www.VRMiddleProject.com

VR Project

VR Drum

20195155 노규호

www.VRMiddleProject.com

INDEX

목차

STEP 1. 프로젝트 소개

VR Drum

www.VRMiddleProject.com

VR Drum - 스매쉬 드럼

VR 콘텐츠 중 Smash Drums라는 게임 콘텐츠에 영감을 받은 프로젝트

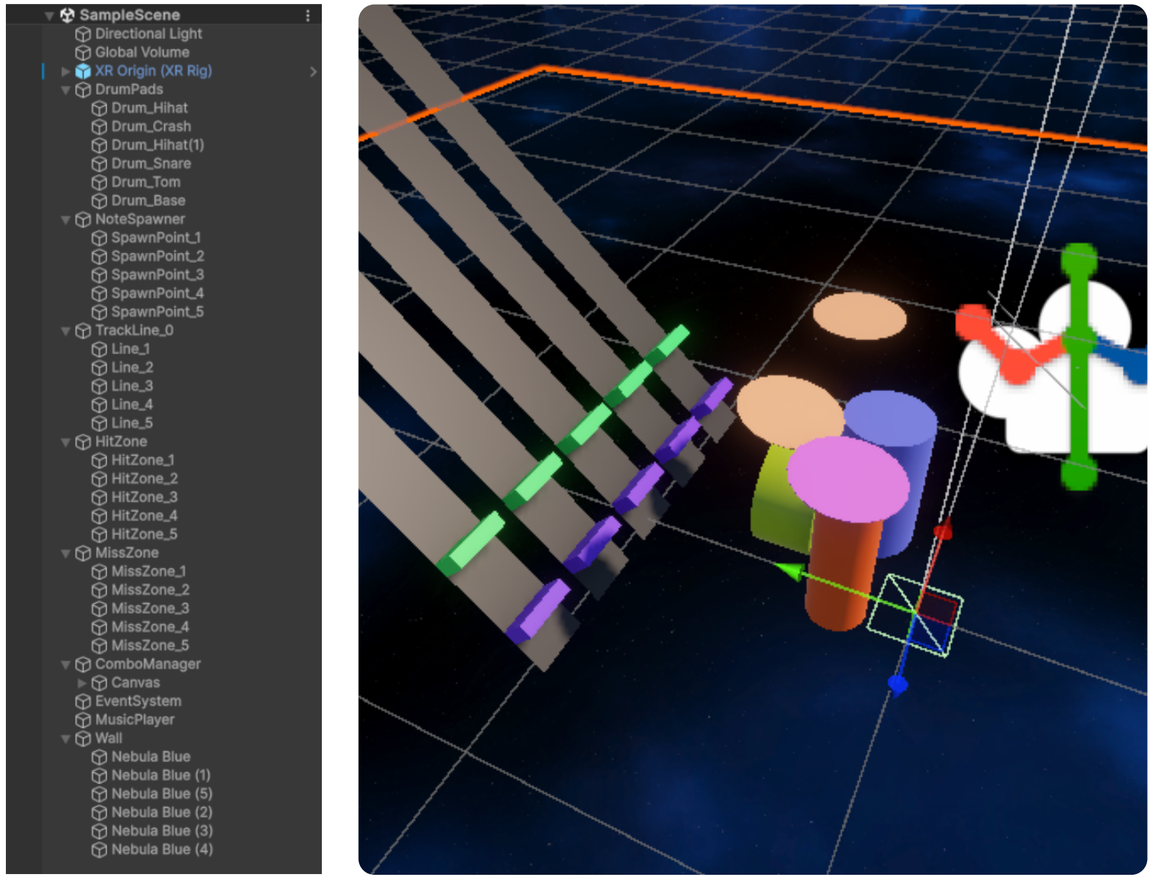
사용자 인터페이스에서 생성되는 노트를 타이밍에 맞춰 드럼패드를 드럼스틱으로 연결된 XR 기기로 치는 방식

콤보 시스템과 판정 시스템을 구성하여 사용자가 플레이 중에 UI로 자연스럽게 확인 가능

각 드럼패드에 해당하는 노트는 JSON 파일로 임의 구현한 데이터로 구성

STEP 2. 프로젝트 설명

VR Drum



www.VRMiddleProject.com

VR Drum - Scene 구성

가상의 공간에 3D 오브젝트들로 드럼패드를 구성

각 라인에서 내려오는 노트들이 초록색 판정박스에 들어올 때 사용자가 드럼패드를 히트하면 콤보가 올라가는 방식

판정박스 밖에서 히트하거나 보라색 판정박스까지 히트하지 못한 경우 MISS 처리

VR Drum 주요 Scripts - 1

```cs
using UnityEngine;

Unity 스크립트(자산 참조 5개)|참조 0개
public class DrumPad : MonoBehaviour
{
    public int laneIndex; // 드럼패드가 담당하는 라인 번호
    public AudioClip drumSound; // 드럼패드 히트사운드
    private AudioSource audioSource; // 드럼패드 사운드
    public JudgementUI judgementUI; // 판정 UI
    private float lastHitTime = -1f;
    public float hitCooldown = 0.15f;
    Unity 메시지|참조 0개
    private void Start()
    {
        audioSource = gameObject.AddComponent<AudioSource>();
        audioSource.clip = drumSound;
        audioSource.playOnAwake = false;
    }

    Unity 메시지|참조 0개
    private void OnTriggerEnter(Collider other)
    {
        if (other.CompareTag("DrumStick"))
        {
            if (Time.time - lastHitTime < hitCooldown)
                return; // 드럼스틱이 드럼패드에 닿을 때 너무 빠른 속도로 닿으면 판정되지 않도록 처리
                        // 드럼스틱이 드럼패드를 통과하여 2~3번씩 동시에 판정되는 것을 막음

            audioSource.PlayOneShot(drumSound);


            Note[] notes = Object.FindObjectsByType<Note>(FindObjectsSortMode.None); // 현재 Scene에 있는 Note 오브젝트를 배열에 저장해서 노트가 떨어지는 라인 번호와
                                                                                     // 드럼패드가 담당하고 있는 라인 번호를 검사하여 히트존 안에 있으면 판정 성공으로 판단
            foreach (Note note in notes)
            {
                if (note.laneIndex == laneIndex && note.IsInHitZone())
                {
                    FindFirstObjectByType<ComboManager>()?.IncreaseCombo(); // 판정 성공했을 때 콤보 1증가 시키는 함수 실행
                    judgementUI?.ShowJudgement("Perfect!");
                    Destroy(note.gameObject);
                    return;
                }
            }

            FindFirstObjectByType<ComboManager>()?.ResetCombo(); // 판정 실패했을 때 콤보 초기화
            judgementUI?.ShowJudgement("Miss!");
        }
    }
}
```

```cs
using TMPro;
using UnityEngine;
using UnityEngine.UI;

Unity 스크립트(자산 참조 1개)|참조 2개
public class ComboManager : MonoBehaviour
{
    public int currentCombo = 0;
    public TextMeshProUGUI comboText;

    Unity 메시지|참조 0개
    void Start()
    {
        UpdateComboUI();
    }

    참조 1개
    public void IncreaseCombo()
    {
        currentCombo++;
        UpdateComboUI();
    }

    참조 1개
    public void ResetCombo()
    {
        currentCombo = 0;
        UpdateComboUI();
    }

    참조 3개
    void UpdateComboUI()
    {
        comboText.text = currentCombo > 0 ? $"{currentCombo} Combo!" : "";
    }
}
```

VR Drum 주요 Scripts - 2

```cs
using UnityEngine;

Unity 스크립트(자산 참조 1개)|참조 4개
public class Note : MonoBehaviour
{
    public float speed = 2f;
    public int laneIndex;
    private bool isInHitZone = false;

    // 이동 방향
    private Vector3 moveDirection = new Vector3(0f, -1f, -1f);

    Unity 메시지|참조 0개
    void Update()
    {
        transform.Translate(moveDirection.normalized * speed * Time.deltaTime, Space.World);

        if (transform.position.y < -1f || transform.position.z < 0f) // 혹시나 바닥으로 내려가면 바로 Destroy
        {
            Destroy(gameObject);
        }
    }

    참조 1개
    public bool IsInHitZone()
    {
        return isInHitZone;
    }

    Unity 메시지|참조 0개
    private void OnTriggerEnter(Collider other)
    {
        if (other.CompareTag("HitZone")) // 노트가 히트존인지 판단
        {
            isInHitZone = true;
        }
        else if (other.CompareTag("MissZone")) // 노트가 미스존이면 바로 Destroy
        {
            Destroy(gameObject);
        }
    }

    Unity 메시지|참조 0개
    private void OnTriggerExit(Collider other)
    {
        if (other.CompareTag("HitZone"))
        {
            isInHitZone = false;
        }
    }
}
```

```cs
using UnityEngine;

Unity 스크립트(자산 참조 1개)|참조 0개
public class NoteSpawner : MonoBehaviour
{
    public TextAsset noteJsonFile;
    public GameObject notePrefab;
    public Transform[] spawnPoints; // 라인별 스폰 위치
    public Transform[] trackLines;  // 각 라인의 트랙 오브젝트
    public float spawnInterval = 1f;

    Unity 메시지|참조 0개
    private void Start()
    {
        InvokeRepeating(nameof(SpawnNote), 1f, spawnInterval);
    }

    참조 1개
    void SpawnNote()
    {
        int lane = Random.Range(0, spawnPoints.Length);

        // 노트 생성
        GameObject note = Instantiate(notePrefab, spawnPoints[lane].position, Quaternion.identity);
        note.GetComponent<Note>().laneIndex = lane;
        // 트랙의 기울기 방향을 따라 이동 방향 설정
        Vector3 direction = -trackLines[lane].up;
    }
}
```

VR Drum 주요 Scripts - 노트생성JSON

```
{
    "notes": [
        {
            "time": 1.0,
            "lane": 0
        },
        {
            "time": 2.0,
            "lane": 1
        },
        {
            "time": 3.0,
            "lane": 2
        },
        {
            "time": 4.0,
            "lane": 3
        },
        {
            "time": 5.0,
            "lane": 1
        },
        {
            "time": 6.0,
            "lane": 4
        }
    ]
}
```

현재는 라인별로 시간에 맞게 노트를 생성하고 있고,
드럼 악보를 Chat GPT를 통해 JSON 파일로 변환하여
구현

★BEST → MIDI 파일 변환기 구현을 통해 정확한 리듬에 드럼패드에 맞는 노트가 떨어지도록 구현

www.VRMiddleProject.com

STEP 3. 시연동영상

VR Drum

www.VRMiddleProject.com

STEP 4. 추후 개발방향

VR Drum

추후 개발 방향

1. 진짜 게임같은 UI / 디자인
2. 노트 판정 세분화
3. 히트 세기에 따른 사운드 변화
4. MIDI 파일 변환기를 만들어, 좀더 리듬감 있는 노트 생성