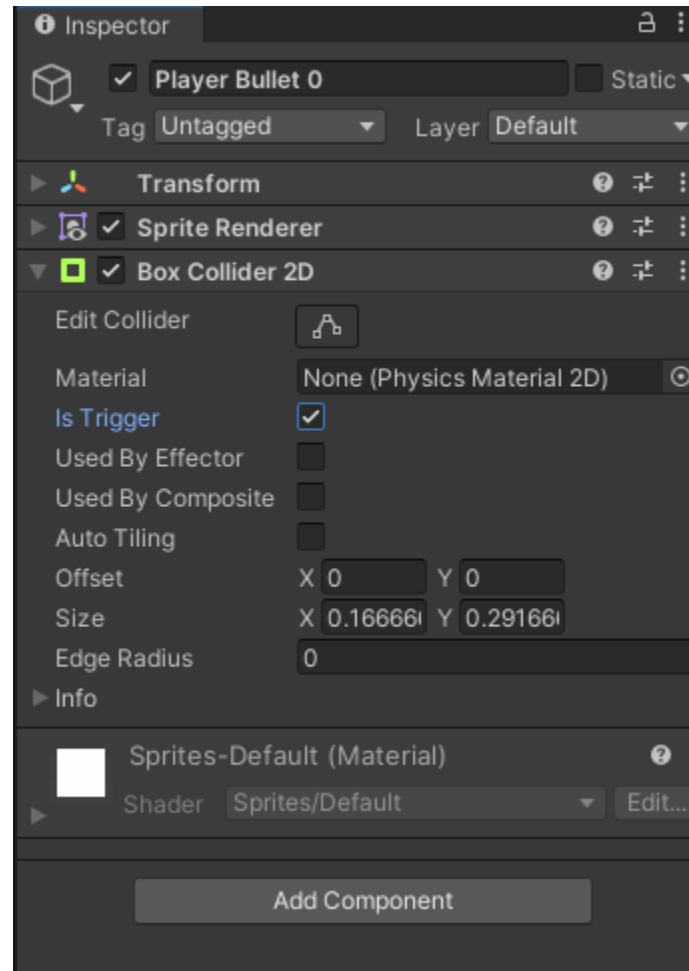
Project창에서 Player Bullet 0을 씬으로 가져 옵니다



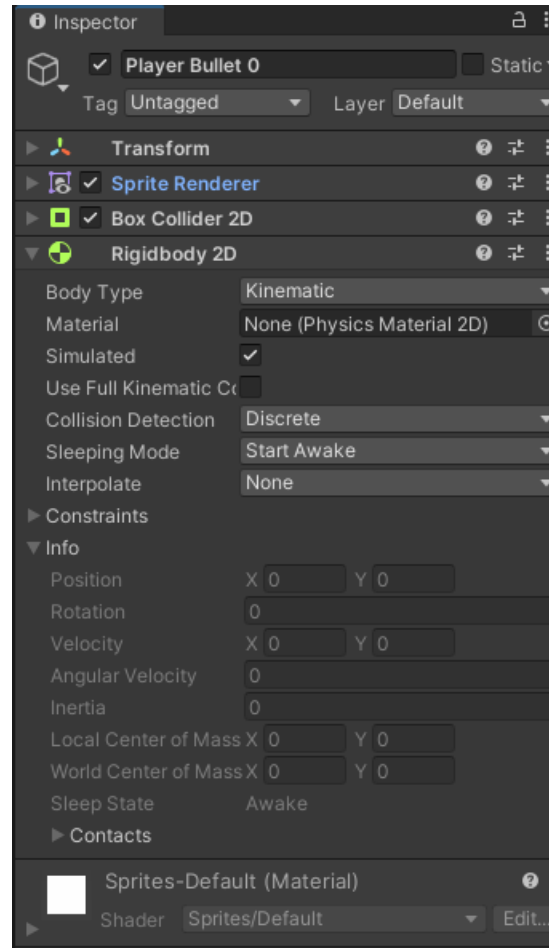
SpriteRenderer컴포넌트의 Order in Layer를 1로 변경 합니다



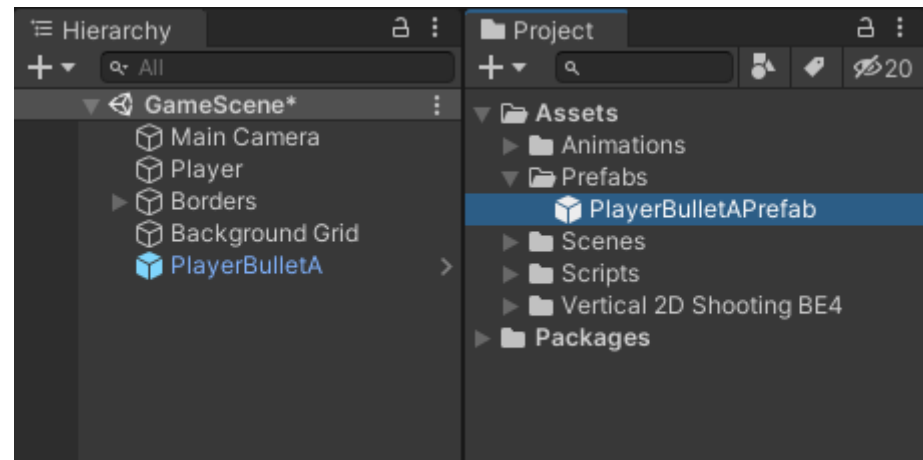
BoxCollider2D 컴포넌트를 부착 하고 isTrigger를 체크 합니다



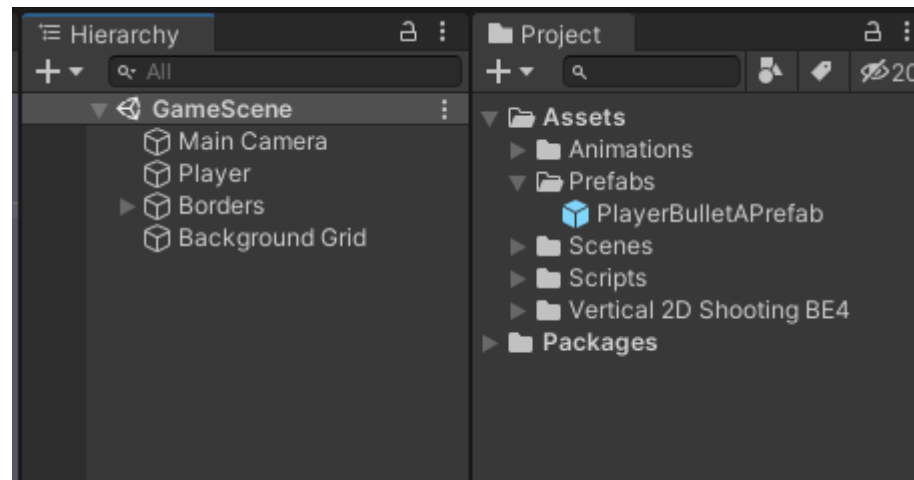
Rigidbody2D컴포넌트를 부착하고 BodyType을 Kinematic으로 변경 합니다



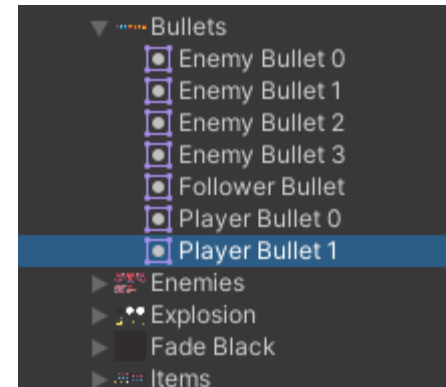
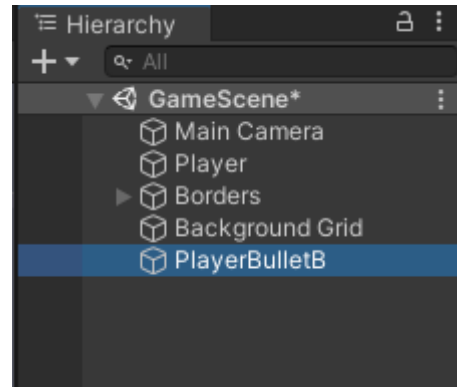
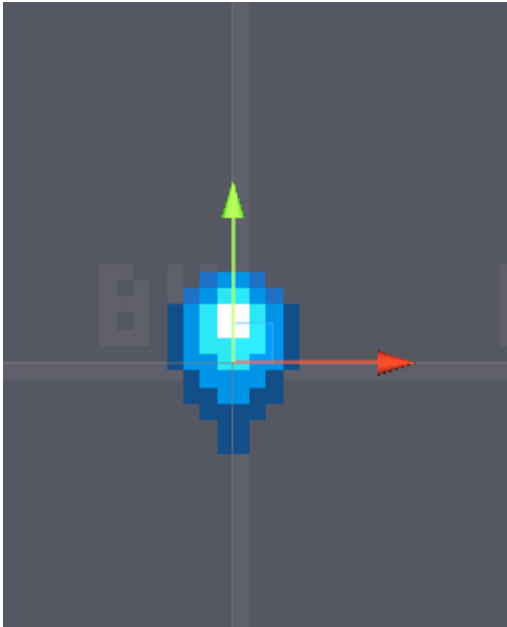
Prefabs폴더를 만들고 PlayerBulletA를 드레그&드롭후 프리팹으로 만들어 줍니다
이름을 PlayerBulletAPrefab이라고 수정합니다



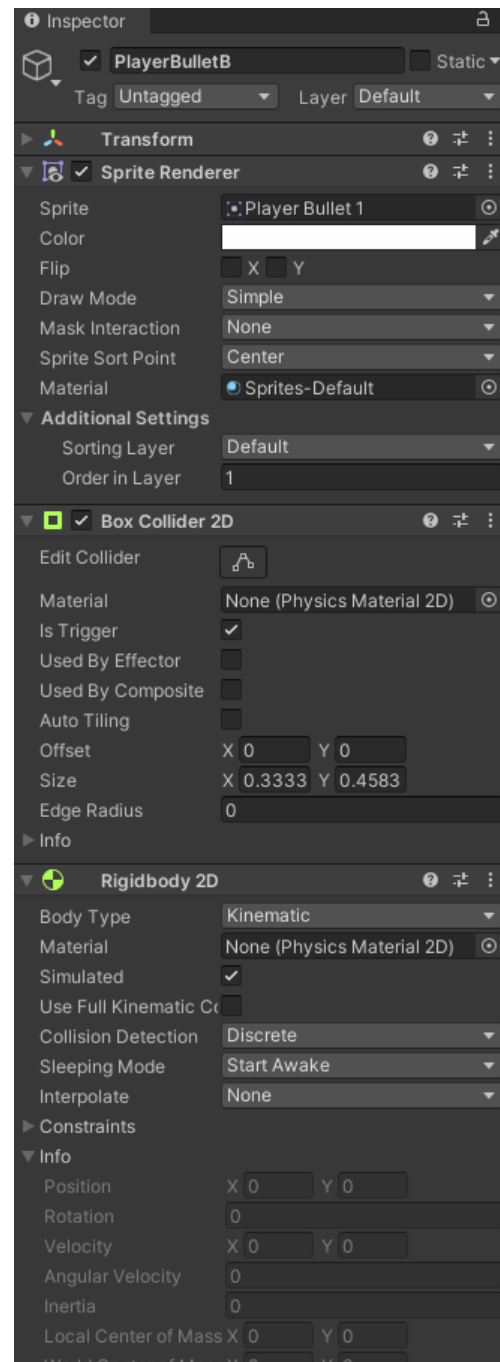
씬에서 PlayerBulletA를 제거 합니다



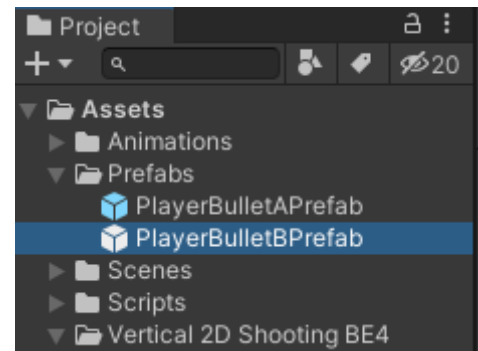
Project창에서 Player Bullet 1을 씬으로 가져 옵니다
이름을 PlayerBulletB로 변경 합니다



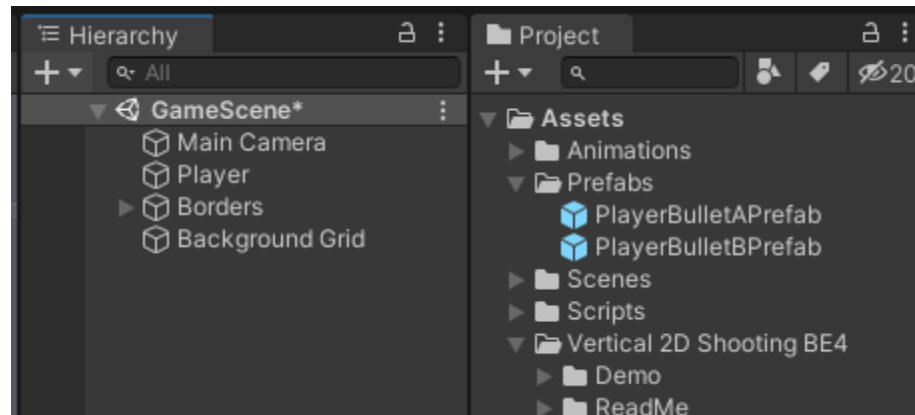
PlayerBulletA와 동일한 방법으로
Sprite Renderer의 Order in Layer를 1로 설정 하고
BoxCollider2D를 붙이고 isTrigger를 체크 합니다
Rigidbody2D 컴포넌트를 부착하고
BodyType을 Kinematic으로 설정 합니다



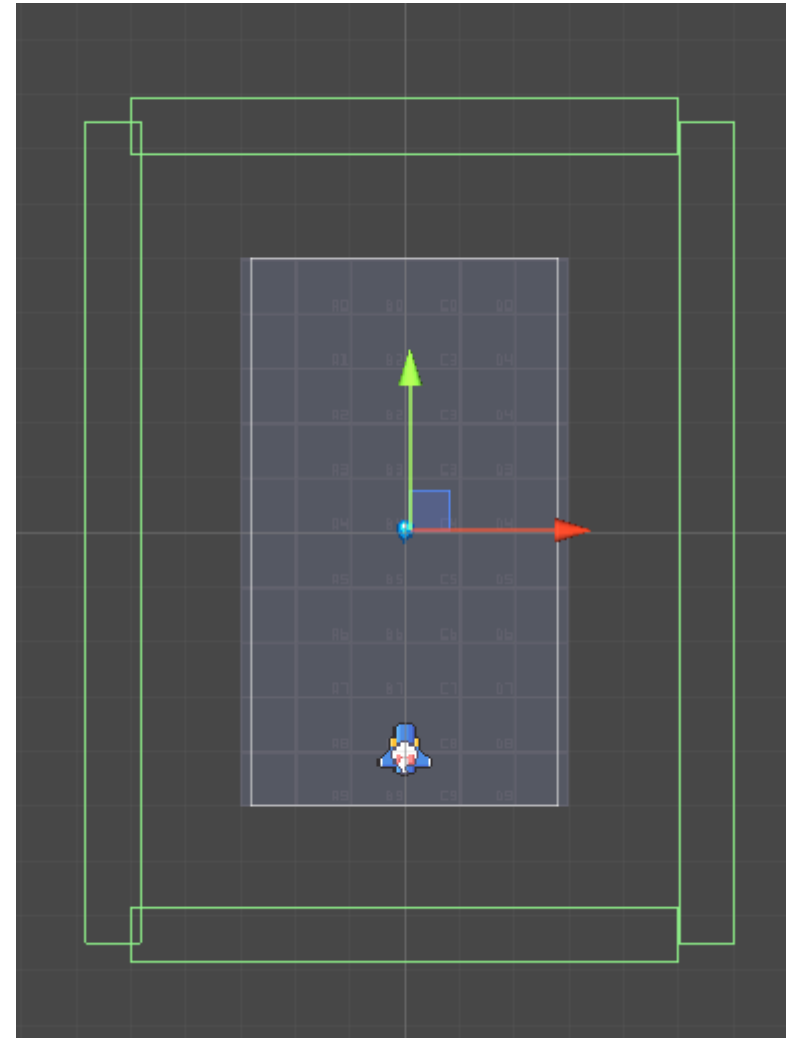
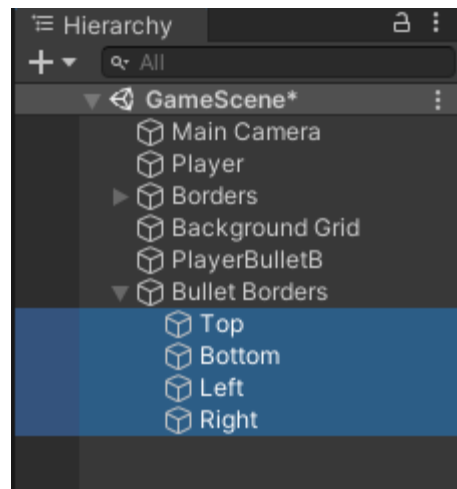
PlayerBulletB 게임오브젝트를 프리팹으로 만들어 주고
이름을 PlayerBulletBPrefab으로 변경합니다



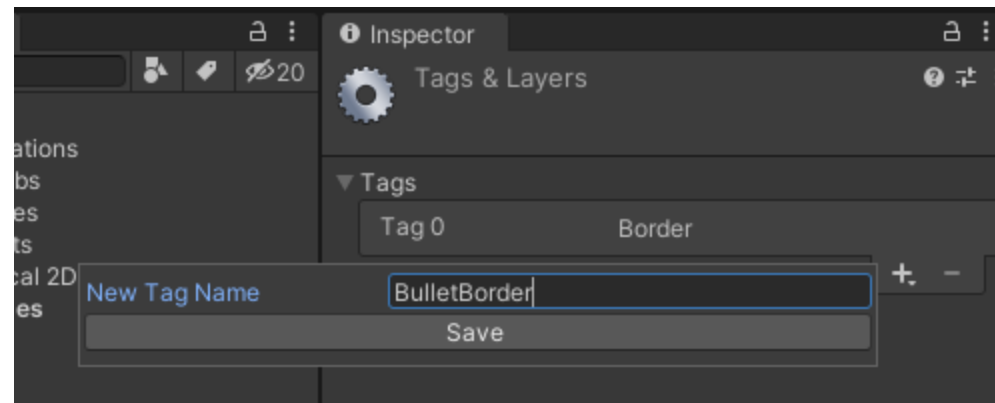
PlayerBulletB를 씬에서 지워줍니다



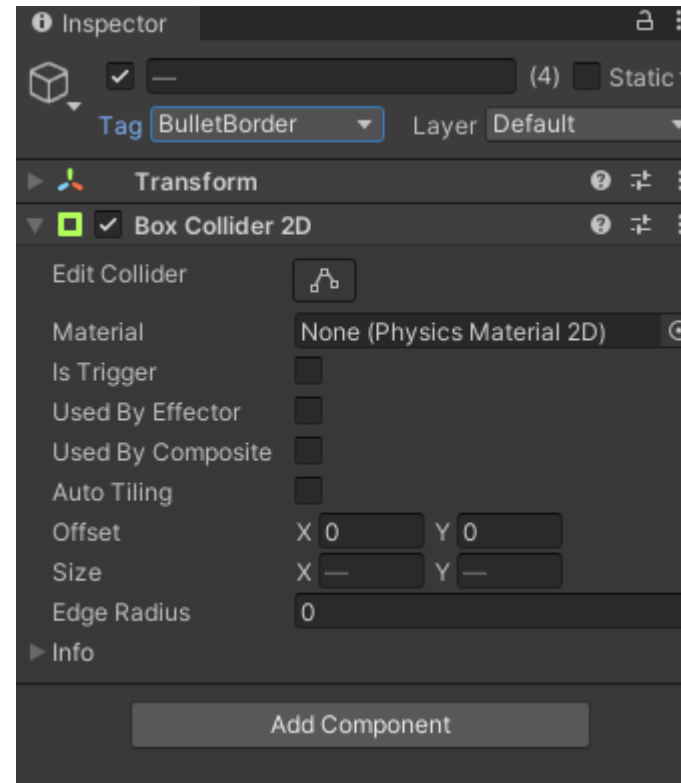
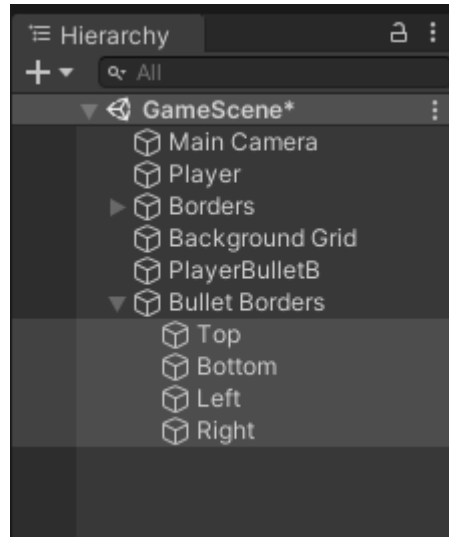
**Borders 게임오브젝트를 복사해서
이름을 Bullet Borders로 수정하고
크기와 위치를 조절 합니다**



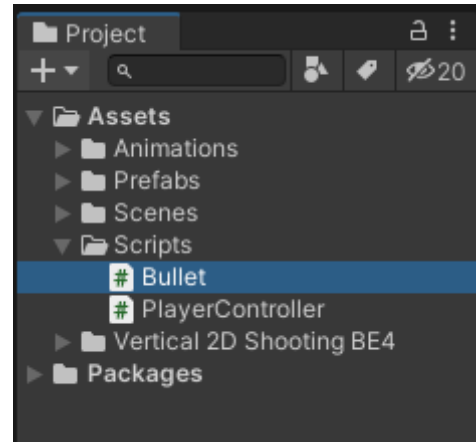
BulletBorder 태그를 추가 합니다



Top, Bottom, Left, Right게임오브젝트를 선택해 Tag를 BulletBorder로 변경 합니다



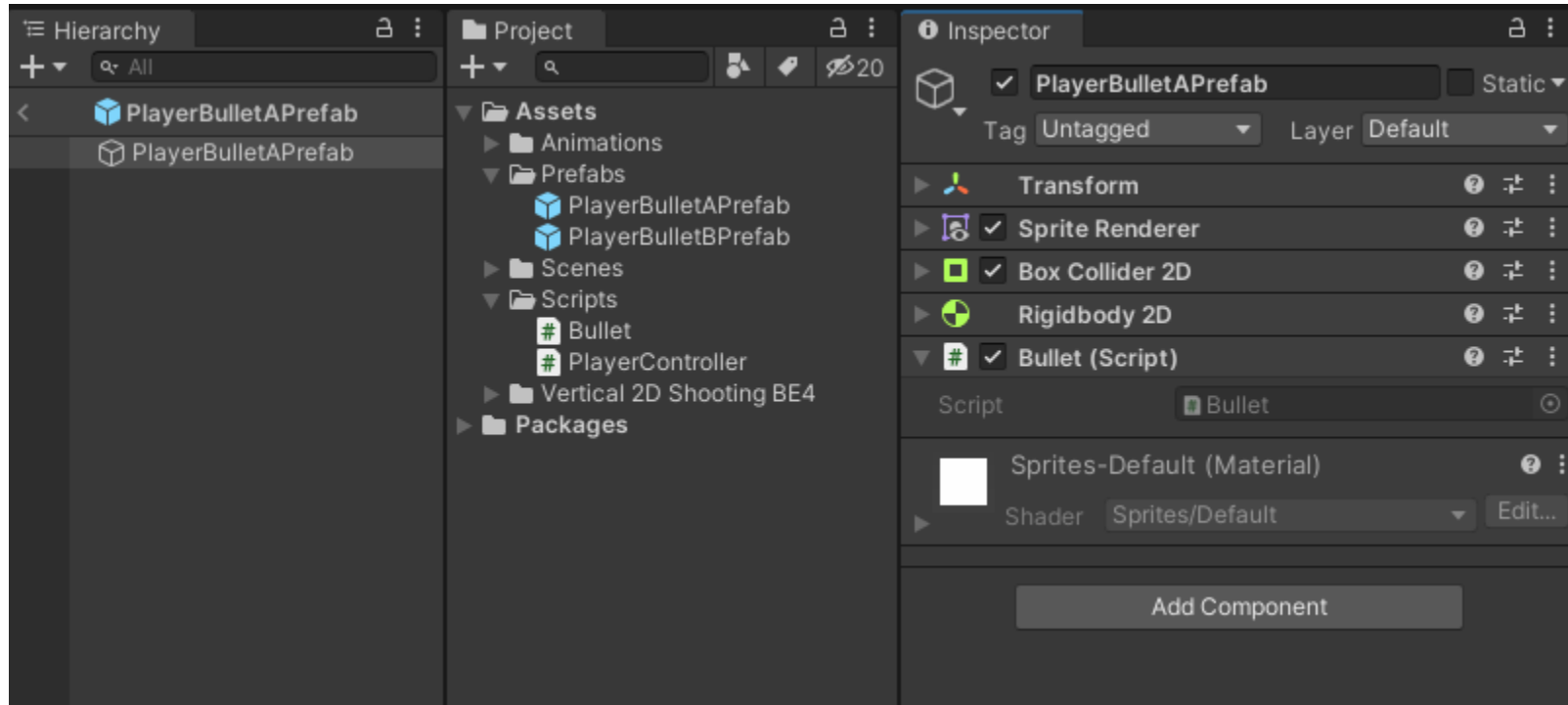
Bullet 스크립트를 생성 합니다



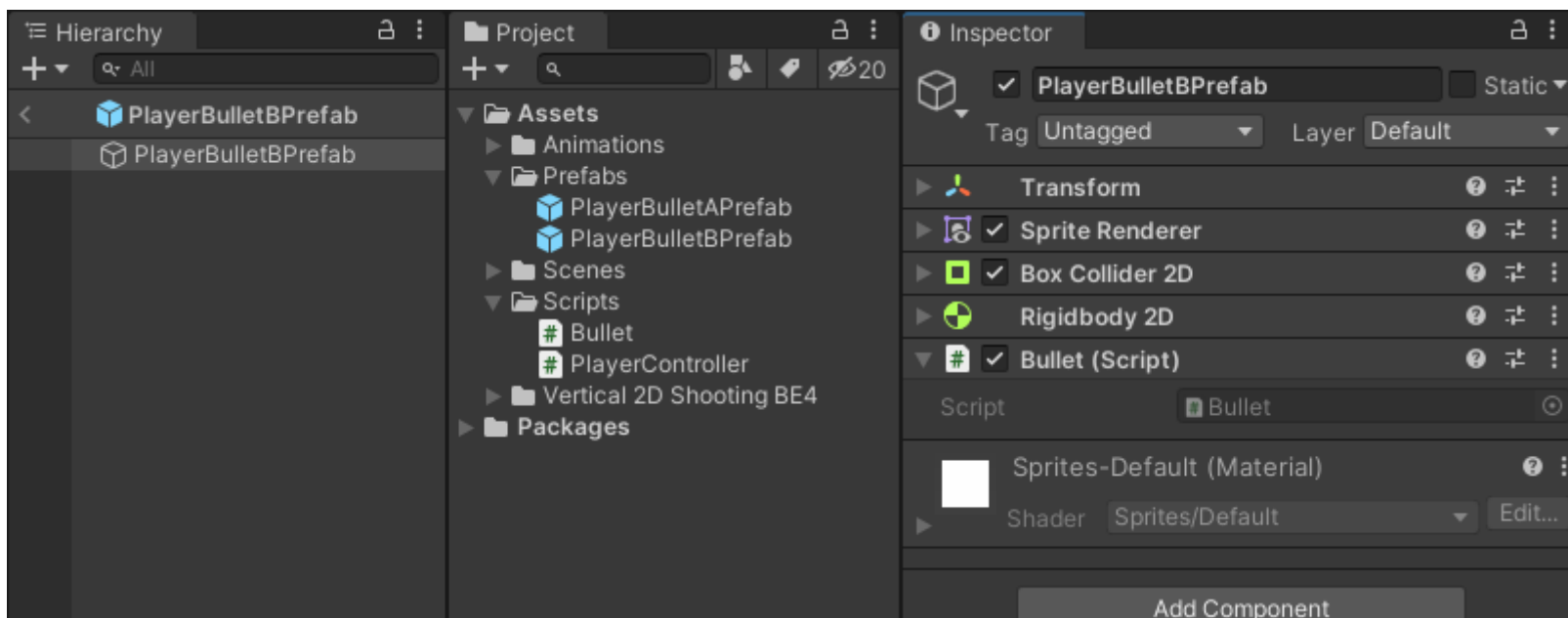
Bullet 스크립트를 수정 합니다

```
1  using UnityEngine;
2
3  public class Bullet : MonoBehaviour
4  {
5      float speed = 2f;
6
7      void Update()
8      {
9          Vector3 movement = Vector2.up * this.speed * Time.deltaTime;
10         this.transform.Translate(movement);
11     }
12
13     private void OnTriggerEnter2D(Collider2D collision)
14     {
15         if (collision.gameObject.tag == "BulletBorder")
16         {
17             Destroy(this.gameObject);
18         }
19     }
20 }
21
```


PlayerBulletAPrefab을 더블클릭해 프리팹 모드로 들어와 Bullet컴포넌트를 부착 합니다



PlayerBulletBPrefab을 더블클릭해 프리팹 모드로 들어와 Bullet컴포넌트를 부착 합니다



PlayerBulletAPrefab과 PlayerBulletBPrefab을 씬으로 가져와 적당히 배치하고 실행합니다
총알이 잘 앞으로 나가는지, BulletBorder와 충돌하면 제거 되는지 확인 합니다



PlayerController스크립트를 수정 합니다

```
public class PlayerController : MonoBehaviour
{
    public float speed = 1;
    private bool isTouchTop;
    private bool isTouchBottom;
    private bool isTouchLeft;
    private bool isTouchRight;

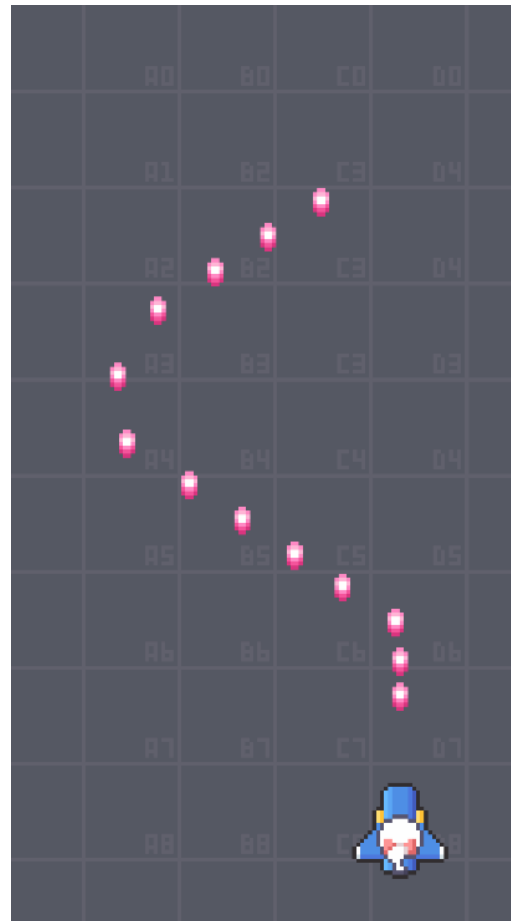
    public GameObject playerBulletAPrefab;
    public GameObject playerBulletBPrefab;

    Animator anim;
```

PlayerController스크립트를 수정 합니다

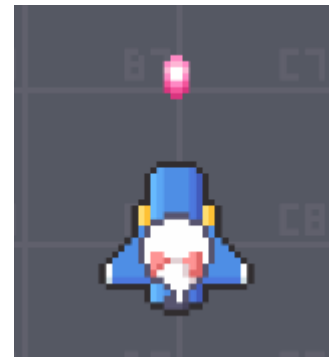
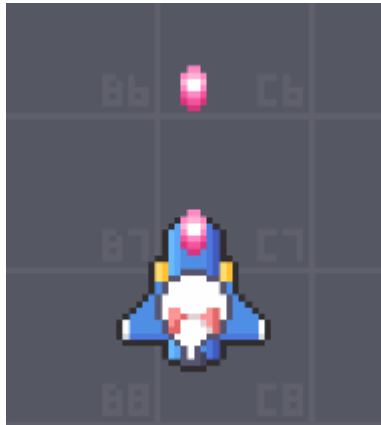
```
30 float v = Input.GetAxisRaw("Vertical");
31
32 if ((this.isTouchTop && v == 1) || (this.isTouchBottom && v == -1))
33 {
34     v = 0;
35 }
36
37 Vector3 currentPos = this.transform.position;
38 Vector3 nextPos = new Vector3(h, v, 0).normalized * this.speed * Time.deltaTime;
39 this.transform.position = currentPos + nextPos;
40
41 if (Input.GetButtonDown("Horizontal") || Input.GetButtonUp("Horizontal"))
42 {
43     this.anim.SetInteger("Input", (int)h);
44 }
45
46
47 if (Input.GetKeyDown(KeyCode.Space))
48 {
49     Instantiate(this.playerBulletAPrefab, this.transform.position, this.transform.rotation);
50 }
51 }
52
53 @ Unity 메시지 | 참조 0개
54 private void OnTriggerEnter2D(Collider2D collision)
55 {
56     if (collision.gameObject.tag == "Border")
57     {
```

실행후 결과를 확인해요
Player를 이동하며 총알을 발사해봅니다



플레이어가 발사하는 총알의 위치를 조절하려면 어떻게 해야 할까요?

직접 수정해 봅니다

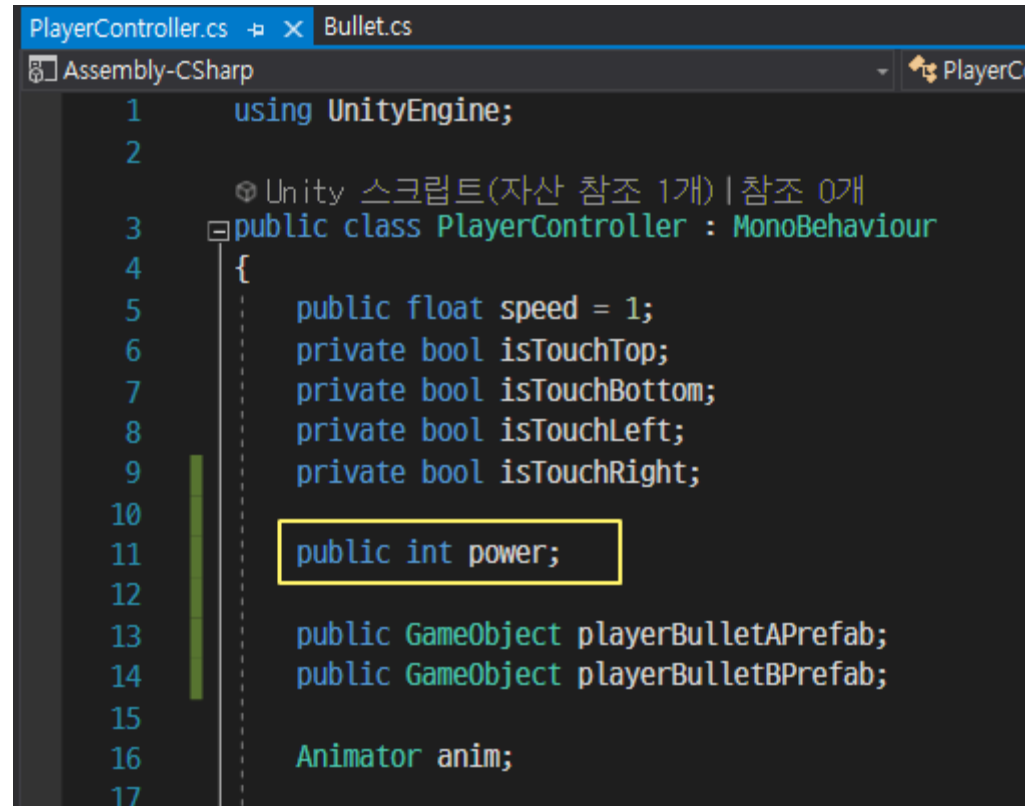


플레이어 총알의 위치를 설정할때 y축으로 약간 올려줍니다

```
if (Input.GetKeyDown(KeyCode.Space))  
{  
    Instantiate(this.playerBulletAPrefab, this.transform.position + Vector3.up * 0.7f, this.transform.rotation);  
}
```

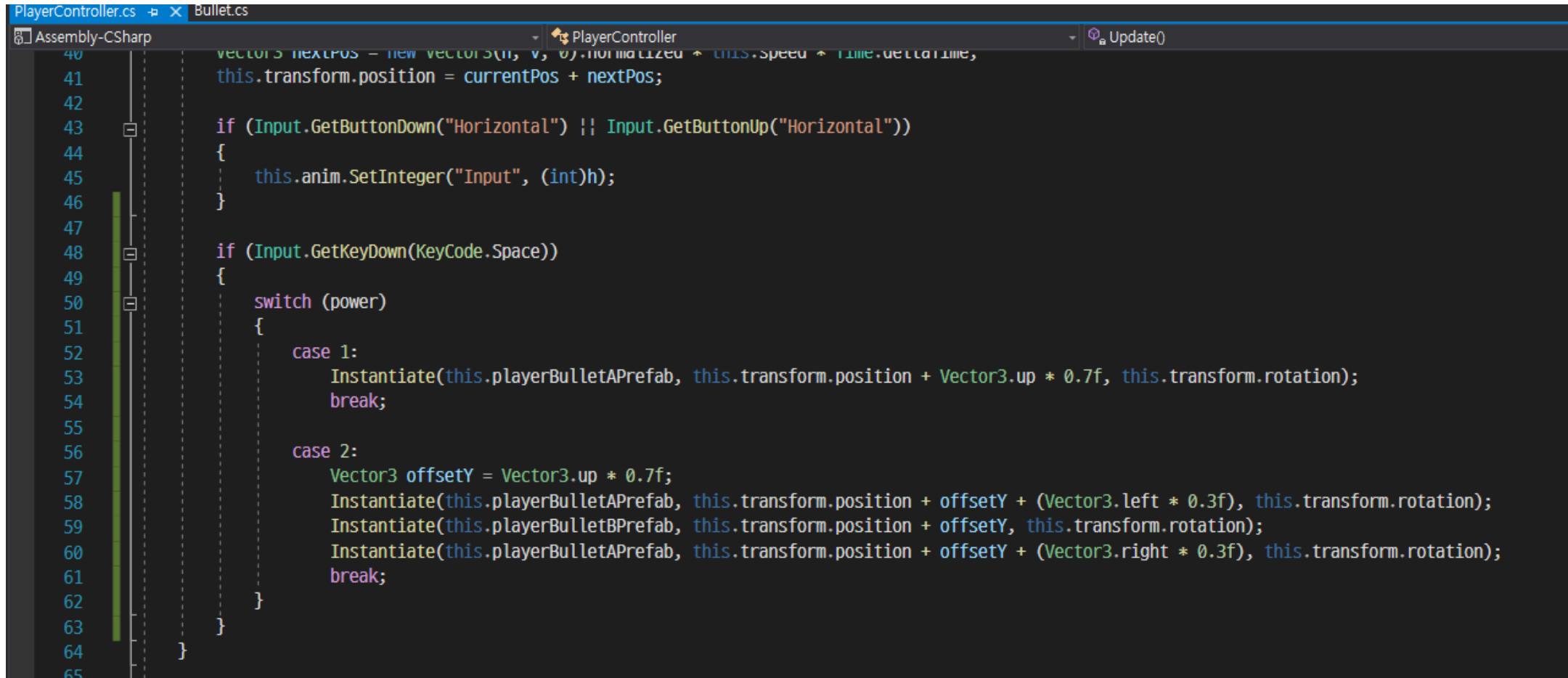

파워에 따라 발사되는 총알 다르게 하기

PlayerController스크립트를 수정 합니다



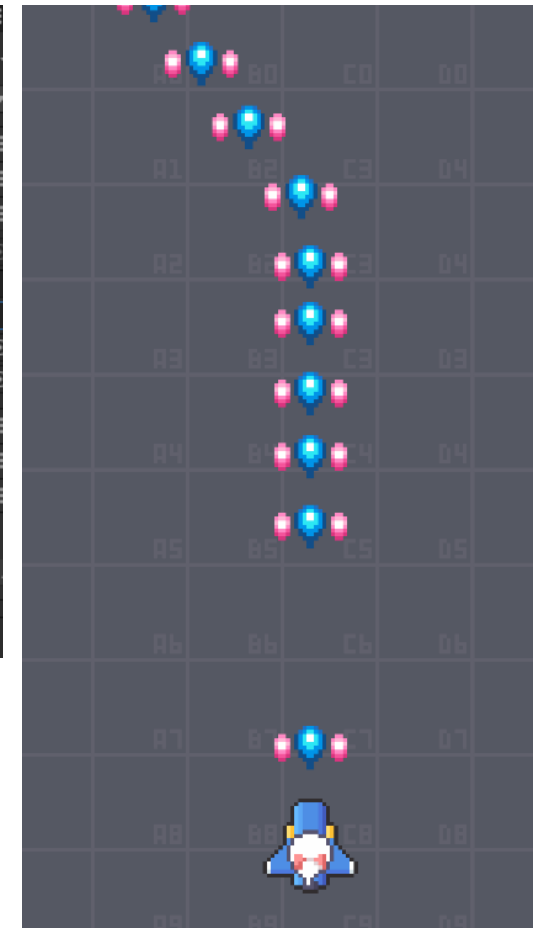
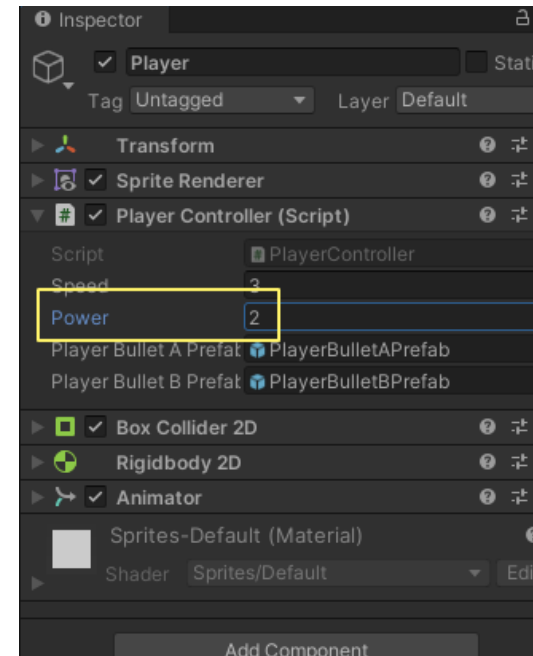
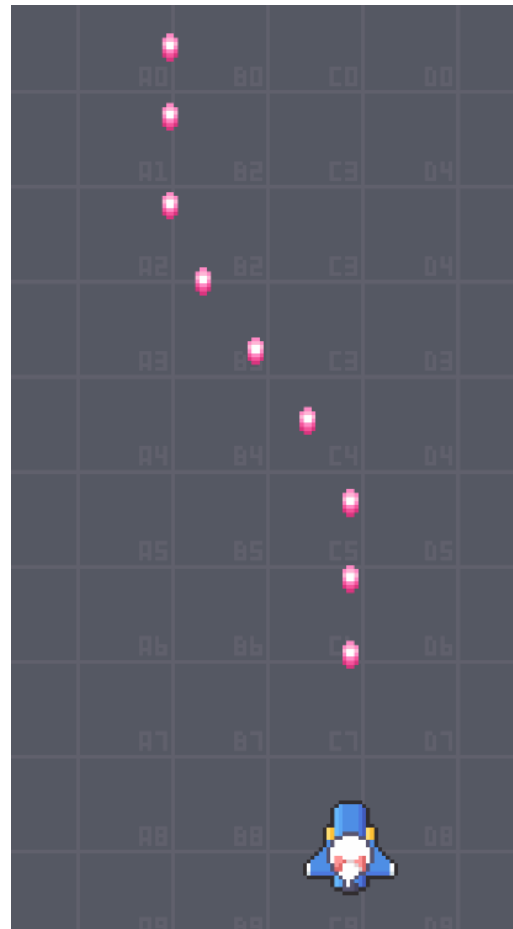
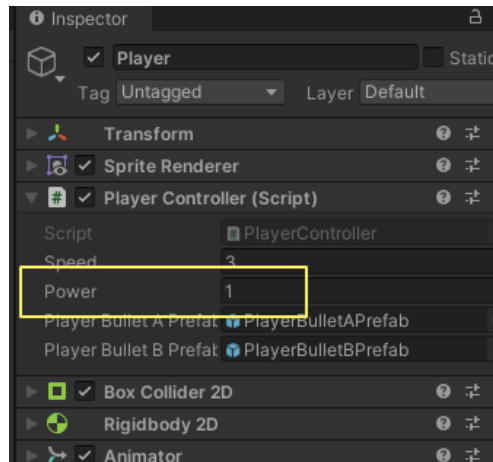
```
1 using UnityEngine;
2
3 Unity 스크립트(자산 참조 1개) | 참조 0개
4 public class PlayerController : MonoBehaviour
5 {
6     public float speed = 1;
7     private bool isTouchTop;
8     private bool isTouchBottom;
9     private bool isTouchLeft;
10    private bool isTouchRight;
11    public int power;
12
13    public GameObject playerBulletAPrefab;
14    public GameObject playerBulletBPrefab;
15
16    Animator anim;
17
```

PlayerController스크립트를 수정 합니다

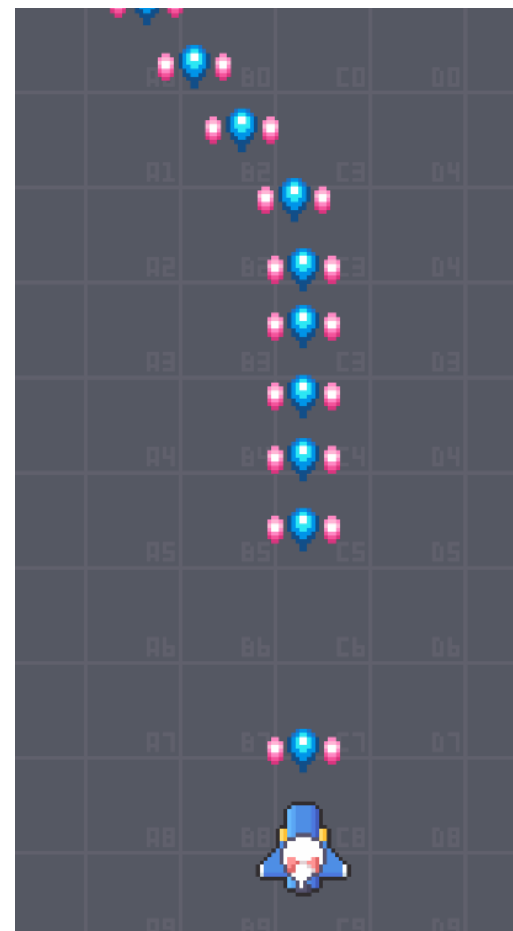
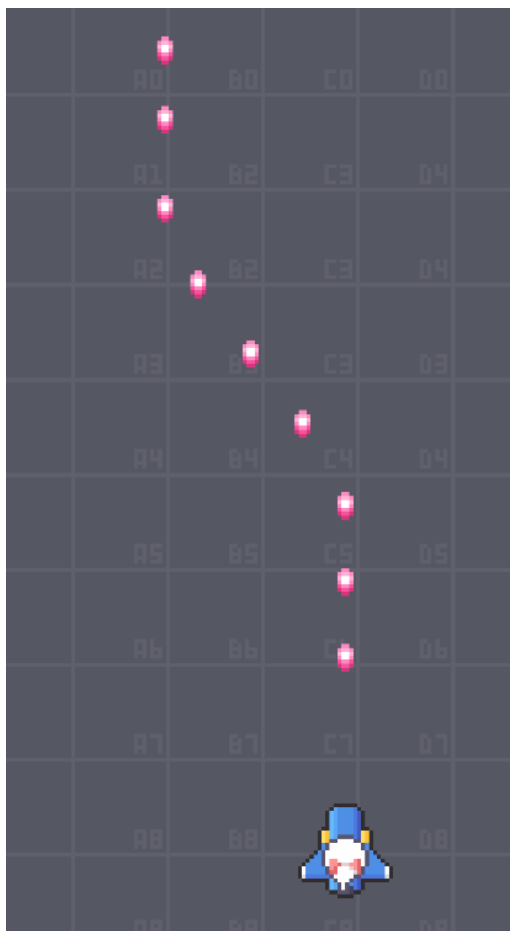


```
40  Vector3 nextPos = new Vector3(h, v, 0).normalized * this.Speed * Time.deltaTime;
41  this.transform.position = currentPos + nextPos;
42
43  if (Input.GetButtonDown("Horizontal") || Input.GetButtonUp("Horizontal"))
44  {
45      this.anim.SetInteger("Input", (int)h);
46  }
47
48  if (Input.GetKeyDown(KeyCode.Space))
49  {
50      switch (power)
51      {
52          case 1:
53              Instantiate(this.playerBulletAPrefab, this.transform.position + Vector3.up * 0.7f, this.transform.rotation);
54              break;
55
56          case 2:
57              Vector3 offsetY = Vector3.up * 0.7f;
58              Instantiate(this.playerBulletAPrefab, this.transform.position + offsetY + (Vector3.left * 0.3f), this.transform.rotation);
59              Instantiate(this.playerBulletBPrefab, this.transform.position + offsetY, this.transform.rotation);
60              Instantiate(this.playerBulletCPrefab, this.transform.position + offsetY + (Vector3.right * 0.3f), this.transform.rotation);
61              break;
62      }
63  }
64  }
65
```

실행후 결과를 확인 합니다



Power에 따라 1단계, 2단계, 3단계가 되도록 코드를 직접 수정해봅니다



```
if (Input.GetKeyDown(KeyCode.Space))
{
    Vector3 offsetY = Vector3.up * 0.7f;

    switch (power)
    {
        case 1:
            Instantiate(this.playerBulletAPrefab, this.transform.position + Vector3.up * 0.7f, this.transform.rotation);
            break;

        case 2:
            Instantiate(this.playerBulletAPrefab, this.transform.position + offsetY + (Vector3.left * 0.2f), this.transform.rotation);
            Instantiate(this.playerBulletAPrefab, this.transform.position + offsetY + (Vector3.right * 0.2f), this.transform.rotation);
            break;

        case 3:
            Instantiate(this.playerBulletAPrefab, this.transform.position + offsetY + (Vector3.left * 0.3f), this.transform.rotation);
            Instantiate(this.playerBulletBPrefab, this.transform.position + offsetY, this.transform.rotation);
            Instantiate(this.playerBulletAPrefab, this.transform.position + offsetY + (Vector3.right * 0.3f), this.transform.rotation);
            break;
    }
}
```