



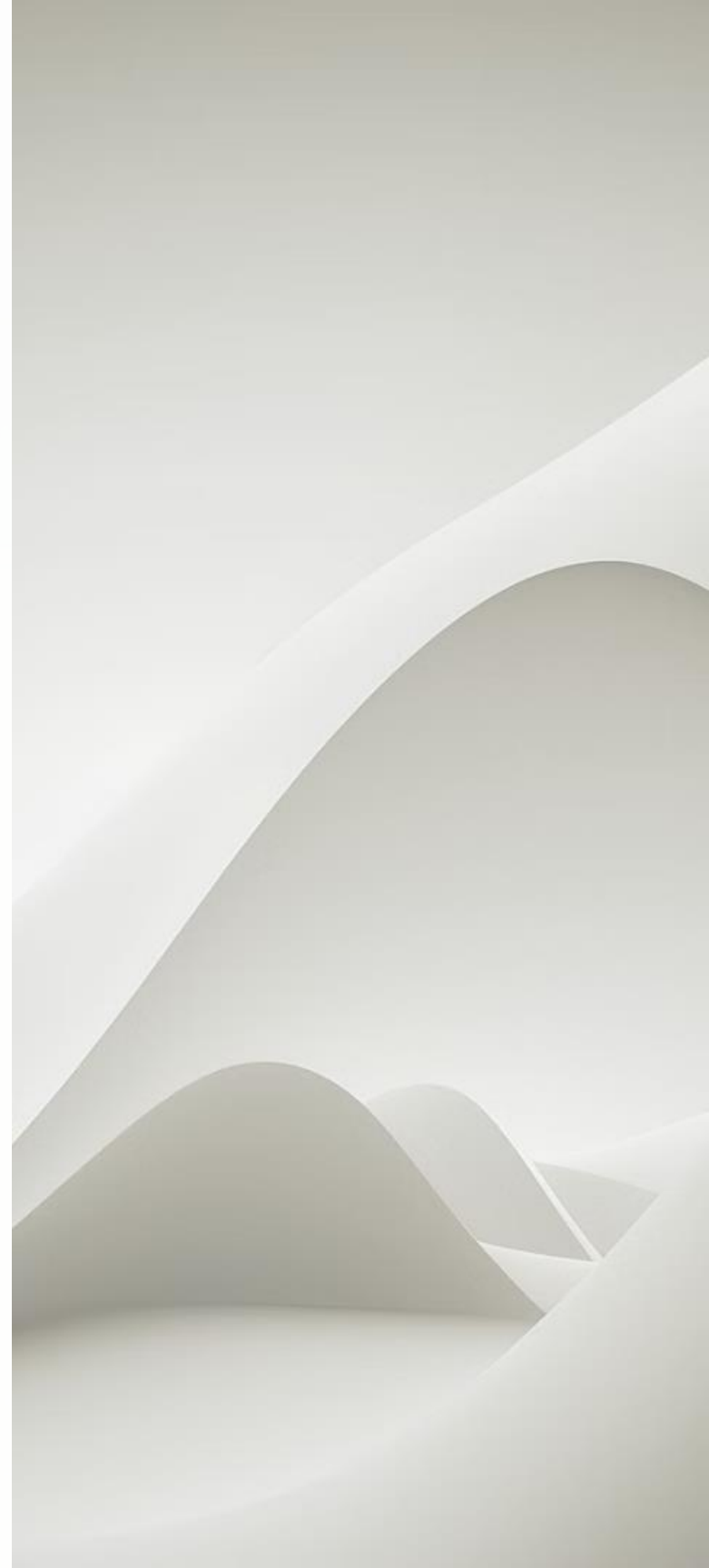
게임 프로그래밍 - 유니티

<https://ftpserver.ksecole.kr/files/Unity/1121/>

2020975005 강현준

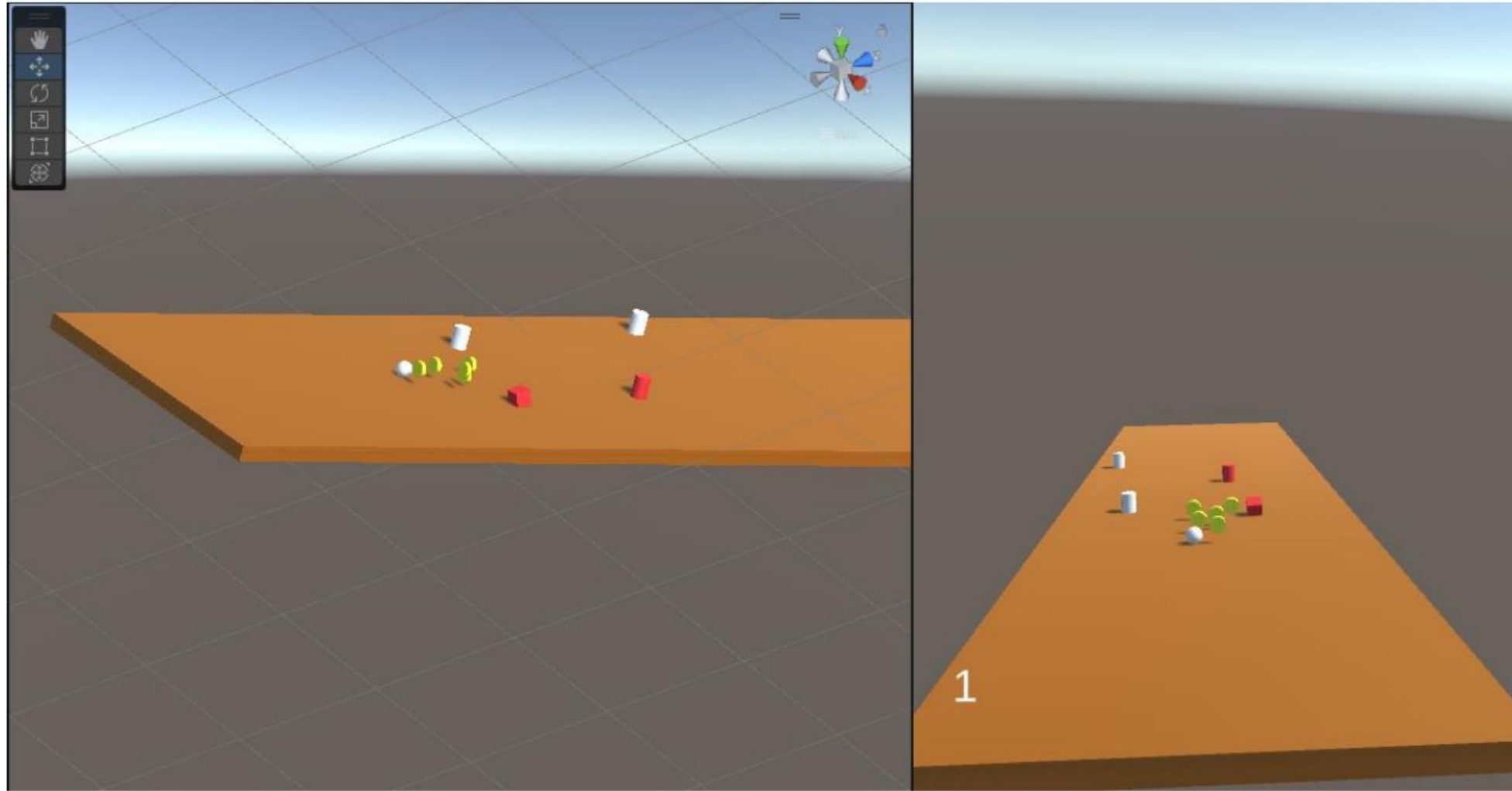
목차

- 1 원본 게임
- 2 업그레이드 요소
- 3 점수 및 레퍼런스



원본 게임

FTPSERVER – Unity – 1121 – 게임프로그래밍03.pdf



땅을 마우스로 기울이고, 공을 움직여서 코인 먹기



업그레이드 요소 – 시작화면

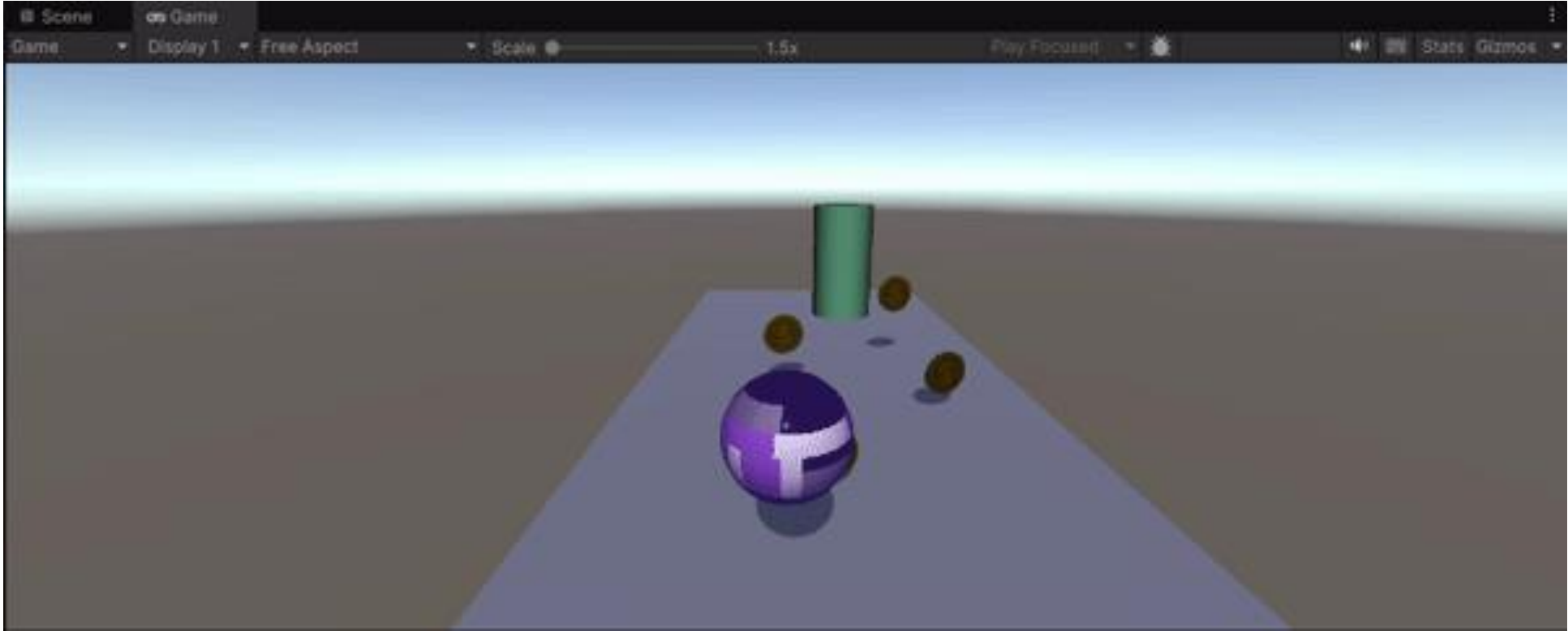


```
using UnityEngine;
using UnityEngine.SceneManagement;

참조 0개
public class MenuManager : MonoBehaviour
{
    참조 0개
    public void StartGame()
    {
        SceneManager.LoadScene("Stage0");
    }
}
```



업그레이드 요소 – 공 조작하기



```
public class MyBall : MonoBehaviour
{
    참조 5개
    Rigidbody rigid;
    참조 0개
    void Start()
    {
        rigid = GetComponent<Rigidbody>();
    }

    참조 0개
    void FixedUpdate()
    {
        if (Input.GetButtonDown("Jump"))
        {
            rigid.AddForce(Vector3.up * 50, ForceMode.Impulse);
        }
    }
}
```

```
        Vector3 vec = new Vector3(
            Input.GetAxisRaw("Horizontal"),
            0,
            Input.GetAxisRaw("Vertical"));

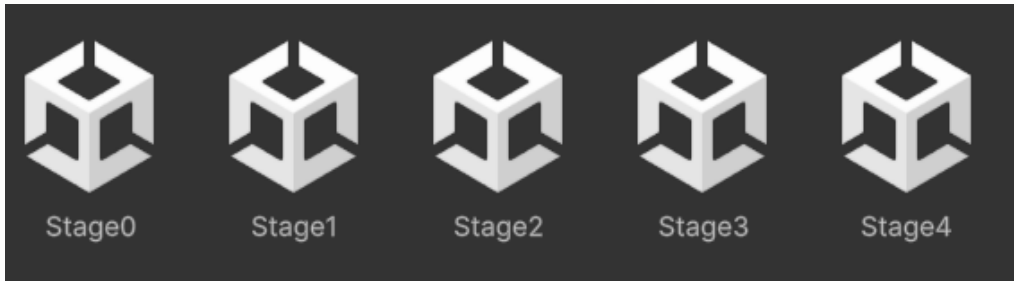
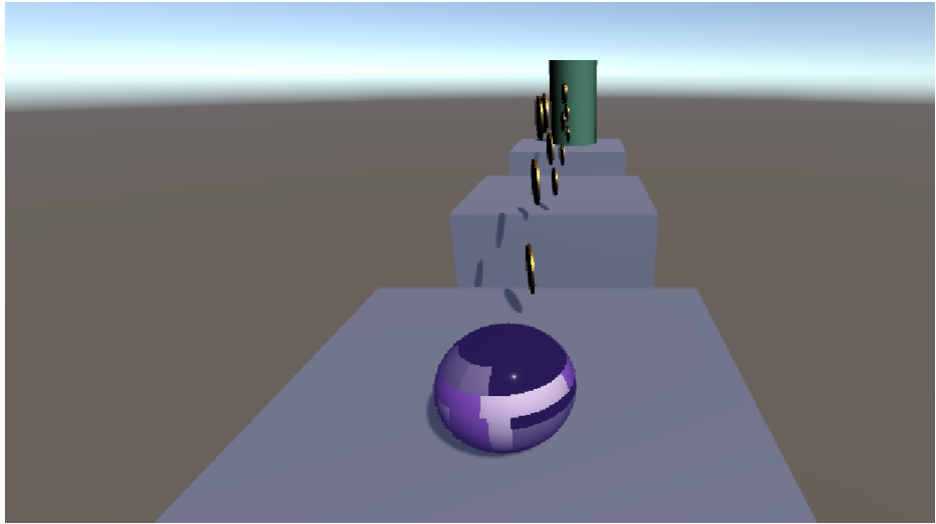
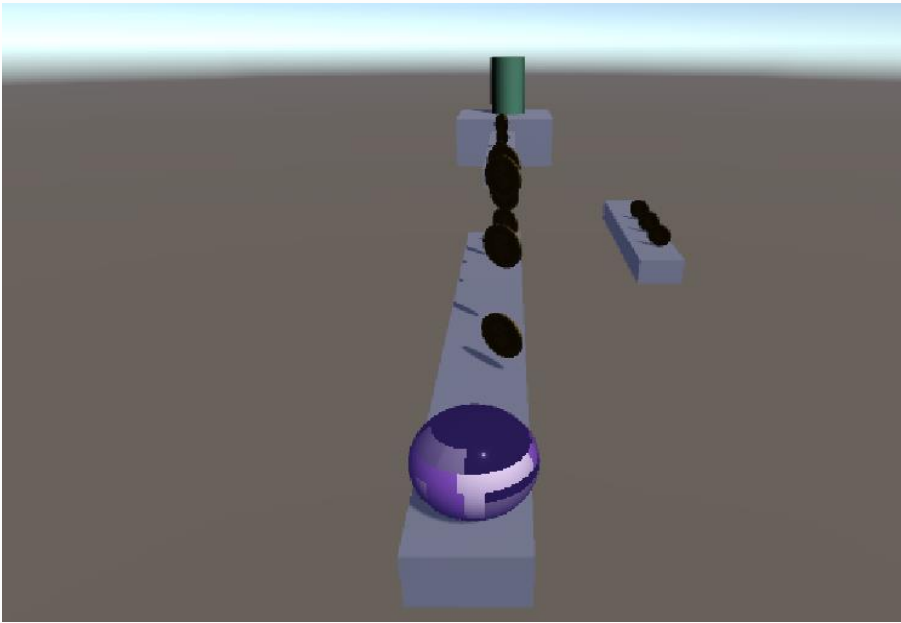
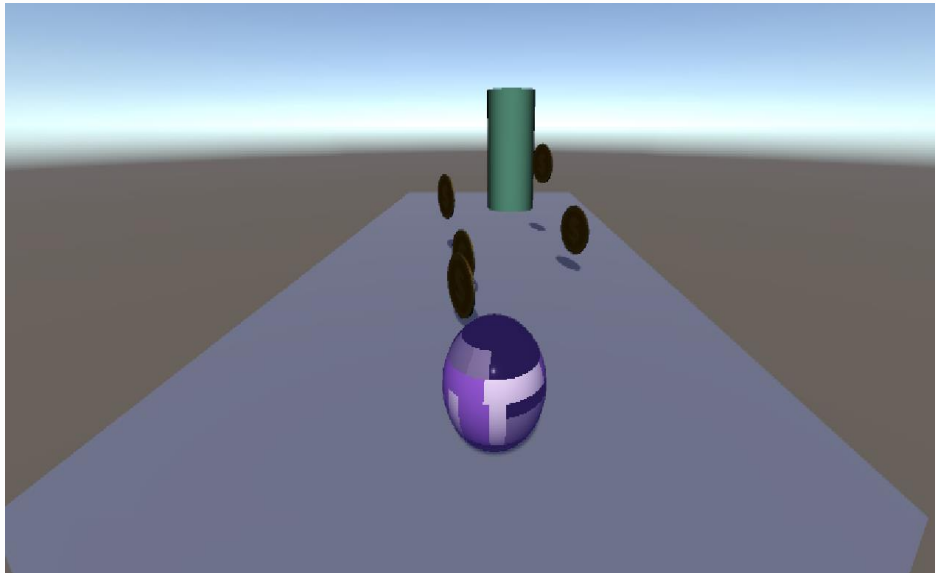
        rigid.AddForce(vec, ForceMode.Impulse);
    }

    참조 0개
    private void OnTriggerStay(Collider other)
    {
        if (other.name == "Cube")
        {
            rigid.AddForce(Vector3.up * 2, ForceMode.Impulse);
        }
    }

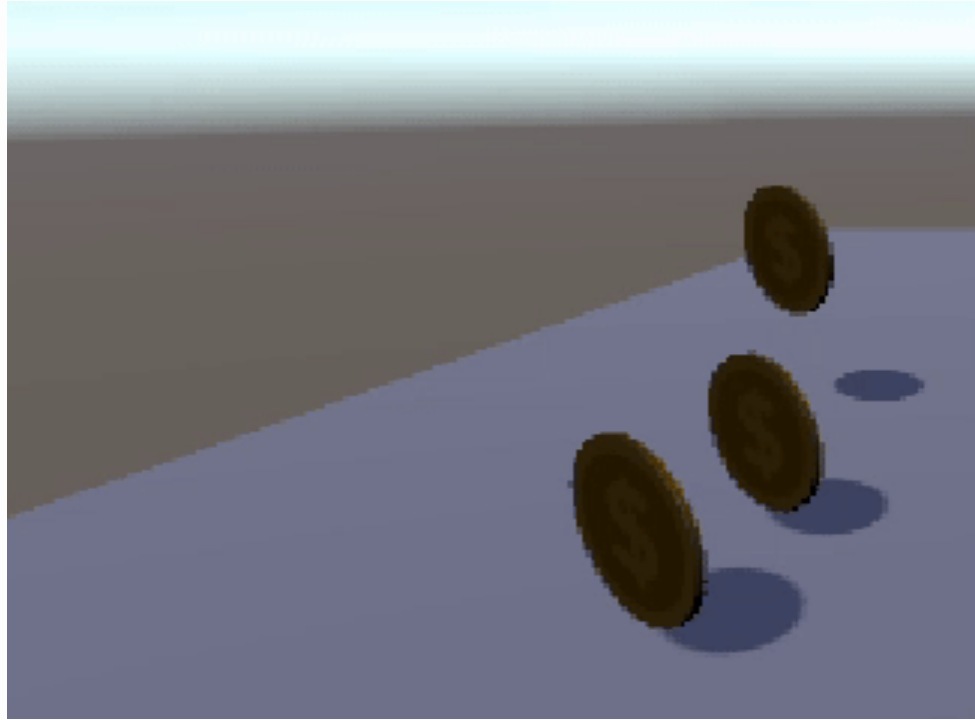
    참조 0개
    public void Jump()
    {
        rigid.AddForce(Vector3.up * 20, ForceMode.Impulse);
    }
}
```



업그레이드 요소 – 다양한 스테이지



업그레이드 요소 – 회전하는 코인



```
Assets > 2Scripts > Coin.cs > Coin
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
   참조 0개
5  public class Coin : MonoBehaviour
6  {
   참조 1개
7      public float rotateSpeed;
8
   참조 0개
9      void Update()
10     {
11         transform.Rotate(Vector3.up * rotateSpeed * Time.deltaTime, Space.World);
12     }
13
14
15 }
```



업그레이드 요소 – 사운드 추가

```
참조 0개
void Update()
{
    if (Input.GetButtonDown("Jump") && !isJump)
    {
        isJump = true;
        // 점프 사운드 재생
        if (jumpClip != null)
            audio1.PlayOneShot(jumpClip);

        rigid.AddForce(Vector3.up * jumpPower, ForceMode.Impulse);
    }
}
```

```
if (other.CompareTag("Coin"))
{
    score++;
    audio1.Play(); // 코인 획득 사운드
    other.gameObject.SetActive(false);
}
```

▼ # ✓ Player Ball (Script) ⓘ ⚙ ⋮

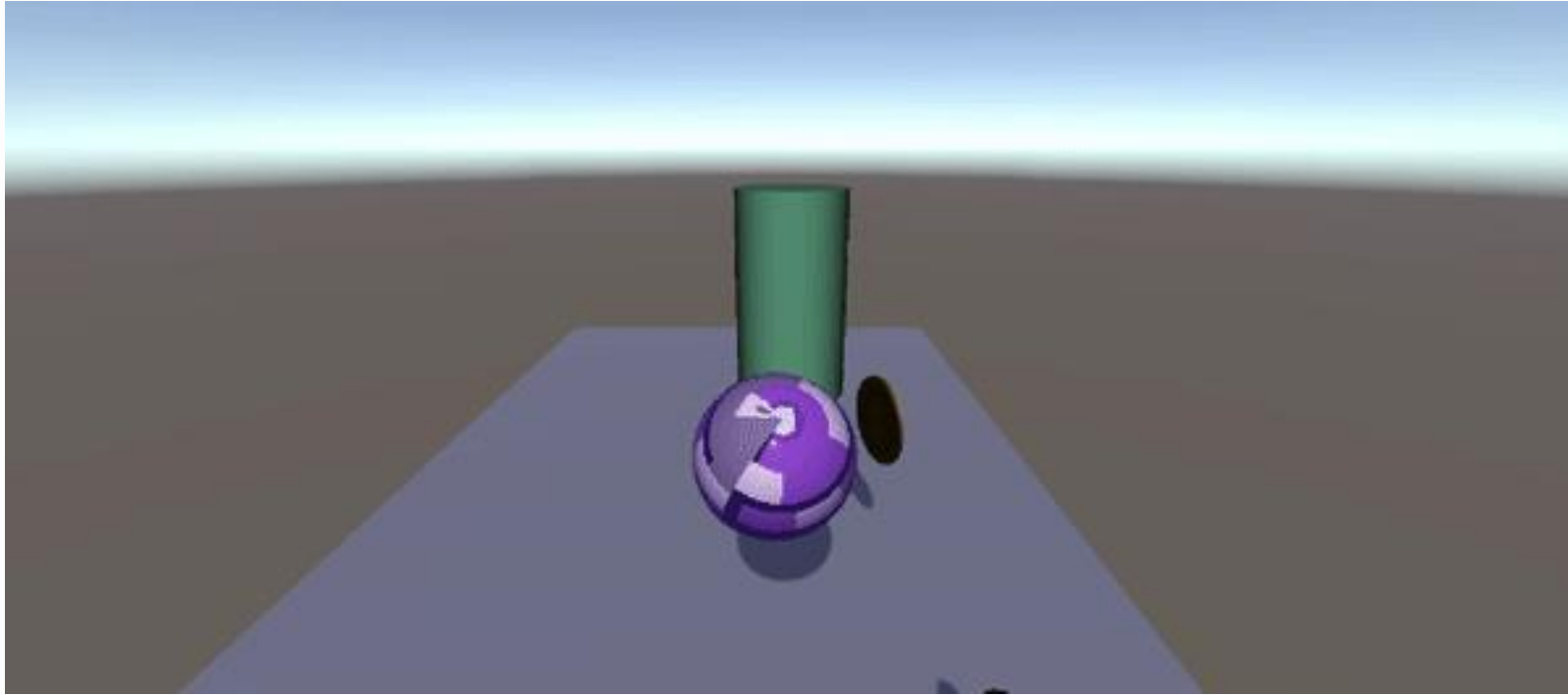
Script	# PlayerBall ⓘ
Jump Power	30
Move Power	0.5
Score	0
Manager	# GameManager (Game Manager Log ⓘ)
Jump Clip	🎵 DM-CGS-41 ⓘ

▼ 🗣️ ✓ Audio Source ⓘ ⚙ ⋮

Audio Resource	🎵 DM-CGS-28 ⓘ
Output	None (Audio Mixer Group) ⓘ
Mute	<input type="checkbox"/>
Bypass Effects	<input type="checkbox"/>
Bypass Listener Effects	<input type="checkbox"/>
Bypass Reverb Zones	<input type="checkbox"/>
Play On Awake	<input type="checkbox"/>
Loop	<input type="checkbox"/>



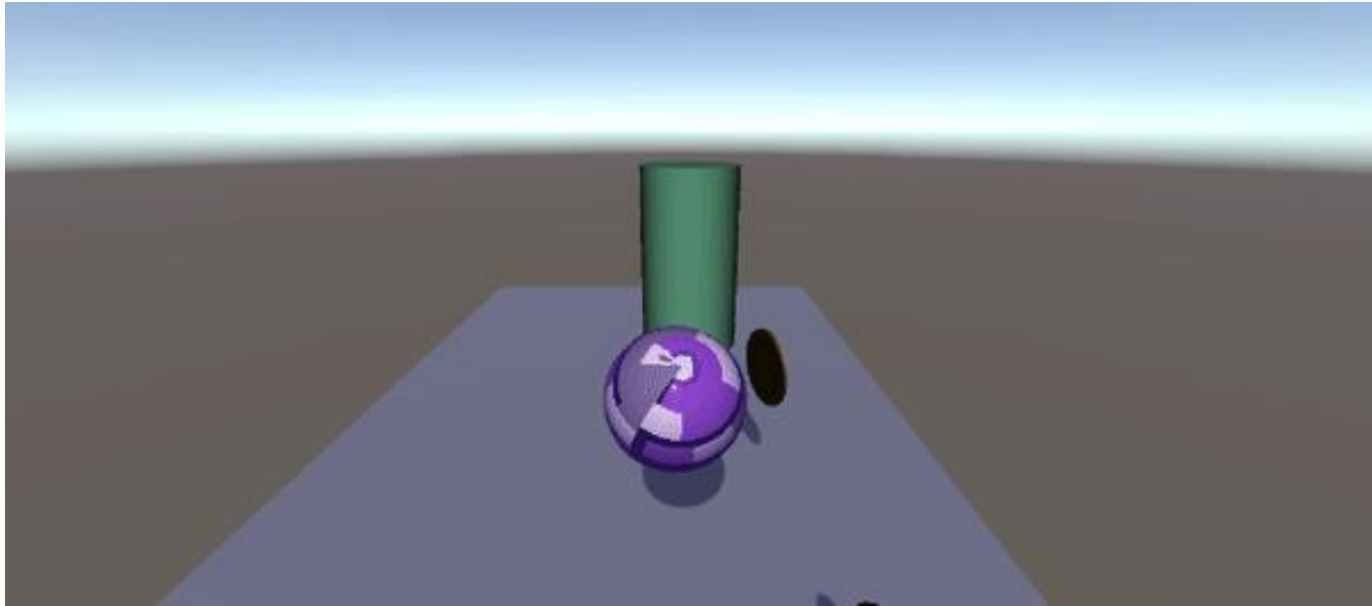
업그레이드 요소 – 골인 포인트



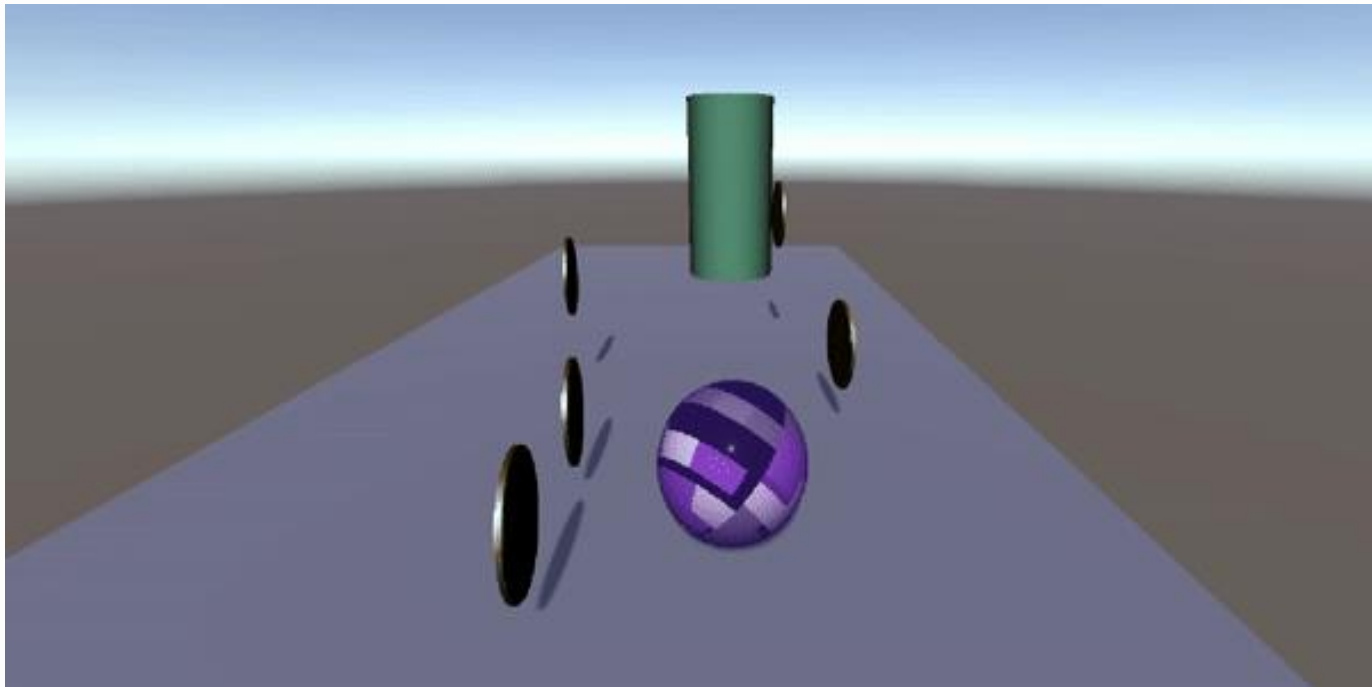
```
else if (other.CompareTag("Finish"))
{
    if (score == manager.totalItemCount)
    {
        if (manager.stage == 4)
        {
            // 마지막 스테이지 클리어 시 메뉴로
            SceneManager.LoadScene("Menu");
        }
        else
        {
            // 다음 스테이지로 이동
            int nextStageIndex = manager.stage + 1;
            string nextStageName = "Stage" + nextStageIndex.ToString();
            SceneManager.LoadScene(nextStageName);
        }
    }
    else
    {
        // 아이템 다 못모으면 현재 스테이지 재시작
        string currentStageName = "Stage" + manager.stage.ToString();
        SceneManager.LoadScene(currentStageName);
    }
}
```



업그레이드 요소 – 도전요소 추가



코인을 다 모았을 경우



코인을 다 모으지 않았을 경우



업그레이드 요소 – 움직이는 플랫폼



```
using UnityEngine;

참조 0개
public class UpDownCycle : MonoBehaviour
{
    // 이동 설정
    참조 1개
    public float amplitude = 10.0f;      // 위/아래 최대 이동 거리
    참조 2개
    public float upDuration = 4.0f;      //위로 가는 데 걸리는 시간
    참조 2개
    public float downDuration = 4.0f;    //아래로 가는 데 걸리는 시간
    참조 1개
    public float topWaitDuration = 3.0f; //위에서 대기 시간
    참조 1개
    public float bottomWaitDuration = 3.0f; //아래에서 대기 시간

    참조 4개
    private Vector3 bottomPosition;
    참조 3개
    private Vector3 topPosition;

    참조 8개
    private float timer = 0f;
}
```

```
void Start()
{
    bottomPosition = transform.position;
    topPosition = bottomPosition + new Vector3(0, amplitude, 0);
}

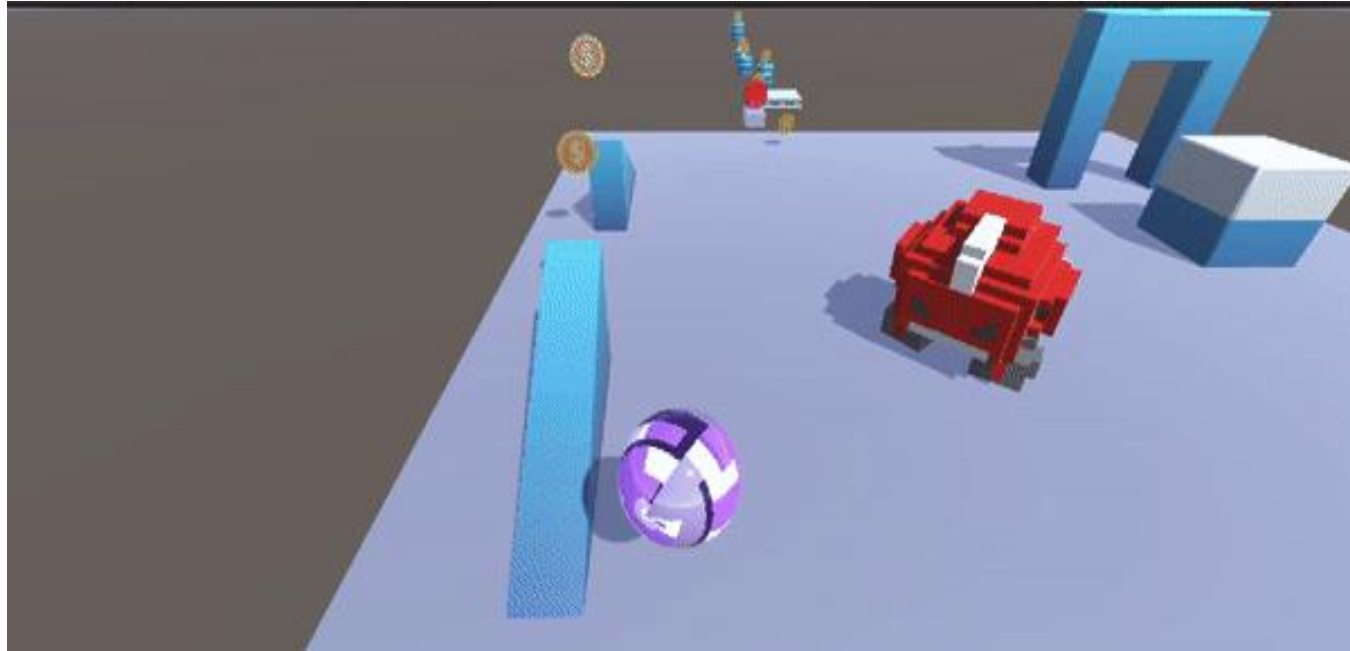
참조 0개
void Update()
{
    timer += Time.deltaTime;

    switch (currentState)
    {
        case State.GoingUp:
            // 0초일 때 bottomPosition, upDuration초일 때 topPosition
            float upT = Mathf.Clamp01(timer / upDuration);
            transform.position = Vector3.Lerp(bottomPosition, topPosition, upT);

            if (timer >= upDuration)
            {
                // 올라가는 것 완료 후 상태 전환
                ChangeState(State.TopWait);
            }
            break;
    }
}
```



업그레이드 요소 – 쫓아오는 적



```
void Update()
{
    if (player == null) return;

    // 발 아래 바닥 감지
    RaycastHit hit;
    bool onEnemyPlatform = false;

    // 적 발 바로 아래로 레이캐스트 발사 (약간 위에서 아래로)
    if (Physics.Raycast(transform.position + Vector3.up * 0.1f, Vector3.down, out hit, 2f))
    {
        // EnemyPlatform 스크립트가 붙어있는지 확인
        EnemyPlatform platform = hit.collider.GetComponent<EnemyPlatform>();
        if (platform != null)
        {
            onEnemyPlatform = true;
        }
    }
}
```

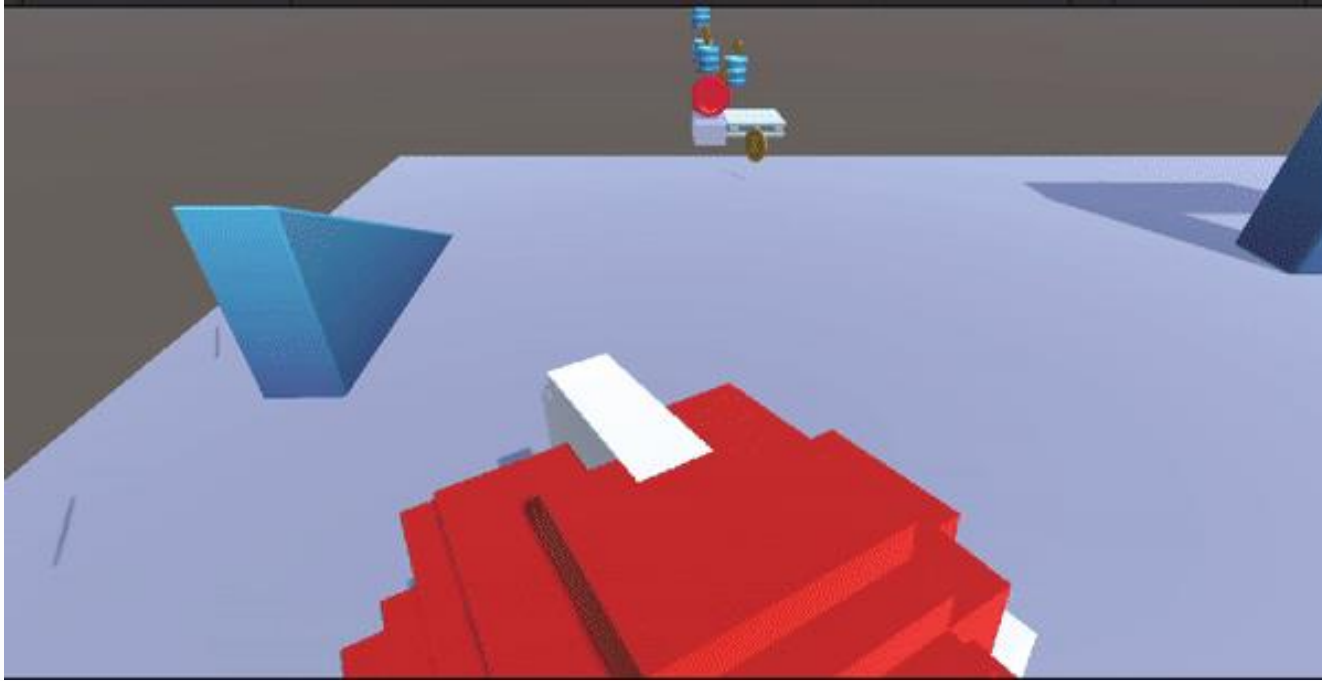
```
void MoveTowardsPlayer()
{
    // 플레이어와의 방향벡터 구하기
    Vector3 direction = (player.position - transform.position).normalized;

    // y축 고정 (상하 회전 방지)
    direction.y = 0f;

    // 부드러운 회전
    Quaternion targetRotation = Quaternion.LookRotation(direction);
    transform.rotation = Quaternion.Slerp(transform.rotation, targetRotation, rotationSpeed * Time.deltaTime);

    // 전진 이동
    transform.position += transform.forward * moveSpeed * Time.deltaTime;
}
```


업그레이드 요소 – 체크포인트



```
else if (other.CompareTag("CheckPoint"))
{
    // 체크포인트 획득
    other.gameObject.SetActive(false);
    lastCheckpointPosition = transform.position;
    checkpointReached = true;
}
else if (other.CompareTag("GameManager"))
{
    // 낙사 트리거에 닿았을 경우
    if (checkpointReached)
    {
        // 체크포인트 존재 시 그 위치로 복귀
        transform.position = lastCheckpointPosition;
        rigid.linearVelocity = Vector3.zero; // 속도 초기화
    }
    else
    {
        // 체크포인트 없으면 스테이지 재시작
        string currentStageName = "Stage" + manager.stage.ToString();
        SceneManager.LoadScene(currentStageName);
    }
}
```



점수

30점

**이유: 몇 개의 스테이지를 새로 제작하면서 업그레이드 요소가 다양하고,
게임 퀄리티도 높다고 생각하기 때문**



Reference

유튜브 골드메탈 - 기초만 꼭꼭 눌러담은 3D 게임 만들기

<https://www.youtube.com/watch?v=pTc1dakebow&t=3532s>

Chat GPT - O1

- 쫓아오는 적의 움직임 구현
- 움직이는 플랫폼 움직임 구현

<https://chatgpt.com/>





감사합니다