

3D콘텐츠 이론 및 활용

10주(1). GUI 오브젝트

- GUI
- PlayerPrefs 함수

학습목표

- GUI오브젝트를 통해 화면 텍스트를 처리할 수 있다.
- 필요한 데이터 값을 시스템에 저장하고 활용 할 수 있다.
- 기본제공되는 파티클 프리펩을 활용할 수 있다.

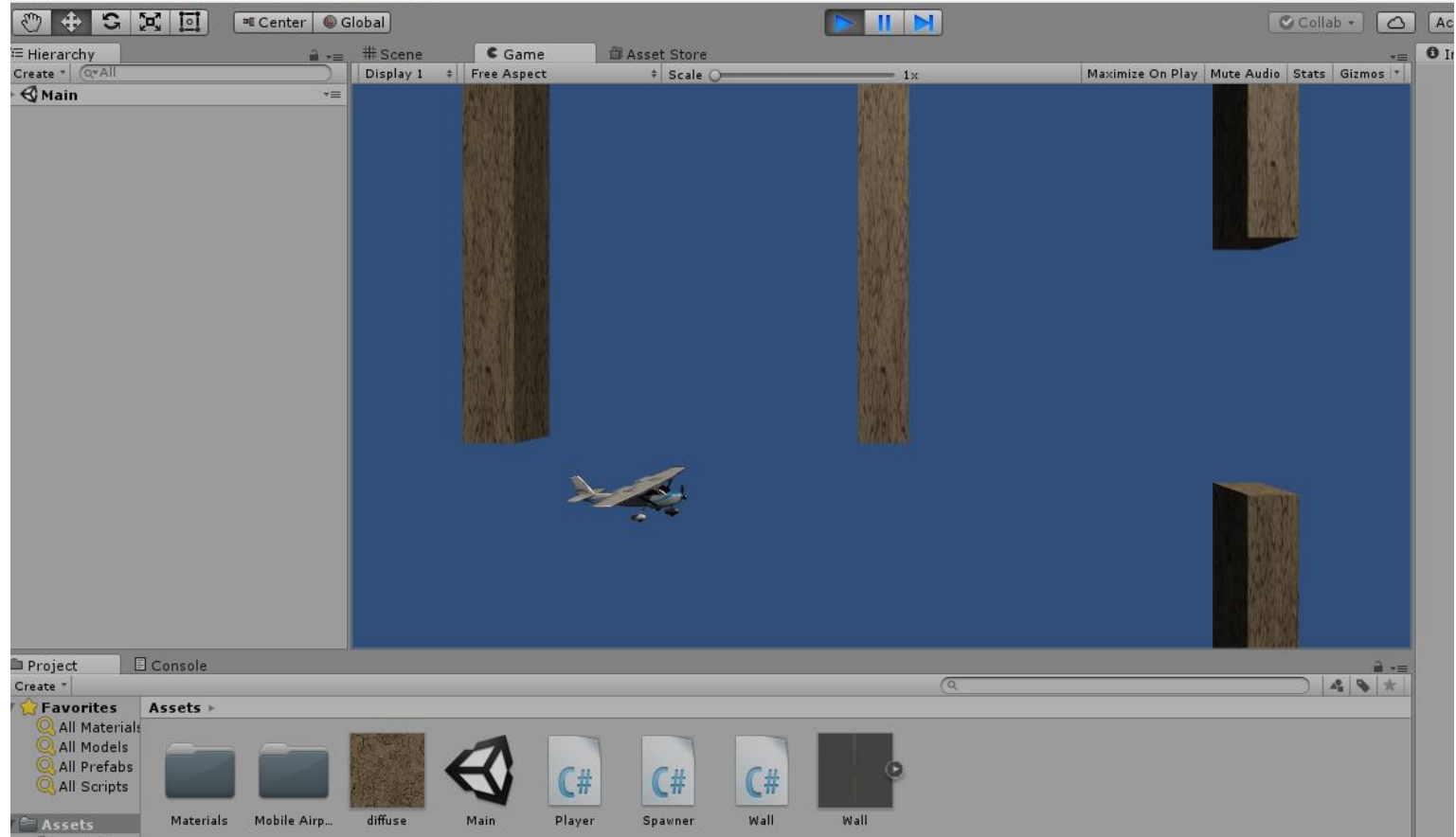
학습내용

- GUI 오브젝트
- PlayerPrefs 클래스
- 파티클 프리펩

1. 텍스처와 게임 오브젝트를 바꾸어 보자

Unity 5.5.0f3 Personal (64bit) - Main.unity - New Unity Project 13 - PC, Mac & Linux Standalone* <DX11>

File Edit Assets GameObject Component Window Help

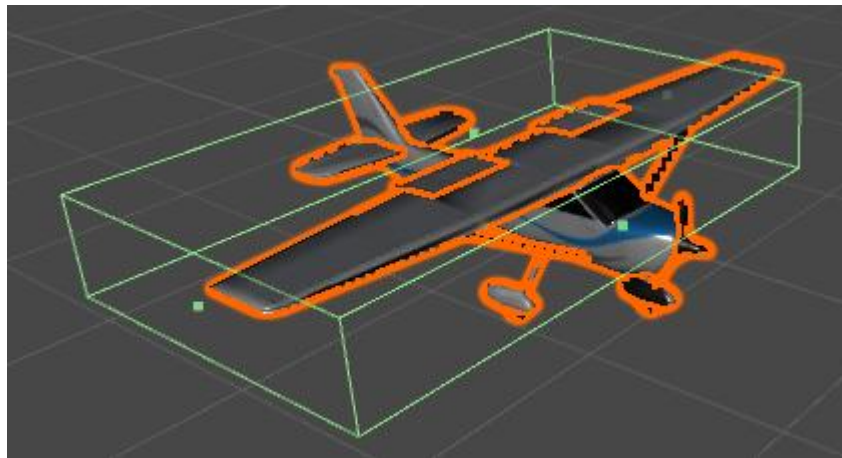


응용 실습1

1. 텍스처와 게임 오브젝트를 바꾸어 보자

플레이어 다운 (Plane 검색) - > Mobile Planes 선택

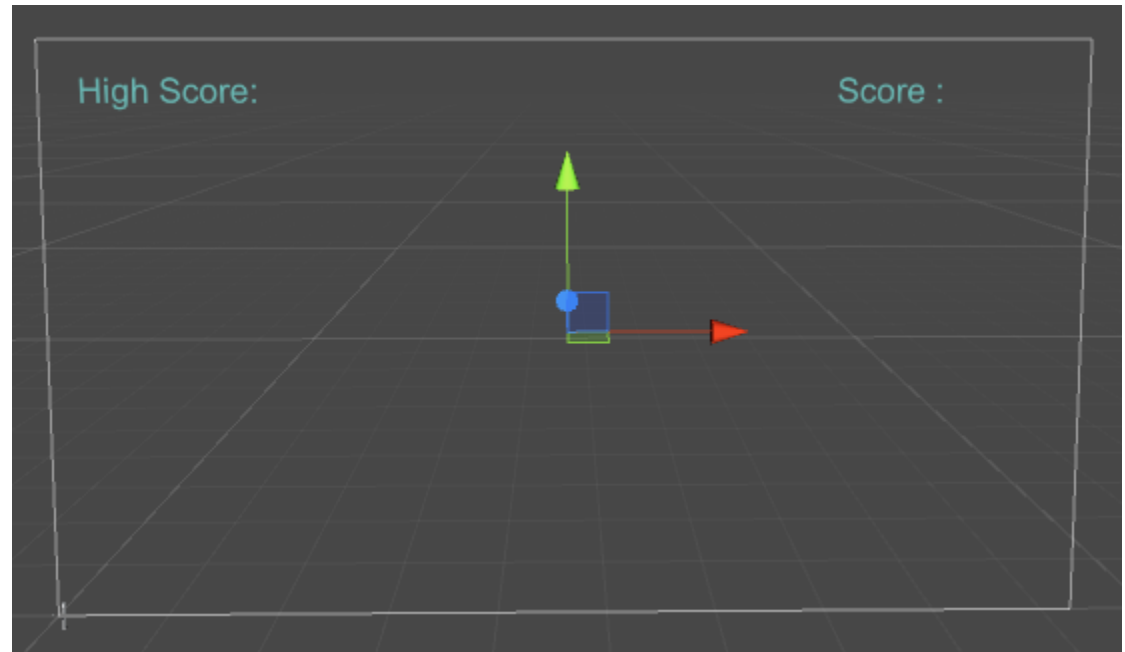
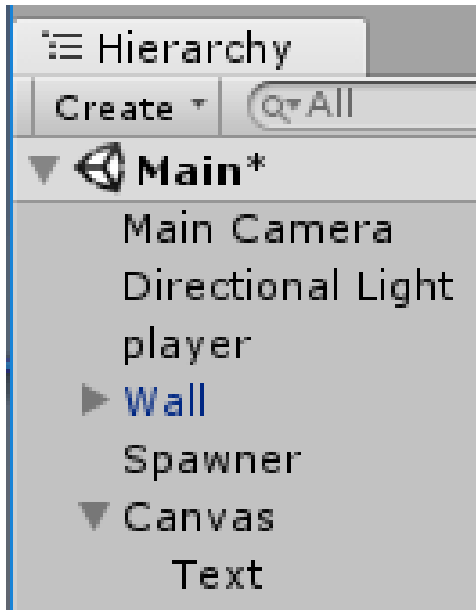
- 비행기의 방향 조정
- 비행기의 스케일 조정(0.3 ~ 0.4)
- 리지드바디 컴포넌트 추가
- 스크립트 추가 후 점프파워 5로 주기
- 박스콜라이드 컴포넌트 추가
- 이름 변경 및 기존 플레이어 삭제



1. GUI 활용

1) GUI 생성

- UI - Text 선택
- GUI 설정(폰트, 위치, 글자크기, 정렬 등)



1. GUI 활용

2) 점수처리 스크립트

- 조건: 게임을 오래 할 수록 포인트 제공
- 포인트는 타임클래스로 처리

```

using UnityEngine.UI; // 추가
public class ScoreGUI : MonoBehaviour
{
    public float RunningTime;
    public Text mText;

    void Start () {
        mText = GetComponent<Text> ();
    }

    void Update ()
    {
        RunningTime += Time.deltaTime;
        mText.text = "Score: " + RunningTime.ToString ("00");
    }
}
    
```

1. GUI 활용

3) 최고 점수처리 스크립트 #1/3



- PlayerPrefs 클래스를 활용하면 간단한 데이터를 시스템에 저장하고 불러오기를 할 수 있다.
- 종류 (SetInt, SetFloat, SetString)
- SetInt함수는 정수형 데이터를 저장하고, GetInt함수는 저장된 정수형 데이터를 불러올 때 사용한다.

```

public class ScoreGUI : MonoBehaviour {
    public float BestScore; // 추가

    // Use this for initialization
    void Start () {
        BestScore = PlayerPrefs.GetFloat ("BestScore"); //추가
        RunningTime = 0;
    }
    
```

1. GUI 활용

3) 최고 점수처리 스크립트 #2/3

- 최고기록과 현재 기록을 비교하여 최고점수 저장

// Update is called once per frame

```
void Update () {
```

```
    RunningTime += Time.deltaTime;
```

```
    mText.text = "Score: " + RunningTime.ToString ("00");
```

```
    if (RunningTime > BestScore) { // 조건문 추가
```

```
        PlayerPrefs.SetFloat ("BestScore", RunningTime);
```

```
    }
```

```
}
```

```
}
```


1. GUI 활용

3) 최고 점수처리 스크립트 #3/3

- 최고점수 출력(BestScore.cs) 스크립트 적용

```

public class BestScore : MonoBehaviour {
    public static float RunningTime;
    public float HighScore;
    public Text mText;

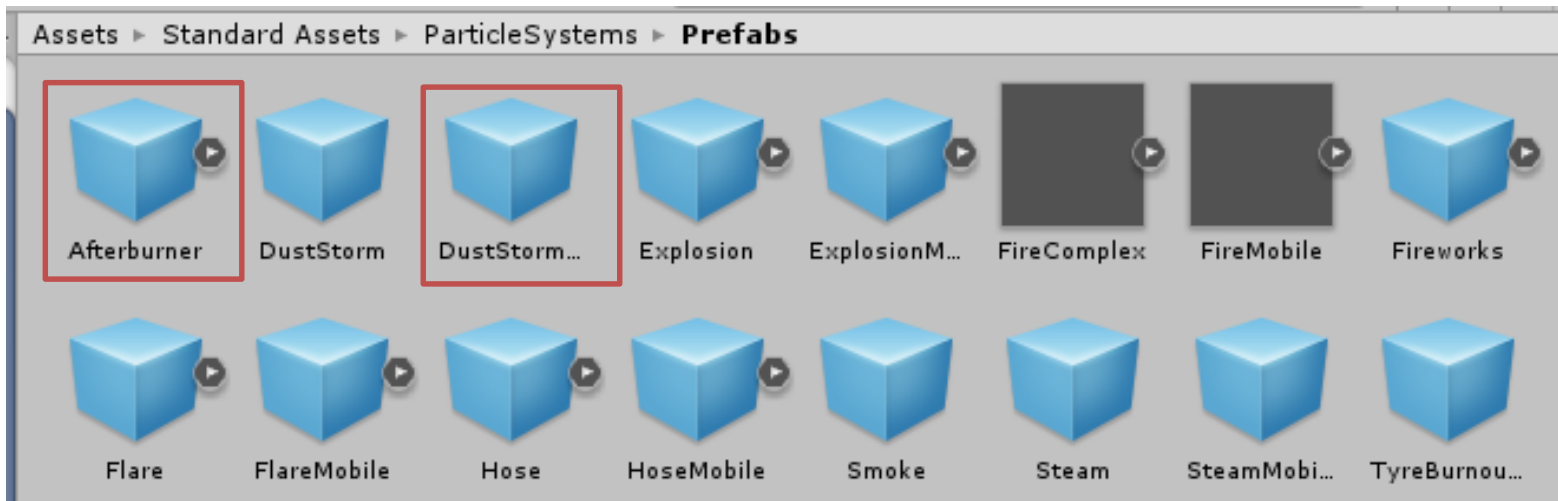
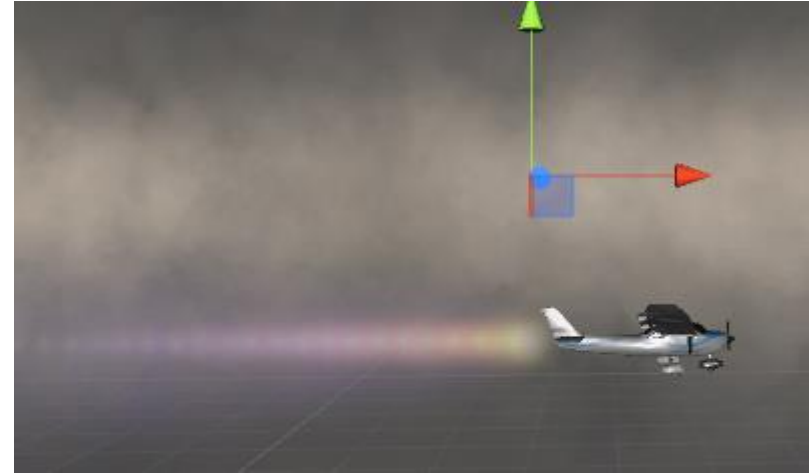
    void Start () {
        mText = GetComponent<Text> ();
        HighScore = PlayerPrefs.GetFloat ("BestScore");
        mText.text = "High Score: " + HighScore.ToString ("000");
    }

    void Update () {
    }
}
    
```

2. 파티클 활용

파티클을 추가하여 사실감을 더해 보자

- ParticleSystems import
- 파티클 프리팹 활용
 - [DustStorm]
 - start color 밝게
 - 최고 파티클 수 = 100개
 - 중력을 주어 비행기의 긴장감을 줌 = 0.39
 - [Afterburner]



3. 난이도 조절하기

시간이 지날 수록 스폰 시간을 줄여준다.

- Spawner 클래스

```

void Update ()
{
    if (Timer >= interval) {
        Timer = 0.0f;
        transform.position = new Vector3 (transform
        .position.x, Random.Range (-
        range, range), transform.position.z);
        Instantiate (wallPrefab, transform.position
        , transform.rotation);
        interval -= 0.02f; // 추가
    } else {
        Timer += Time.deltaTime;
    }
}
}

```

1

화면상의 텍스처는 GUI오브젝트를 이용하여 처리한다.

2

PlayerPrefs 클래스를 활용하면 간단한 데이터를 시스템에 저장하고 불러오기를 할 수 있다.(SetInt, SetFloat, SetString)