

게임엔진 이해와 실습

중간고사 대체 - 개인별 과제

디지털콘텐츠공학과 20183302 김수진



Index

01 / 게임 소개

02 / 기존 게임 및 추가한 점

03 / 수정·보완 방법

04 / 최종 목표



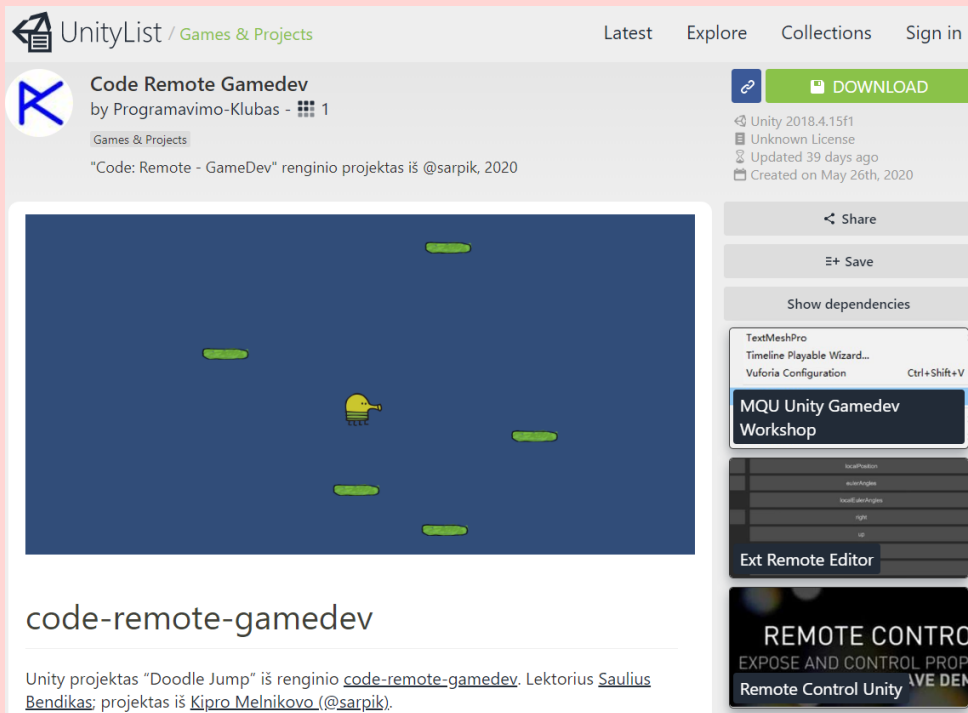
A falling game

Game Start

'A falling game'은 캐릭터가 떨어지면서 아이템도 얻고, 장애물을 피하면서 기록을 세우는 게임이다.



02 / 기존 게임 및 추가한 점



UnityList / Games & Projects Latest Explore Collections Sign in

Code Remote Gamedev
by Programavimo-Klubas - 1

Games & Projects

"Code: Remote - GameDev" renginio projektas iš @sarpik, 2020

DOWNLOAD

Unity 2018.4.15f1
Unknown License
Updated 39 days ago
Created on May 26th, 2020

Share

Save

Show dependencies

TextMeshPro
Timeline Playable Wizard...
Vuforia Configuration Ctrl+Shift+V

MQU Unity Gamedev Workshop

Ext Remote Editor

REMOTE CONTROL
EXPOSE AND CONTROL PROP
Remote Control Unity

code-remote-gamedev

Unity projektas "Doodle Jump" iš renginio [code-remote-gamedev](#). Lektorius [Saulius Bendikas](#); projektas iš [Kipro Melnikovo \(@sarpik\)](#).

Code Remote Gamedev
-> 단순히 캐릭터가 좌우로 움직이도록 되어있던 게임

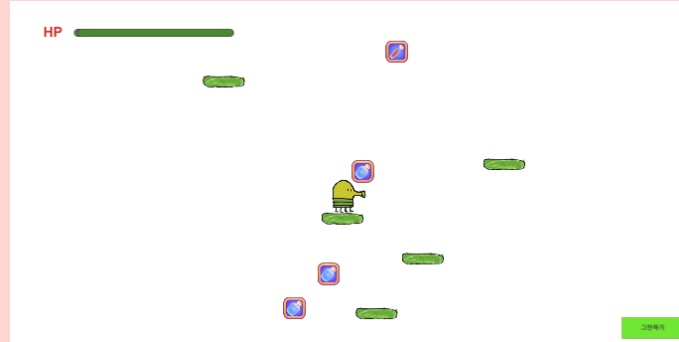
이번 수업에서 배운 내용을 중심으로 추가하고 싶어, 최대한으로 퀄리티가 낮은 게임을 고르게 되었다.



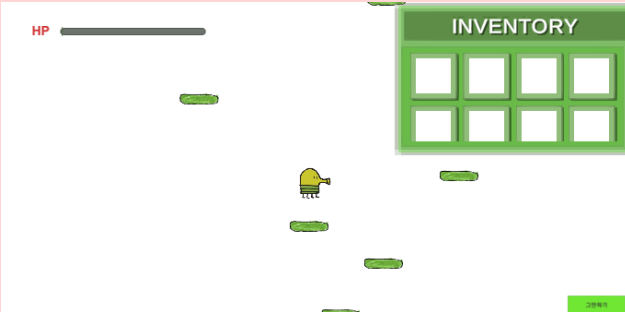
A falling game

Game Start

1. 게임 시작 화면을 추가하여 버튼을 눌러 게임 화면으로 전환시킴



2. Hp와 아이템을 추가 시켜 게임의 재미를 높임



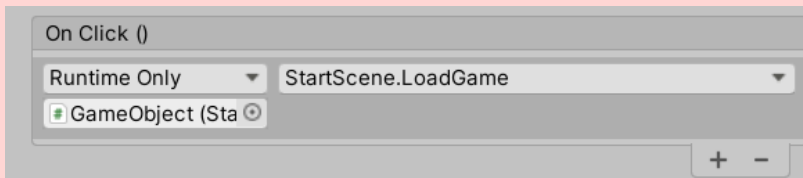
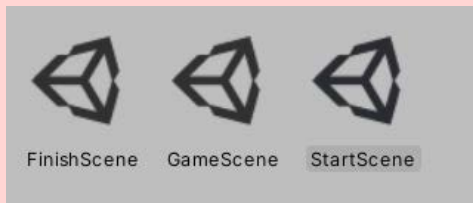
3. 인벤토리 창을 추가하여 아이템을 얻을 때, 창에 추가 할 수 있도록 함

게임 해 주셔서
감사합니다 :)

다시 하기 !

4. 게임 종료 버튼과 다시하기 버튼을 추가하여 게임화면에서 화면을 전환시킴



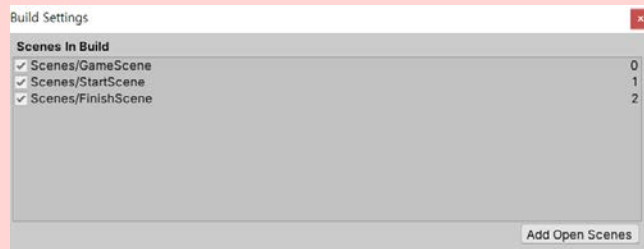
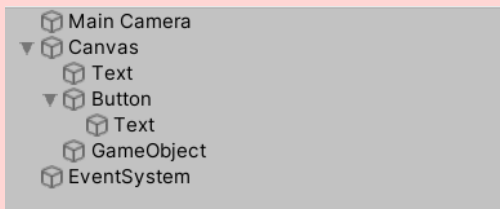


다시 하기 !

그만하기

Game Start

-> 버튼마다 게임오브젝트와 전환 될 scene을 load 시킴



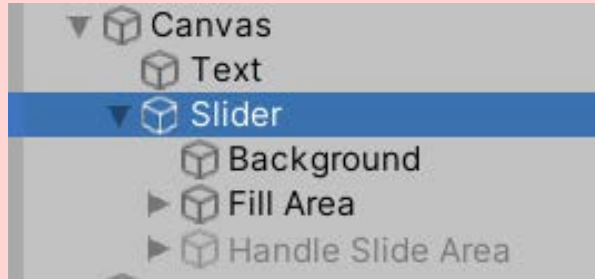
-> Build Settings창에서 빌드 할 scene을 추가 시킴 (추가 하지 않을 시 화면 전환이 안됨)

```
using UnityEngine.SceneManagement;
```

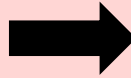
```
public void LoadGame()
{
    SceneManager.LoadScene("GameScene");
}
```

-> 총 세번의 화면 전환이 있기 때문에 새로운 scene을 만들고, 각자 화면에서 버튼을 추가시키고, 위의 스크립트를 추가하여 전환 할 수 있도록 함.





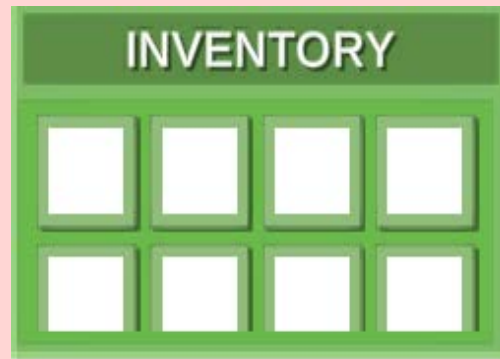
-> ui - slider를 사용하여 hp를 나타냈고, 원래는 장애물과 부딪치는 경우에 hp가 닳도록 해야 하지만, 장애물을 만드는 과정에서 오류가 발생하여 키보드 A를 눌렀을 때 (움직일 때) 소모가 되도록 하였음.



```
void Update()
{
    if (Input.GetKeyDown(KeyCode.A))
    {
        if (curHp > 0)
        {
            curHp -= 10;
        }
        else
        {
            curHp = 0;
        }
    }
    HandleHp();
}

private void HandleHp()
{
    hpbar.value = Mathf.Lerp(hpbar.value,
    (float)curHp / (float)maxHp, Time.deltaTime*10);
}
}
```





```

if (Input.GetKeyDown(KeyCode.I))
{
    activeInventory
= !activeInventory;

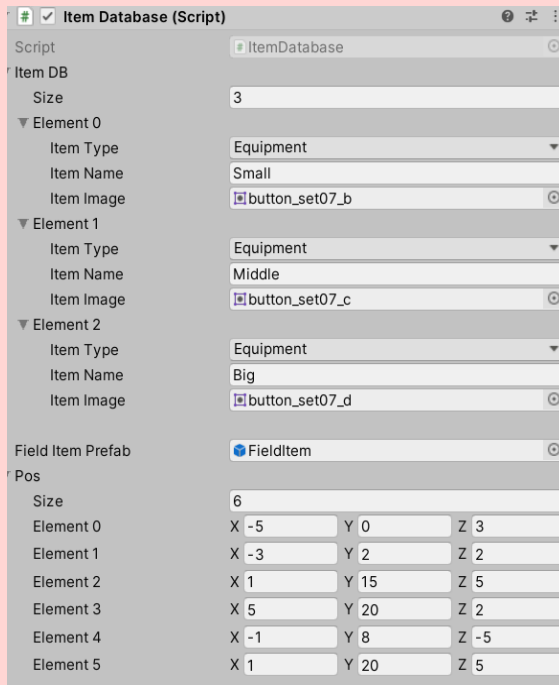
inventoryPanel.SetActive(active
Inventory);
}

```



-> 게임에서 얻은 아이템을 확인하고자 하면 I 버튼을 눌러 인벤토리 창을 띄움





```

public List<Item> itemDB = new
List<Item>();
    [Space(20)]
    public GameObject
    fieldItemPrefab;
    public Vector3[] pos;

    private void Start()
    {
        for(int i = 0; i<5; i++)
        {
            GameObject go =
            Instantiate(fieldItemPrefab,
            pos[i], Quaternion.identity);

            go.GetComponent<FieldItems>().Se
            tItem(itemDB[Random.Range(0,
            3)]);
        }
    }

```

-> 아이템의 사이즈를 지정해 개수를 정하고, 아이템 이미지 prefab을 script를 통해 위치도 설정함



1. 캐릭터와 부딪칠 장애물 추가
- 오류가 나서 뺐
2. 단계가 지날 때마다 레벨 업이 되도록 하고, 뒷배경이 단계별로 바뀌면서 새로운 맵을 생성
3. 캐릭터, 아이템 추가
4. 점수 기능 추가 - 기록 갱신

-> 개인의 능력 부족으로 미숙하지만, 게임을 **완성**시켜 안드로이드 플랫폼에 지원되도록 빌드시키고, **앱스토어에 등록**



한학기 동안 모두 수고하셨습니다 !!

Thank you ;)

