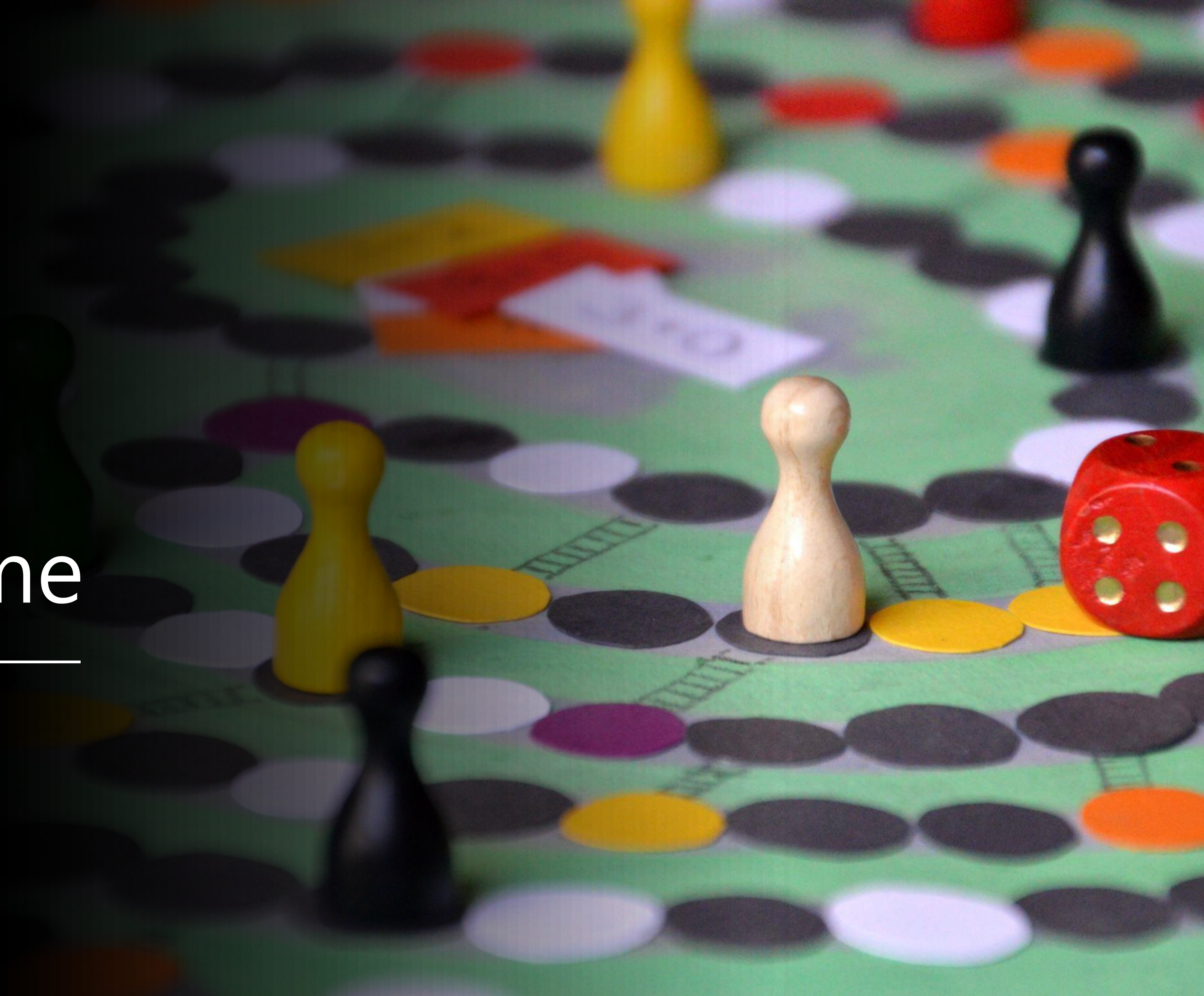
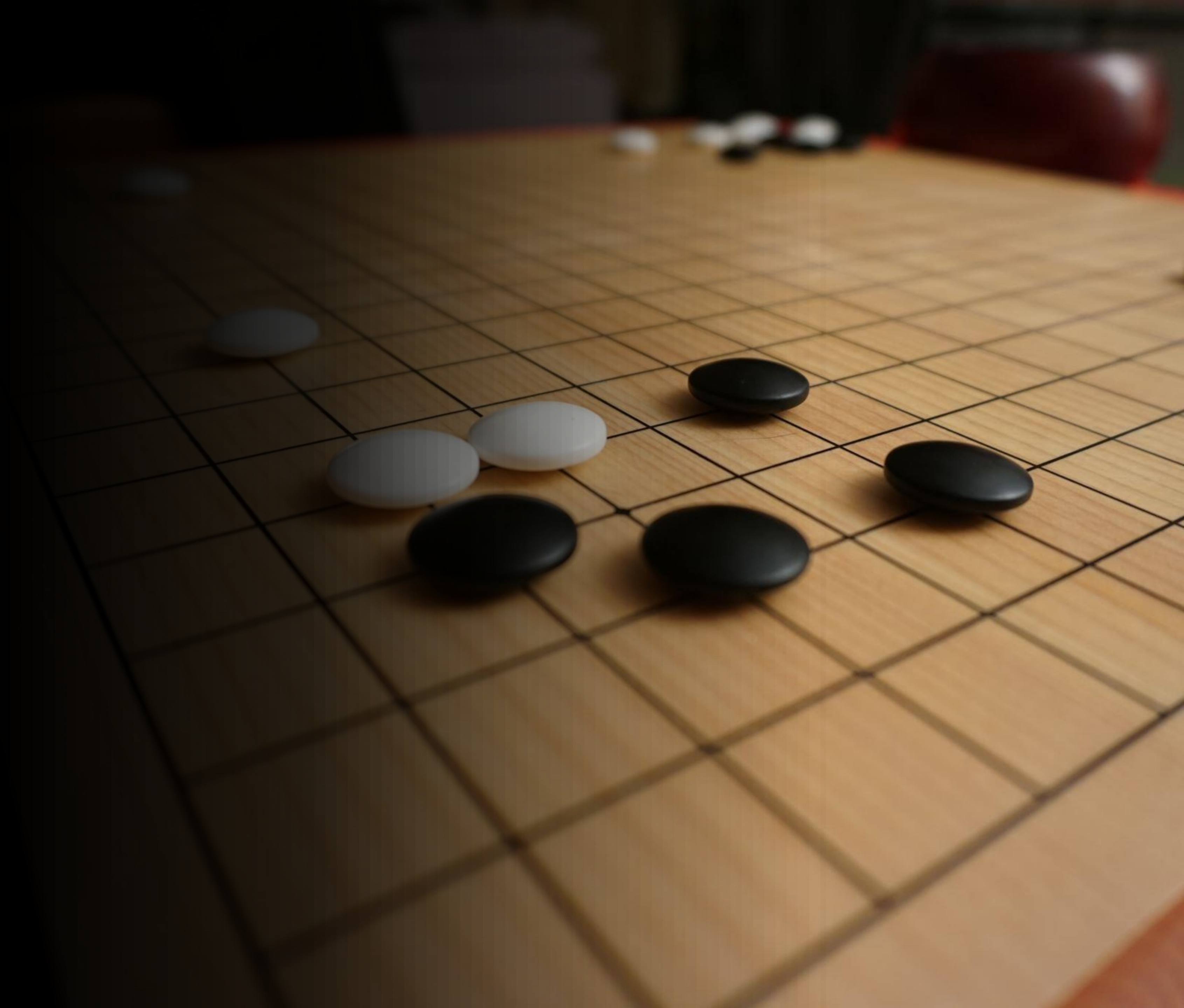


Small Gimmic Game

게임공학과 1588026 유승효

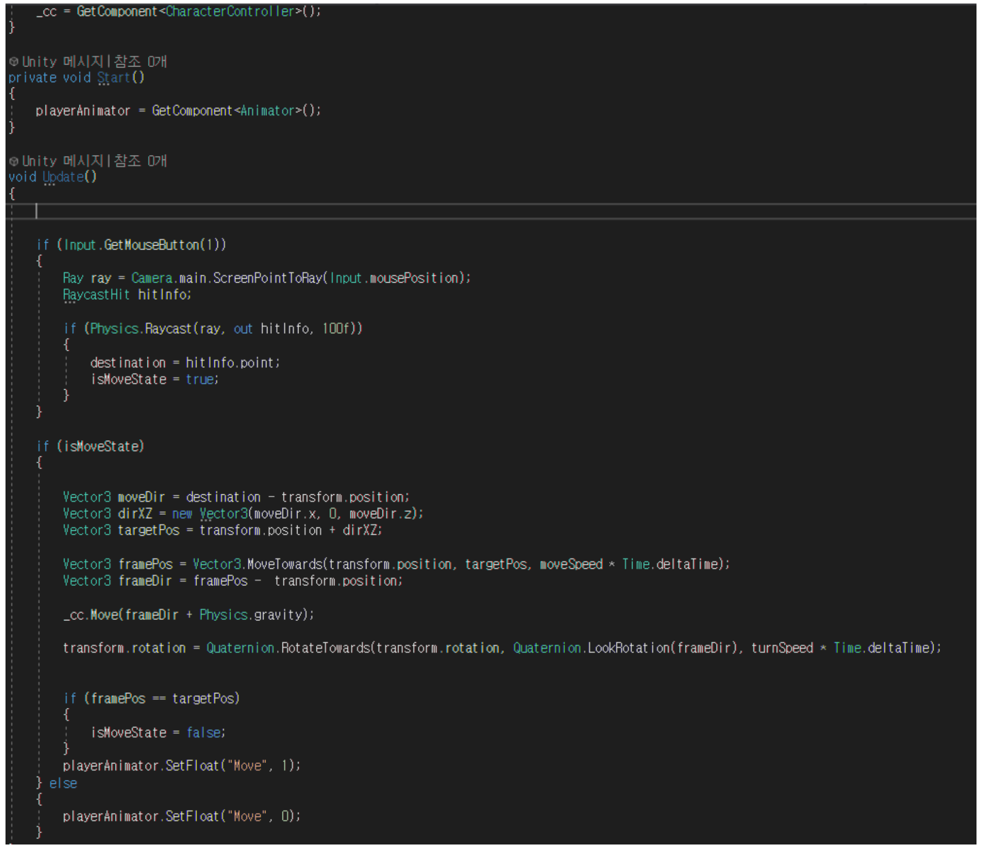


목차

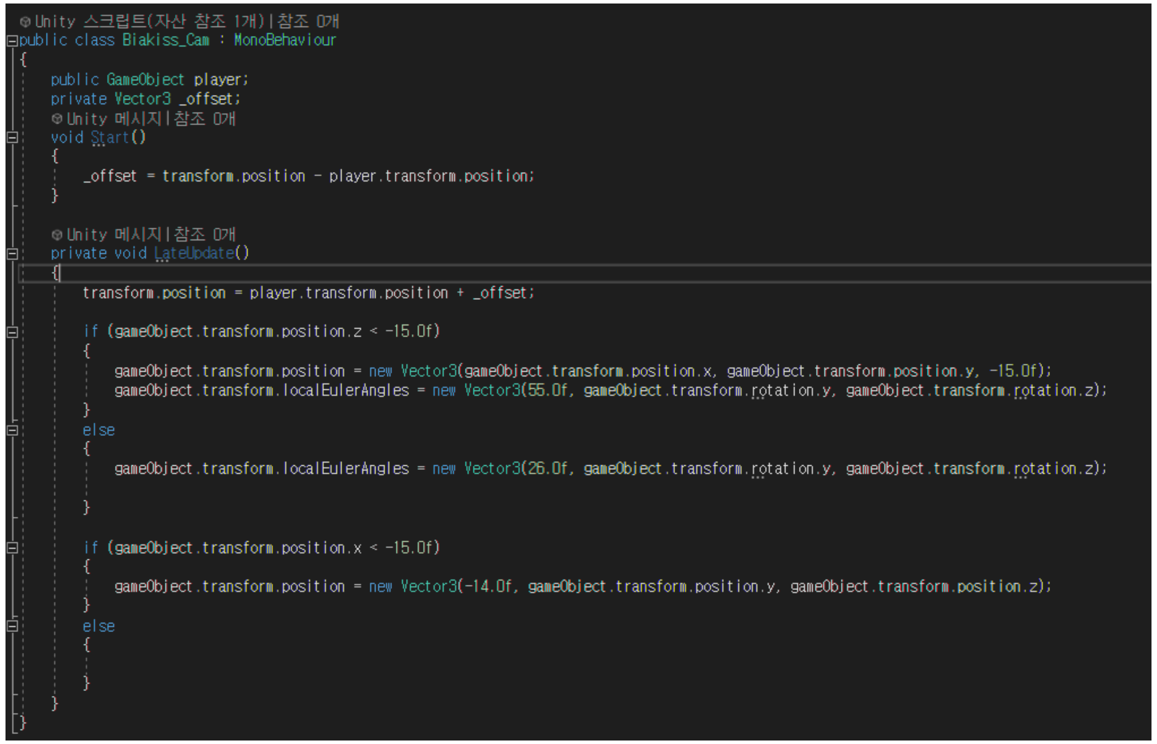
구슬넣기

- 구슬을 유인해 알맞은 색상의 문에 넣는것을 목표로 하는 게임입니다.
- 서술할 코드로는 플레이어 , 공에 들어간 코드, 스폰 포인트, 기타 그래픽적 요소 입니다.

플레이어

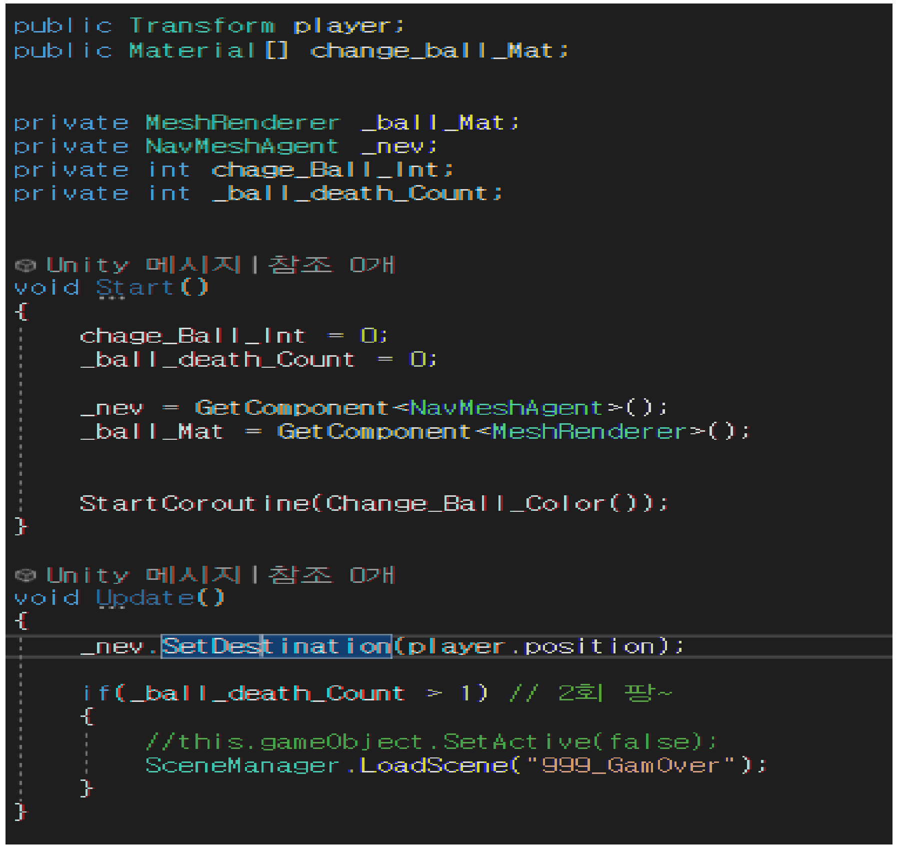


```
    _cc = GetComponent<CharacterController>();
}

Unity 메시지 | 참조 0개
private void Start()
{
    playerAnimator = GetComponent<Animator>();
}

Unity 메시지 | 참조 0개
void Update()
{
    

    if (Input.GetMouseButton(1))
    {
        Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);
        RaycastHit hitInfo;

        if (Physics.Raycast(ray, out hitInfo, 100f))
        {
            destination = hitInfo.point;
            isMoveState = true;
        }
    }

    if (isMoveState)
    {

        Vector3 moveDir = destination - transform.position;
        Vector3 dirXZ = new Vector3(moveDir.x, 0, moveDir.z);
        Vector3 targetPos = transform.position + dirXZ;

        Vector3 framePos = Vector3.MoveTowards(transform.position, targetPos, moveSpeed * Time.deltaTime);
        Vector3 frameDir = framePos -  transform.position;

        _cc.Move(frameDir + Physics.gravity);

        transform.rotation = Quaternion.RotateTowards(transform.rotation, Quaternion.LookRotation(frameDir), turnSpeed * Time.deltaTime);


        if (framePos == targetPos)
        {
            isMoveState = false;
        }
        playerAnimator.SetFloat("Move", 1);
    } else
    {
        playerAnimator.SetFloat("Move", 0);
    }
```

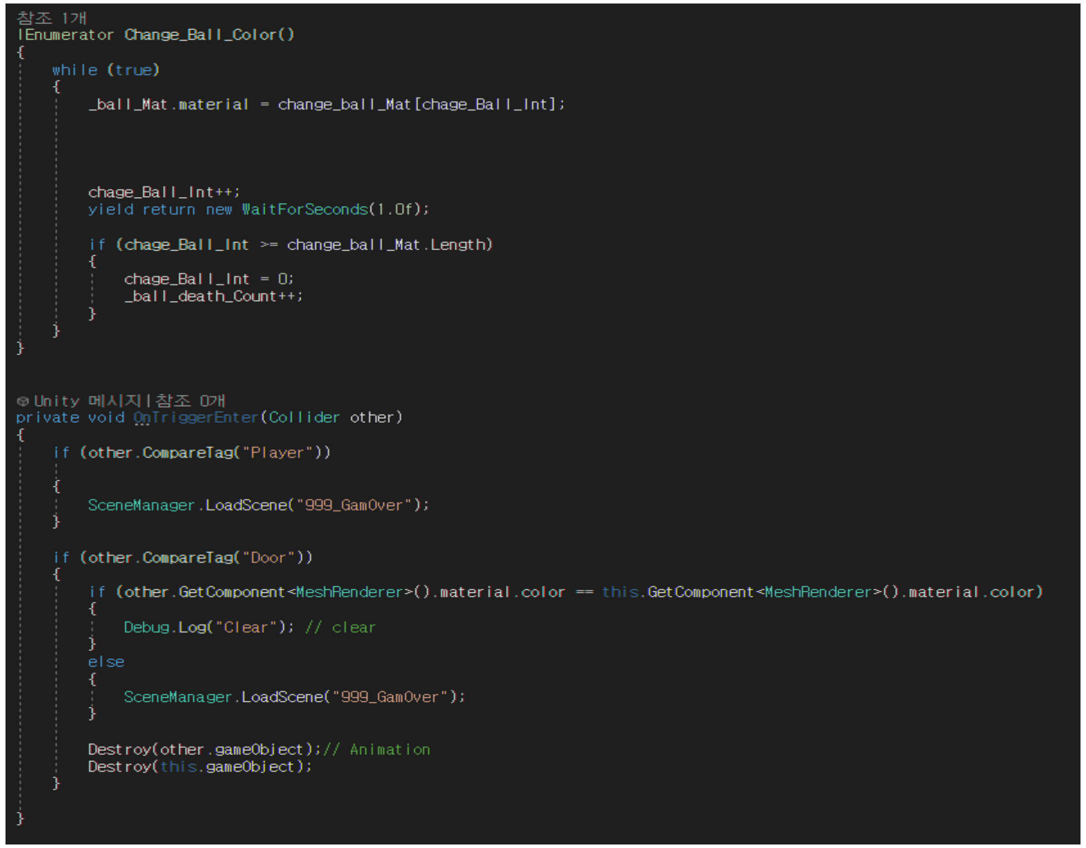


```
Unity 스크립트(자산 참조 1개) | 참조 0개
public class Biakiss_Cam : MonoBehaviour
{
    public GameObject player;
    private Vector3 _offset;
    Unity 메시지 | 참조 0개
    void Start()
    {
        _offset = transform.position - player.transform.position;
    }

    Unity 메시지 | 참조 0개
    private void LateUpdate()
    {
        transform.position = player.transform.position + _offset;

        if (gameObject.transform.position.z < -15.0f)
        {
            gameObject.transform.position = new Vector3(gameObject.transform.position.x, gameObject.transform.position.y, -15.0f);
            gameObject.transform.localEulerAngles = new Vector3(55.0f, gameObject.transform.rotation.y, gameObject.transform.rotation.z);
        }
        else
        {
            gameObject.transform.localEulerAngles = new Vector3(26.0f, gameObject.transform.rotation.y, gameObject.transform.rotation.z);

        }

        if (gameObject.transform.position.x < -15.0f)
        {
            gameObject.transform.position = new Vector3(-14.0f, gameObject.transform.position.y, gameObject.transform.position.z);
        }
        else
        {

        }
    }
}
```

공

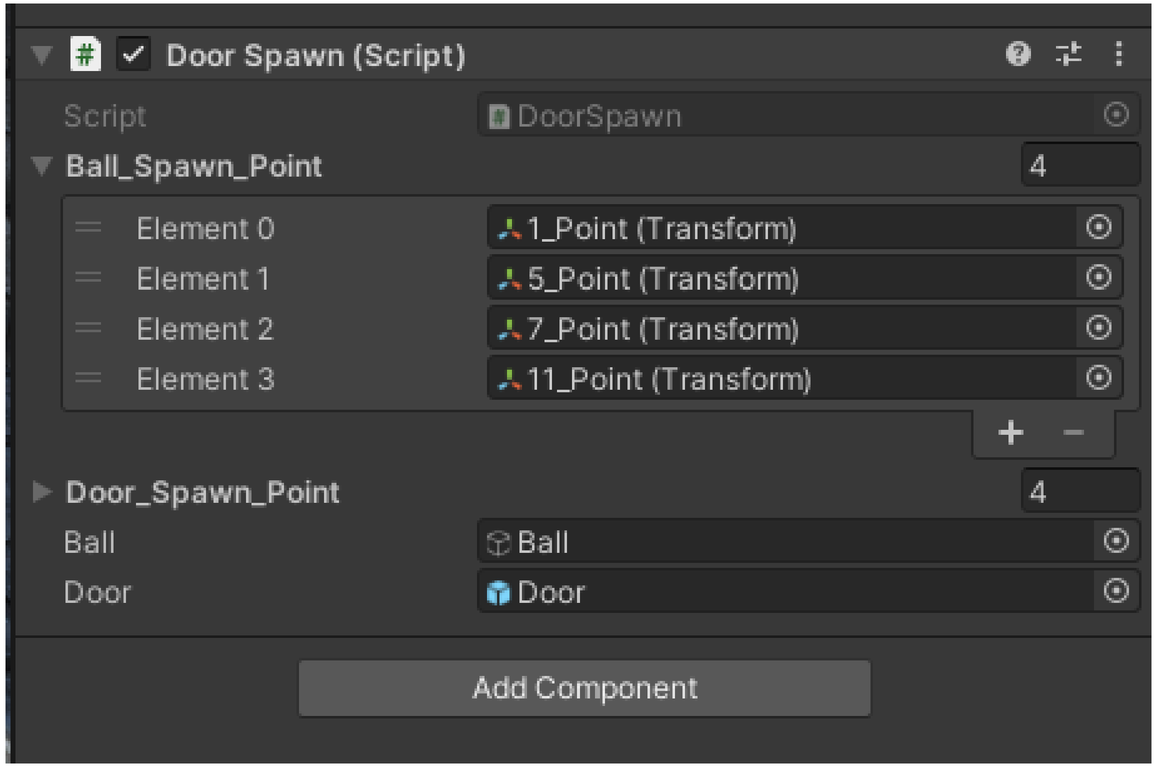


```
public Transform player;
public Material[] change_ball_Mat;


private MeshRenderer _ball_Mat;
private NavMeshAgent _nev;
private int chage_Ball_Int;
private int _ball_death_Count;

Unity 메시지|참조 0개
void Start()
{
    chage_Ball_Int  = 0;
    _ball_death_Count  = 0;

    _nev = GetComponent<NavMeshAgent>();
    _ball_Mat  = GetComponent<MeshRenderer>();


    StartCoroutine(Change_Ball_Color());
}

Unity 메시지|참조 0개
void Update()
{
    _nev.SetDestination(player.position);

    if(_ball_death_Count > 1) // 2회 팡~
    {
        //this.gameObject.SetActive(false);
        SceneManager.LoadScene("999_GamOver");
    }
}
```



```
참조 1개
IEnumerator Change_Ball_Color()
{
    while (true)
    {
        _ball_Mat.material = change_ball_Mat[chage_Ball_Int];




        chage_Ball_Int++;
        yield return new WaitForSeconds(1.0f);

        if (chage_Ball_Int >= change_ball_Mat.Length)
        {
            chage_Ball_Int = 0;
            _ball_death_Count++;
        }
    }
}


Unity 메시지|참조 0개
private void OnTriggerEnter(Collider other)
{
    if (other.CompareTag("Player"))

    {
        SceneManager.LoadScene("999_GamOver");
    }

    if (other.CompareTag("Door"))
    {
        if (other.GetComponent<MeshRenderer>().material.color == this.GetComponent<MeshRenderer>().material.color)
        {
            Debug.Log("Clear"); // clear
        }
        else
        {
            SceneManager.LoadScene("999_GamOver");
        }

        Destroy(other.gameObject);// Animation
        Destroy(this.gameObject);
    }

}
```

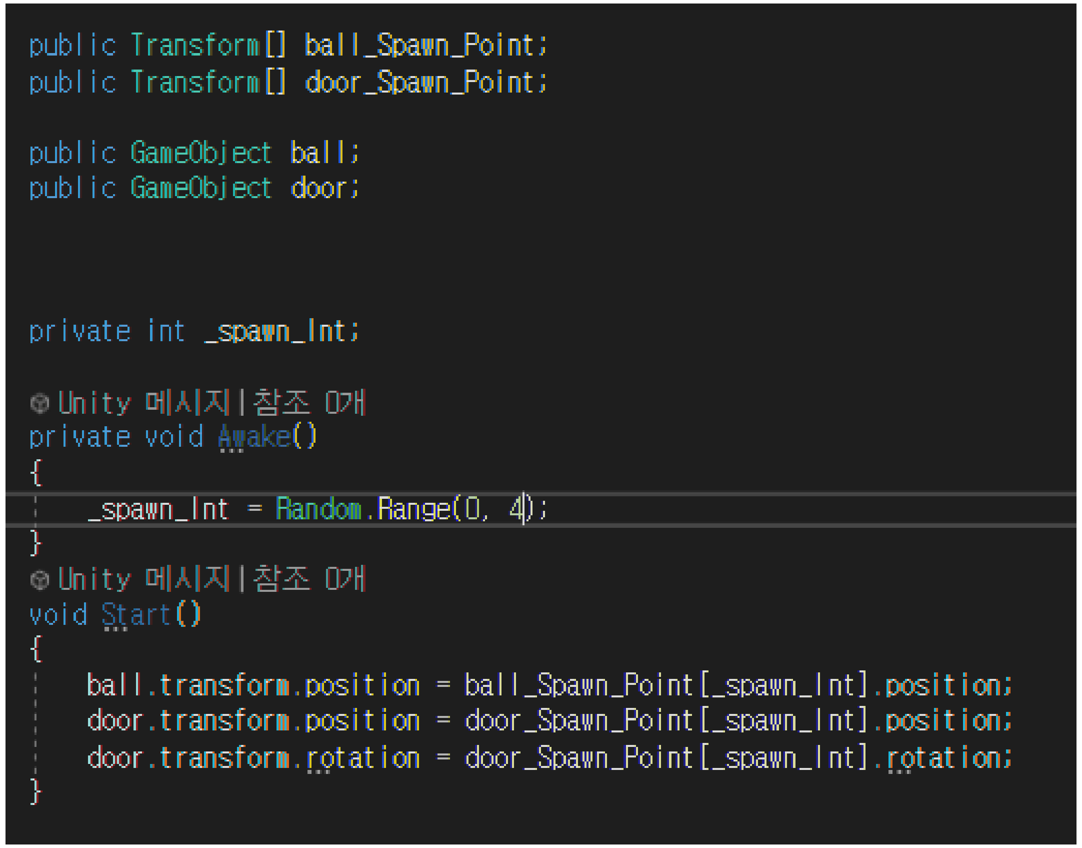
스폰 포인트



Door Spawn (Script)

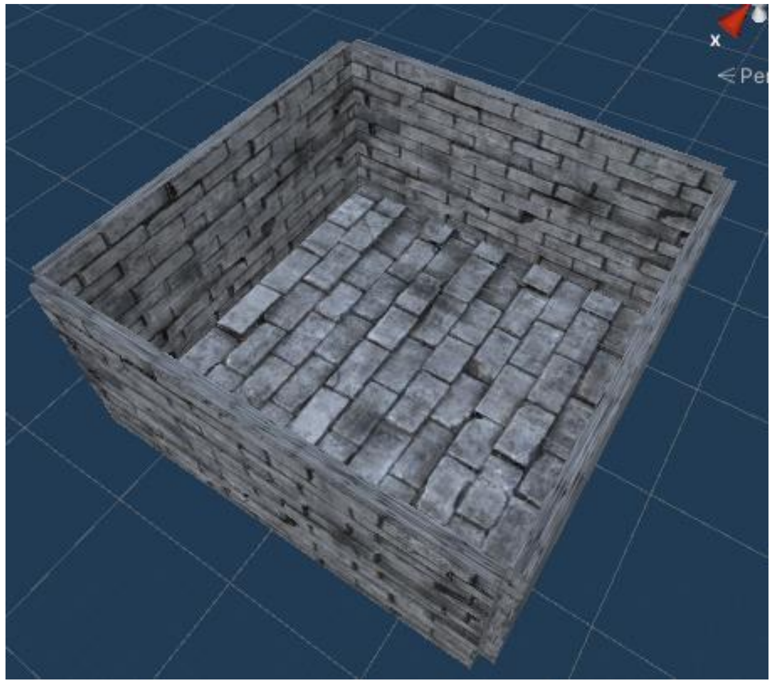
Script	DoorSpawn
Ball_Spawn_Point	4
Element 0	1_Point (Transform)
Element 1	5_Point (Transform)
Element 2	7_Point (Transform)
Element 3	11_Point (Transform)
Door_Spawn_Point	4
Ball	Ball
Door	Door

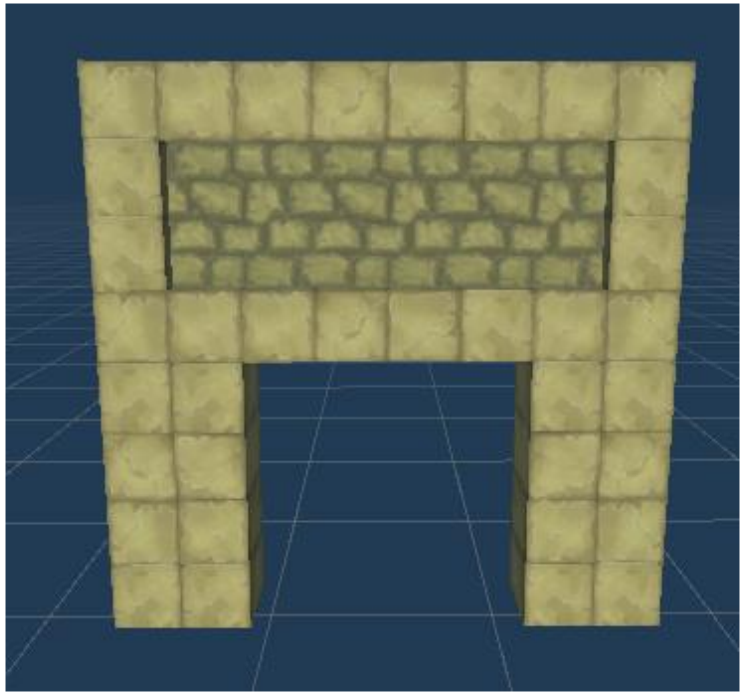
Add Component



```
public Transform[] ball_Spawn_Point;
public Transform[] door_Spawn_Point;

public GameObject ball;
public GameObject door;



private int _spawn_Int;

Unity 메시지|참조 0개
private void Awake()
{
    _spawn_Int = Random.Range(0, 4);
}
Unity 메시지|참조 0개
void Start()
{
    ball.transform.position = ball_Spawn_Point[_spawn_Int].position;
    door.transform.position = door_Spawn_Point[_spawn_Int].position;
    door.transform.rotation = door_Spawn_Point[_spawn_Int].rotation;
}
```

그래픽적 요소

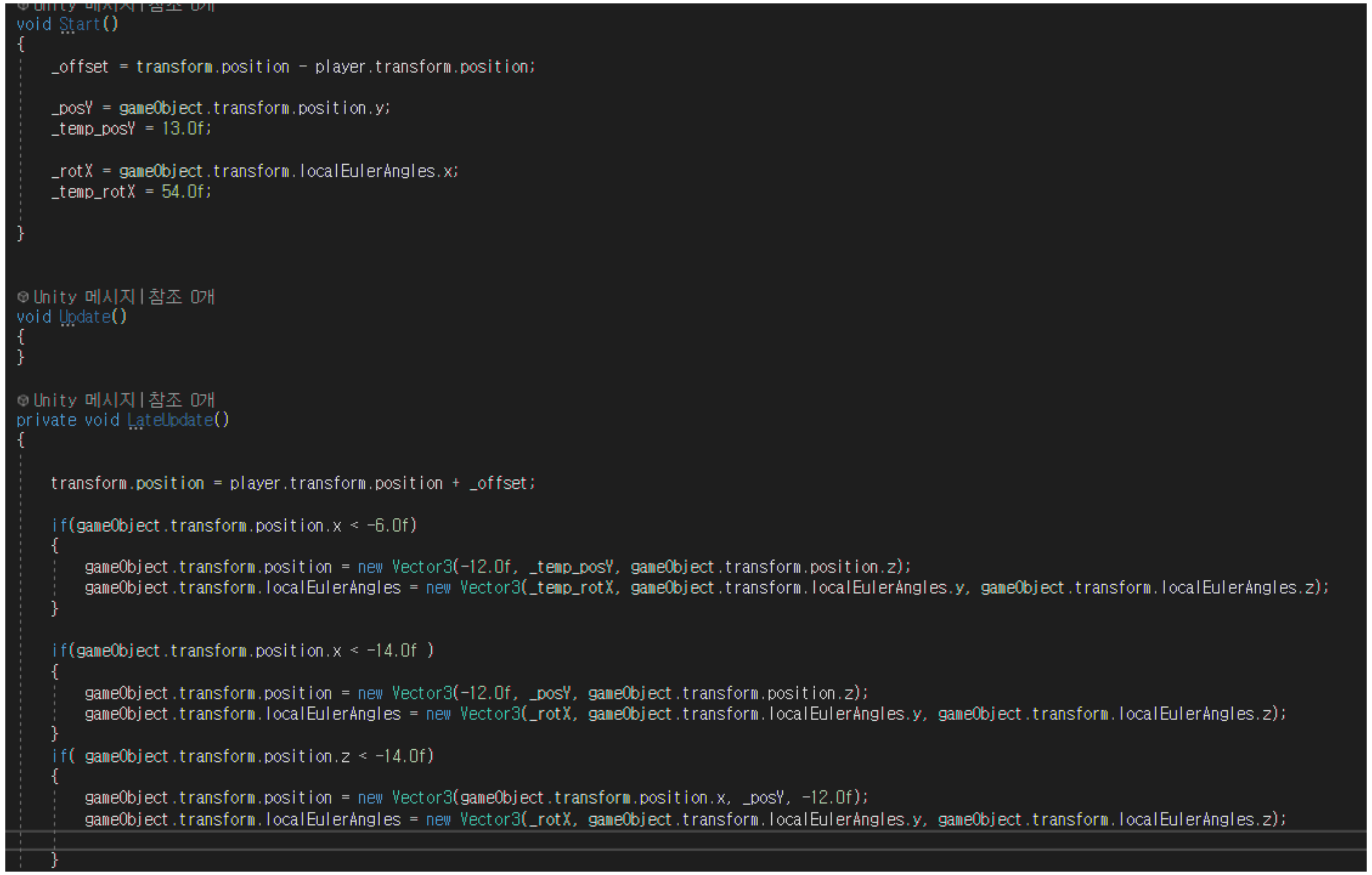




빙고

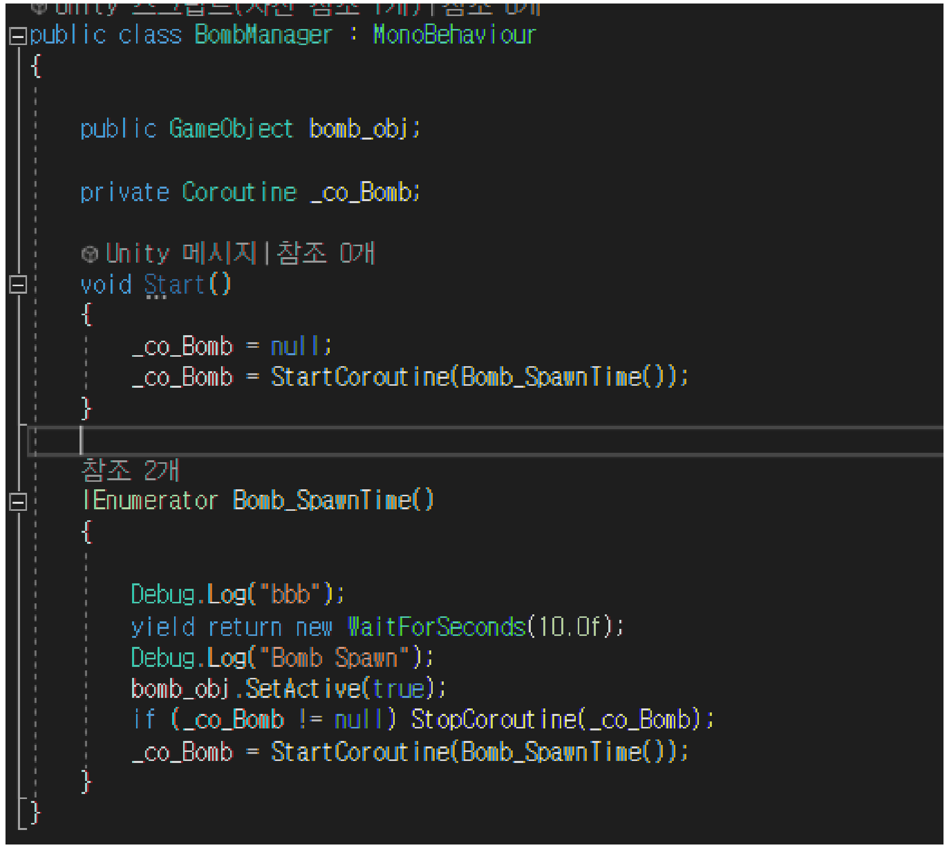
- 플레이어가 10초동안 나오는 폭탄을 적절하게 배치해서 랜덤으로 나오는 망치를 피해 최대한 많이 버티는 것을 목표로 플레이하는 시뮬레이션 형태로 만들어졌습니다.
- 서술할 코드로는 플레이어, 폭탄 알고리즘, 바닥 알고리즘, 해머, 그래픽적 요소입니다.

플레이어

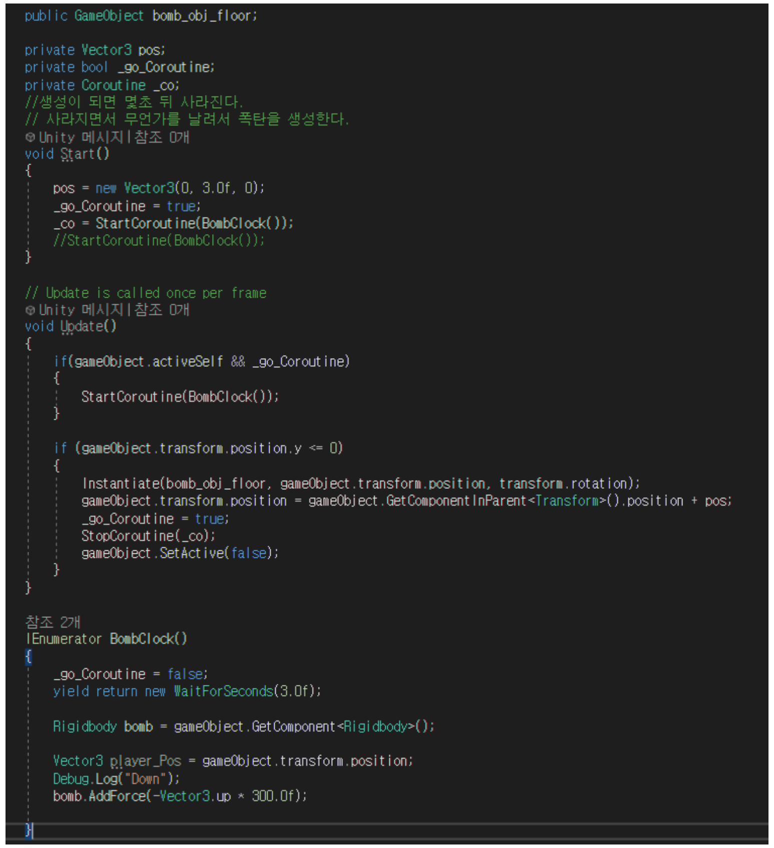


```
void Start()
{
    _offset = transform.position - player.transform.position;

    _posY = gameObject.transform.position.y;
    _temp_posY = 13.0f;

    _rotX = gameObject.transform.localEulerAngles.x;
    _temp_rotX = 54.0f;

}


Unity 메시지 | 참조 0개
void Update()
{
}

Unity 메시지 | 참조 0개
private void LateUpdate()
{

    transform.position = player.transform.position + _offset;

    if(gameObject.transform.position.x < -6.0f)
    {
        gameObject.transform.position = new Vector3(-12.0f, _temp_posY, gameObject.transform.position.z);
        gameObject.transform.localEulerAngles = new Vector3(_temp_rotX, gameObject.transform.localEulerAngles.y, gameObject.transform.localEulerAngles.z);
    }

    if(gameObject.transform.position.x < -14.0f )
    {
        gameObject.transform.position = new Vector3(-12.0f, _posY, gameObject.transform.position.z);
        gameObject.transform.localEulerAngles = new Vector3(_rotX, gameObject.transform.localEulerAngles.y, gameObject.transform.localEulerAngles.z);
    }
    if( gameObject.transform.position.z < -14.0f)
    {
        gameObject.transform.position = new Vector3(gameObject.transform.position.x, _posY, -12.0f);
        gameObject.transform.localEulerAngles = new Vector3(_rotX, gameObject.transform.localEulerAngles.y, gameObject.transform.localEulerAngles.z);

    }
```

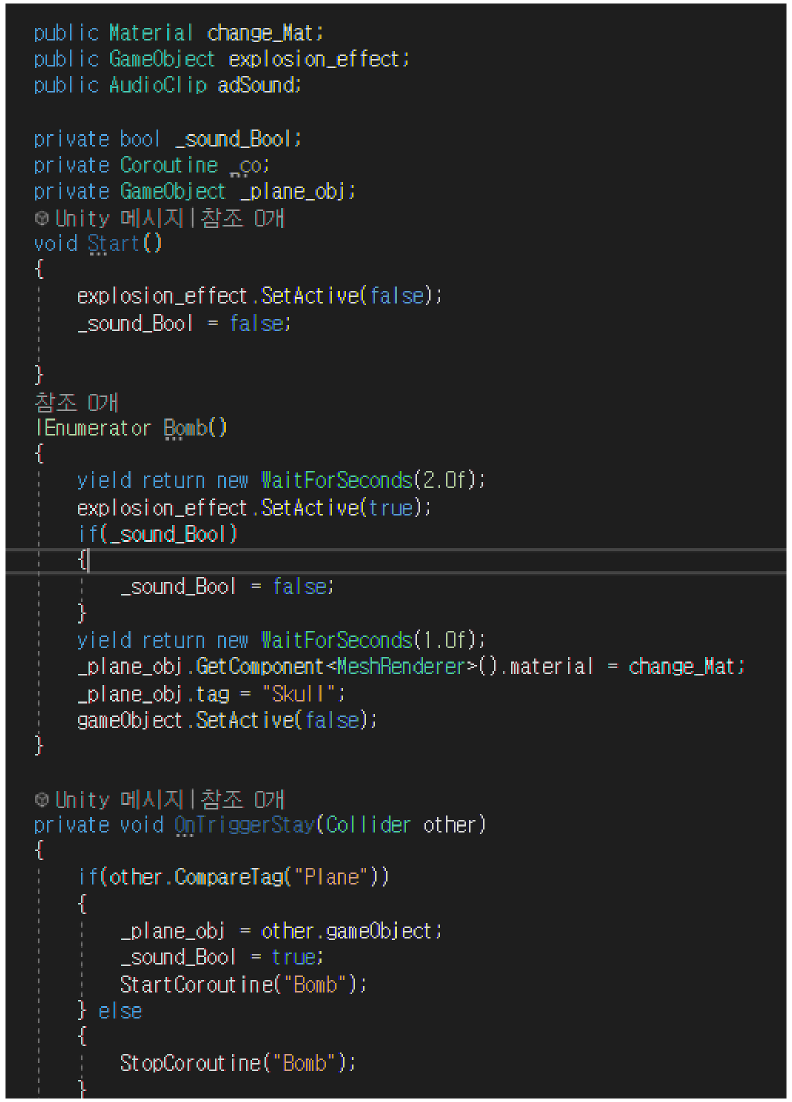
폭탄



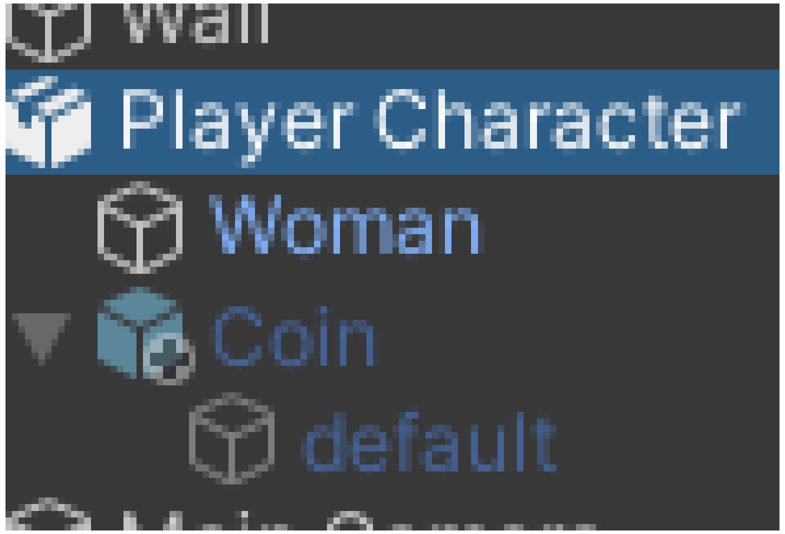
```
public class BombManager : MonoBehaviour
{

    public GameObject bomb_obj;

    private Coroutine _co_Bomb;

    Unity 메시지|참조 0개
    void Start()
    {
        _co_Bomb = null;
        _co_Bomb = StartCoroutine(Bomb_SpawnTime());
    }

    참조 2개
    IEnumerator Bomb_SpawnTime()
    {

        Debug.Log("bbb");
        yield return new WaitForSeconds(10.0f);
        Debug.Log("Bomb Spawn");
        bomb_obj.SetActive(true);
        if (_co_Bomb != null) StopCoroutine(_co_Bomb);
        _co_Bomb = StartCoroutine(Bomb_SpawnTime());
    }
}
```



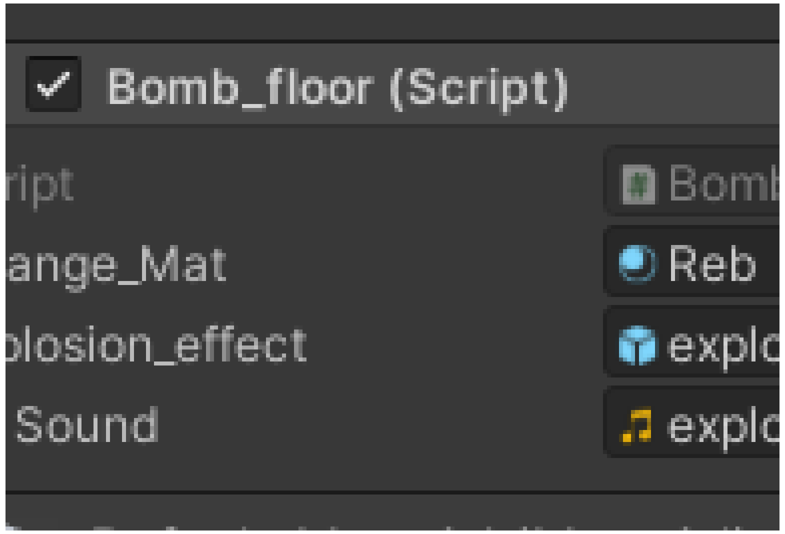
```
public GameObject bomb_obj_floor;

private Vector3 pos;
private bool _go_Coroutine;
private Coroutine _co;
//생성이 되면 몇초 뒤 사라진다.
// 사라지면서 무언가를 날려서 폭탄을 생성한다.
Unity 메시지|참조 0개
void Start()
{
    pos = new Vector3(0, 3.0f, 0);
    _go_Coroutine = true;
    _co = StartCoroutine(BombClock());
    //StartCoroutine(BombClock());
}

// Update is called once per frame
Unity 메시지|참조 0개
void Update()
{
    if(gameObject.activeSelf && _go_Coroutine)
    {
        StartCoroutine(BombClock());
    }

    if (gameObject.transform.position.y <= 0)
    {
        Instantiate(bomb_obj_floor, gameObject.transform.position, transform.rotation);
        gameObject.transform.position = gameObject.GetComponentInParent<Transform>().position + pos;
        _go_Coroutine = true;
        StopCoroutine(_co);
        gameObject.SetActive(false);
    }
}

참조 2개
IEnumerator BombClock()
{
    _go_Coroutine = false;
    yield return new WaitForSeconds(3.0f);

    Rigidbody bomb = gameObject.GetComponent<Rigidbody>();

    Vector3 player_Pos = gameObject.transform.position;
    Debug.Log("Down");
    bomb.AddForce(-Vector3.up * 300.0f);

}
```



```
public Material change_Mat;
public GameObject explosion_effect;
public AudioClip adSound;

private bool _sound_Bool;
private Coroutine _co;
private GameObject _plane_obj;
⊕Unity 메시지|참조 0개
void Start()
{
    explosion_effect.SetActive(false);
    _sound_Bool = false;

}
참조 0개
IEnumerator Bomb()
{
    yield return new WaitForSeconds(2.0f);
    explosion_effect.SetActive(true);
    if(_sound_Bool)
    {
        _sound_Bool = false;
    }
    yield return new WaitForSeconds(1.0f);
    _plane_obj.GetComponent<MeshRenderer>().material = change_Mat;
    _plane_obj.tag = "Skull";
    gameObject.SetActive(false);
}

⊕Unity 메시지|참조 0개
private void OnTriggerStay(Collider other)
{
    if(other.CompareTag("Plane"))
    {
        _plane_obj = other.gameObject;
        _sound_Bool = true;
        StartCoroutine("Bomb");
    } else
    {
        StopCoroutine("Bomb");
    }
```

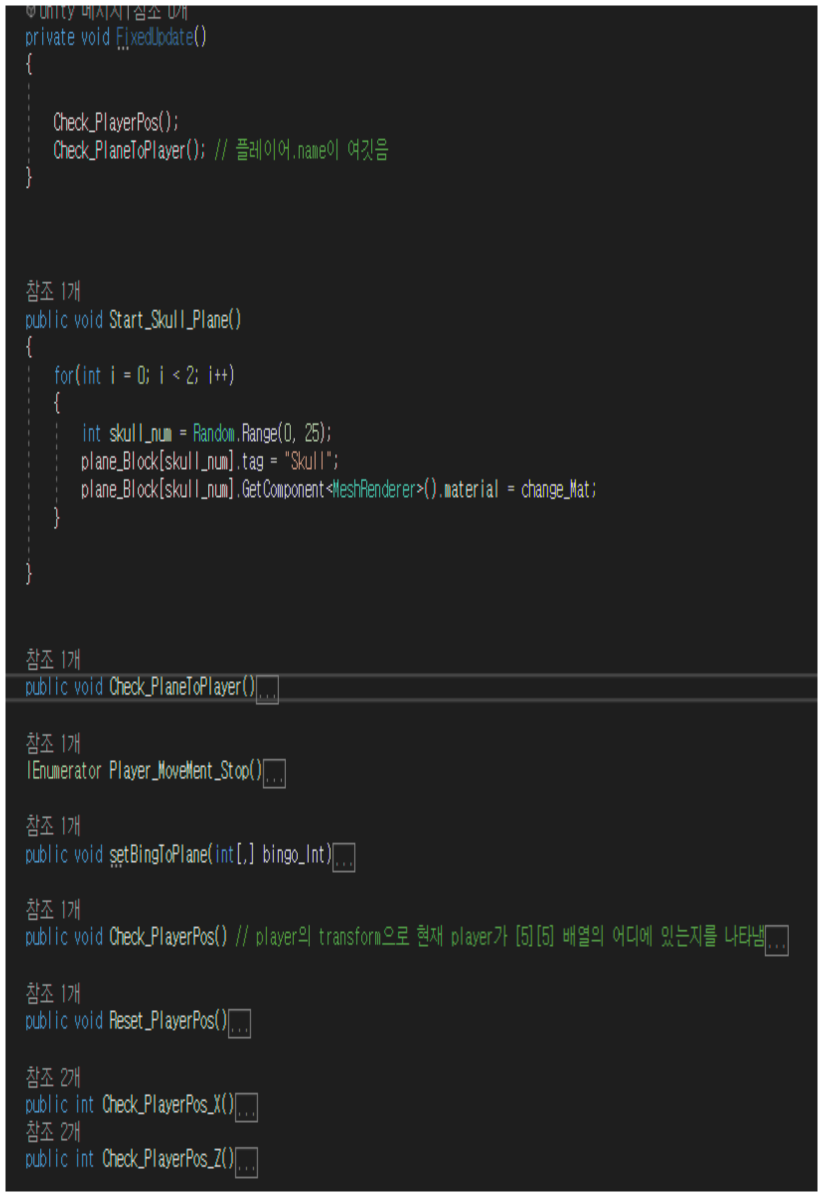


Wall
Player Character
Woman
Coin
default

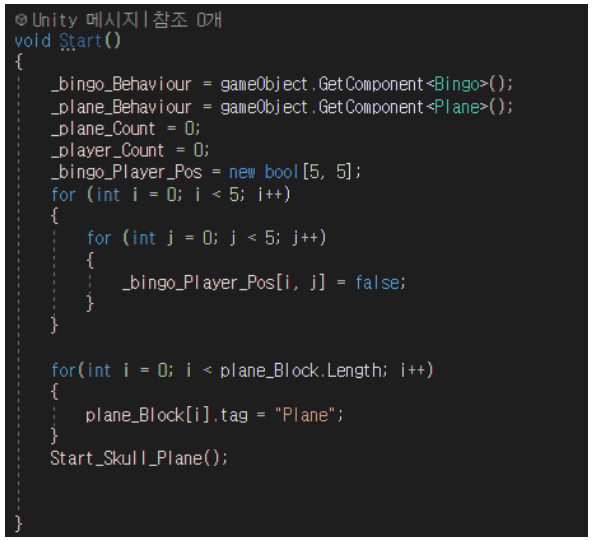


✓ Bomb_floor (Script)

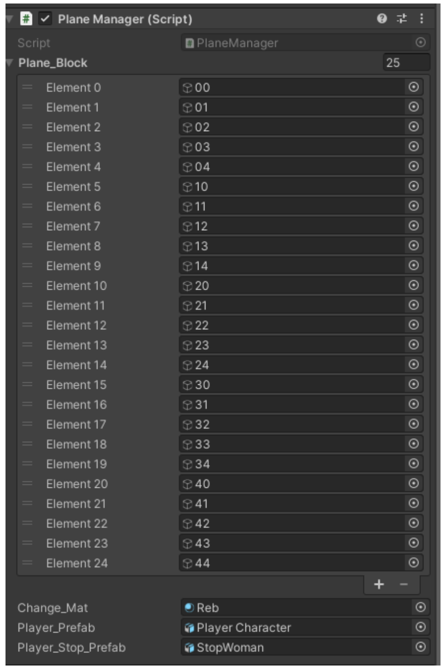
ript	Bomb
ange_Mat	Reb
plosion_effect	explo
Sound	explo



```
ⓤ Unity 메시지|참조 0개
private void FixedUpdate()
{


    Check_PlayerPos();
    Check_PlaneToPlayer(); // 플레이어.name이 여깃음
}




참조 1개
public void Start_Skull_Plane()
{
    for(int i = 0; i < 2; i++)
    {
        int skull_num = Random.Range(0, 25);
        plane_Block[skull_num].tag = "Skull";
        plane_Block[skull_num].GetComponent<MeshRenderer>().material = change_Mat;
    }

}


참조 1개
public void Check_PlaneToPlayer()[...]

참조 1개
IEnumerator Player_MoveMent_Stop()[...]

참조 1개
public void setBingToPlane(int[,] bingo_Int)[...]

참조 1개
public void Check_PlayerPos() // player의 transform으로 현재 player가 [5][5] 배열의 어디에 있는지를 나타냄[...]

참조 1개
public void Reset_PlayerPos()[...]

참조 2개
public int Check_PlayerPos_X()[...]
참조 2개
public int Check_PlayerPos_Z()[...]
```



```
ⓤ Unity 메시지|참조 0개
void Start()
{
    _bingo_Behaviour = gameObject.GetComponent<Bingo>();
    _plane_Behaviour = gameObject.GetComponent<Plane>();
    _plane_Count = 0;
    _player_Count = 0;
    _bingo_Player_Pos = new bool[5, 5];
    for (int i = 0; i < 5; i++)
    {
        for (int j = 0; j < 5; j++)
        {
            _bingo_Player_Pos[i, j] = false;
        }
    }

    for(int i = 0; i < plane_Block.Length; i++)
    {
        plane_Block[i].tag = "Plane";
    }
    Start_Skull_Plane();

}
```

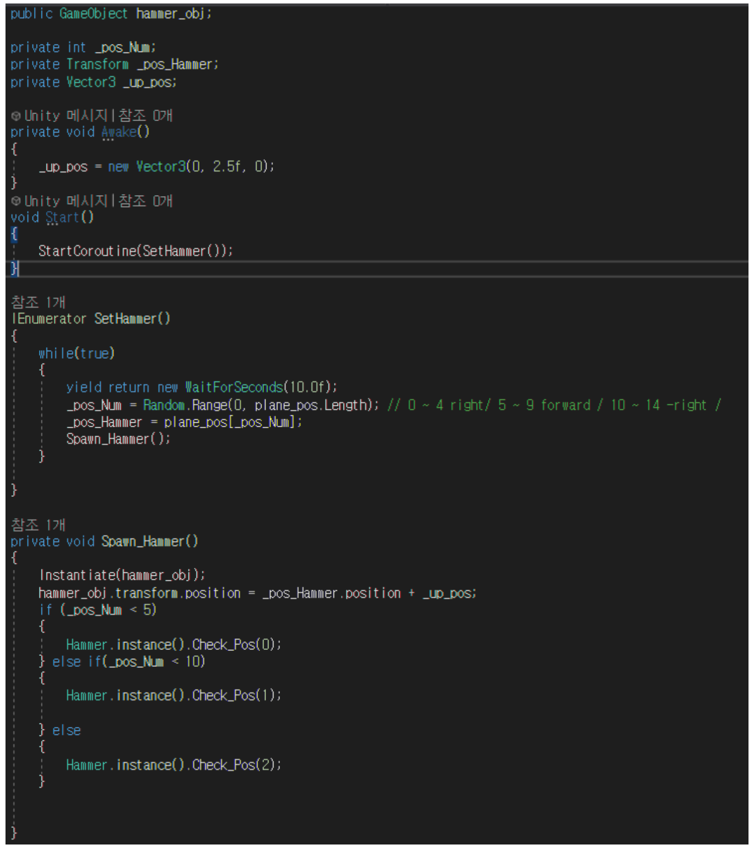


Plane Manager (Script)

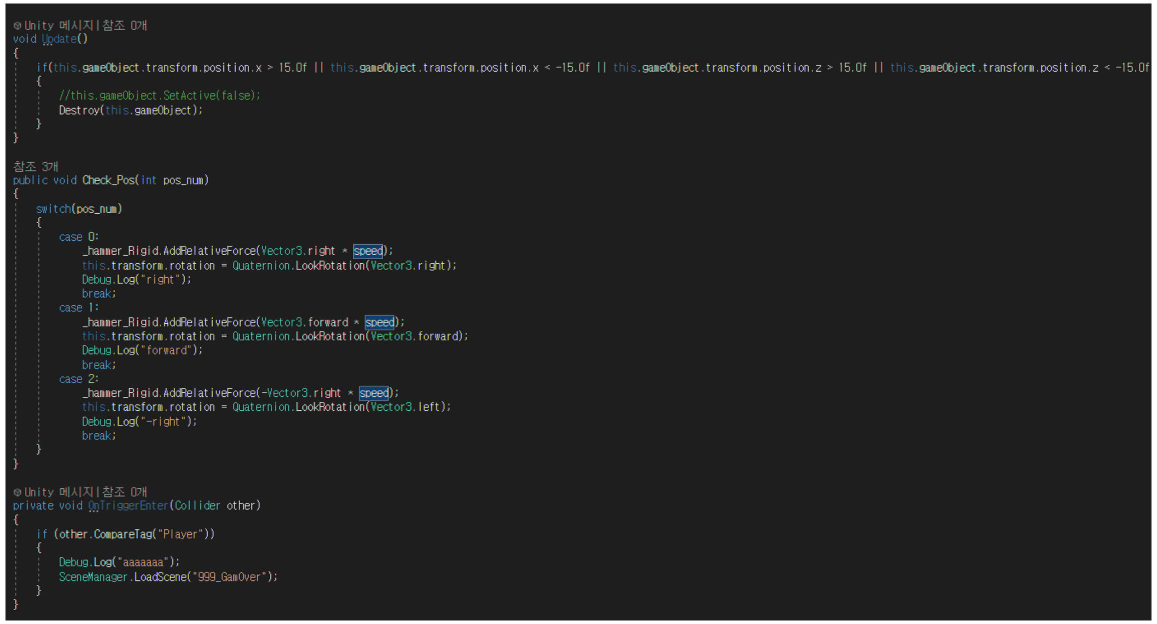
Script	PlaneManager
Plane_Block	25
Element 0	00
Element 1	01
Element 2	02
Element 3	03
Element 4	04
Element 5	10
Element 6	11
Element 7	12
Element 8	13
Element 9	14
Element 10	20
Element 11	21
Element 12	22
Element 13	23
Element 14	24
Element 15	30
Element 16	31
Element 17	32
Element 18	33
Element 19	34
Element 20	40
Element 21	41
Element 22	42
Element 23	43
Element 24	44
Change_Mat	Reb
Player_Prefab	Player Character
Player_Stop_Prefab	StopWoman

바닥

해머

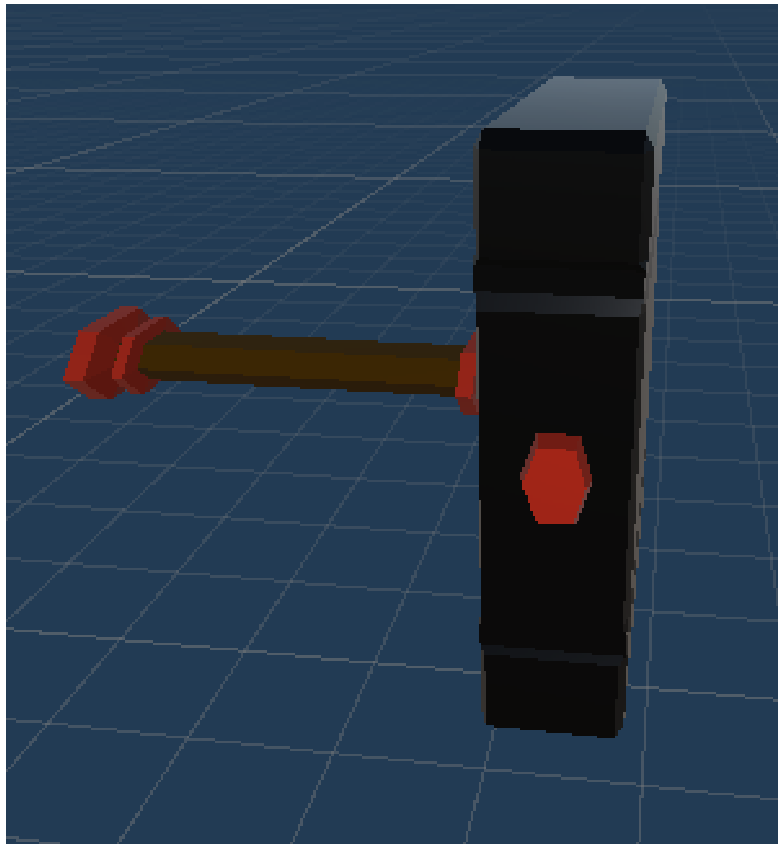


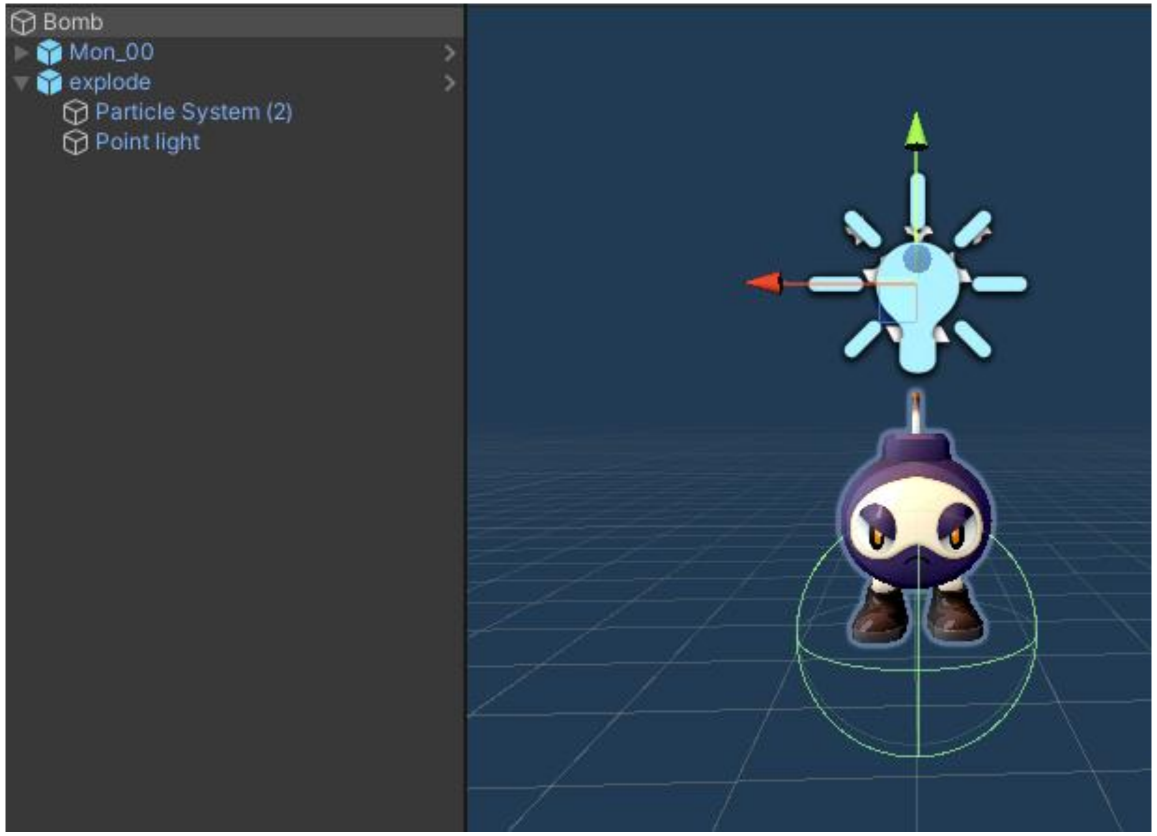
```
public GameObject hammer_obj;

private int _pos_Num;
private Transform _pos_Hammer;
private Vector3 _up_pos;

Unity 메시지|참조 0개
private void Awake()
{
    _up_pos = new Vector3(0, 2.5f, 0);
}
Unity 메시지|참조 0개
void Start()
{
    StartCoroutine(SetHammer());
}

참조 1개
IEnumerator SetHammer()
{
    while(true)
    {
        yield return new WaitForSeconds(10.0f);
        _pos_Num = Random.Range(0, plane_pos.Length); // 0 ~ 4 right/ 5 ~ 9 forward / 10 ~ 14 -right /
        _pos_Hammer = plane_pos[_pos_Num];
        Spawn_Hammer();
    }

}

참조 1개
private void Spawn_Hammer()
{
    Instantiate(hammer_obj);
    hammer_obj.transform.position = _pos_Hammer.position + _up_pos;
    if (_pos_Num < 5)
    {
        Hammer.instance().Check_Pos(0);
    } else if(_pos_Num < 10)
    {
        Hammer.instance().Check_Pos(1);

    } else
    {
        Hammer.instance().Check_Pos(2);
    }


}
```



```
Unity 메시지|참조 0개
void Update()
{
    if(this.gameObject.transform.position.x > 15.0f || this.gameObject.transform.position.x < -15.0f || this.gameObject.transform.position.z > 15.0f || this.gameObject.transform.position.z < -15.0f
    {
        //this.gameObject.SetActive(false);
        Destroy(this.gameObject);
    }
}

참조 3개
public void Check_Pos(int pos_num)
{
    switch(pos_num)
    {
        case 0:
            _hammer_Rigid.AddRelativeForce(Vector3.right * speed);
            this.transform.rotation = Quaternion.LookRotation(Vector3.right);
            Debug.Log("right");
            break;
        case 1:
            _hammer_Rigid.AddRelativeForce(Vector3.forward * speed);
            this.transform.rotation = Quaternion.LookRotation(Vector3.forward);
            Debug.Log("forward");
            break;
        case 2:
            _hammer_Rigid.AddRelativeForce(-Vector3.right * speed);
            this.transform.rotation = Quaternion.LookRotation(Vector3.left);
            Debug.Log("-right");
            break;
    }
}

Unity 메시지|참조 0개
private void OnTriggerEnter(Collider other)
{
    if (other.CompareTag("Player"))
    {
        Debug.Log("aaaaaaa");
        SceneManager.LoadScene("999_GamOver");
    }
}
```

그래픽적 요소




Bomb
Mon_00
explode
Particle System (2)
Point light