



# [6주차] 2D RunGame-3

---

인천정보과학고등학교 3학년 전산과 게임프로그래밍2

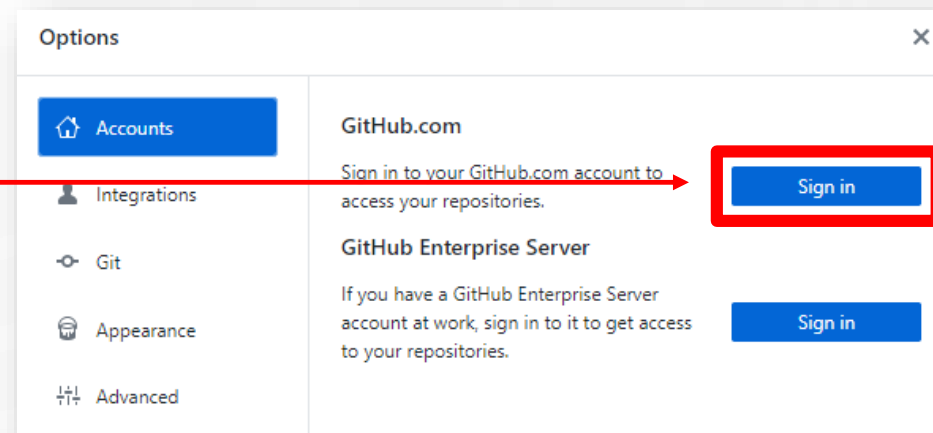
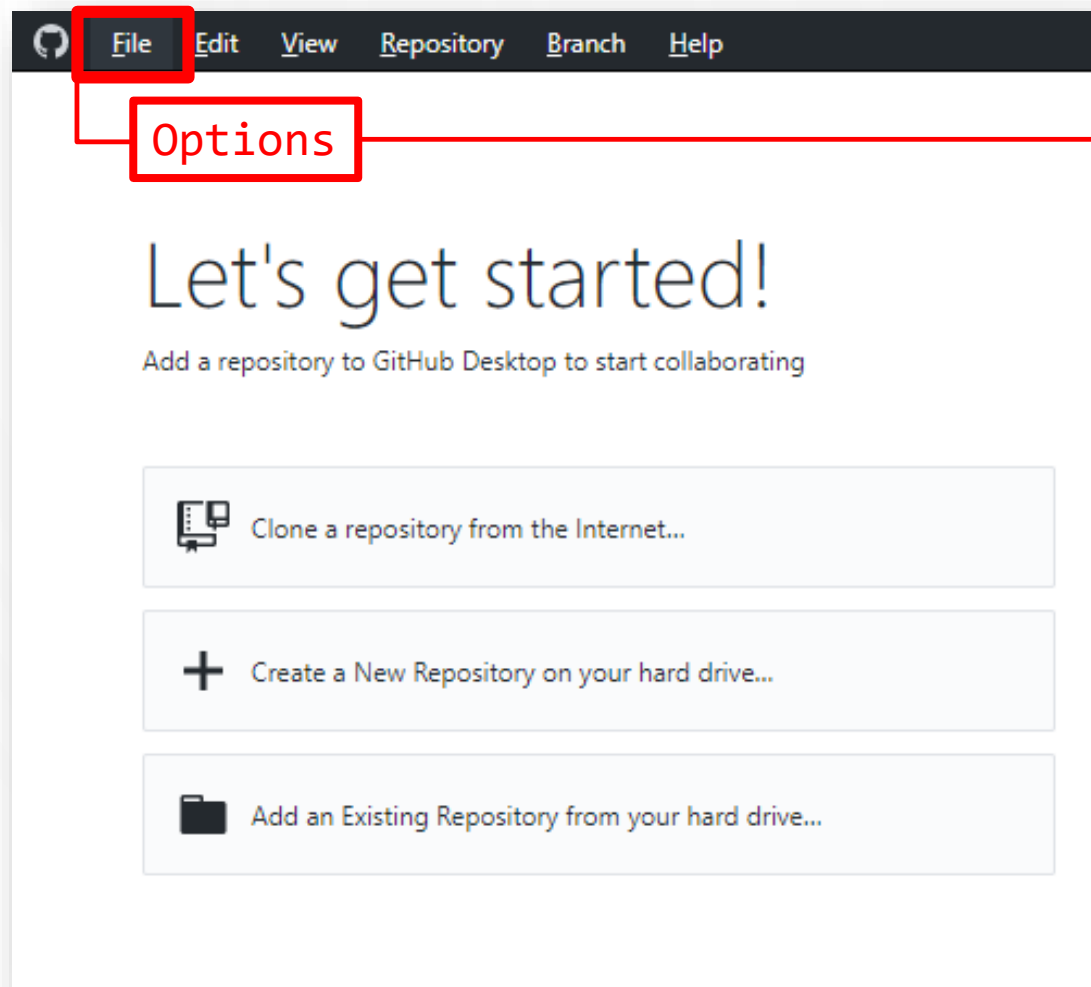
© all right reserved 인천정보과학고 IT소프트웨어과 정수영

# GITHUB CLONE

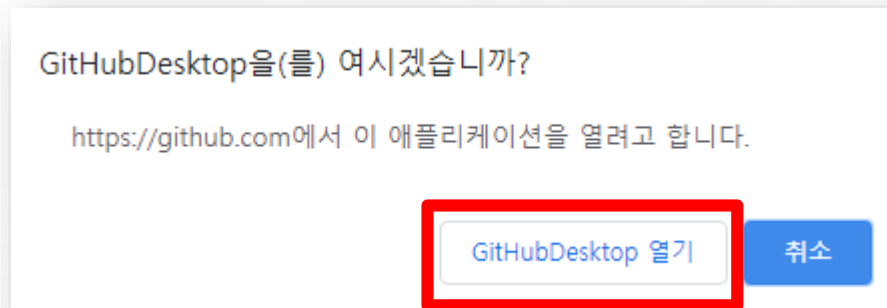
---



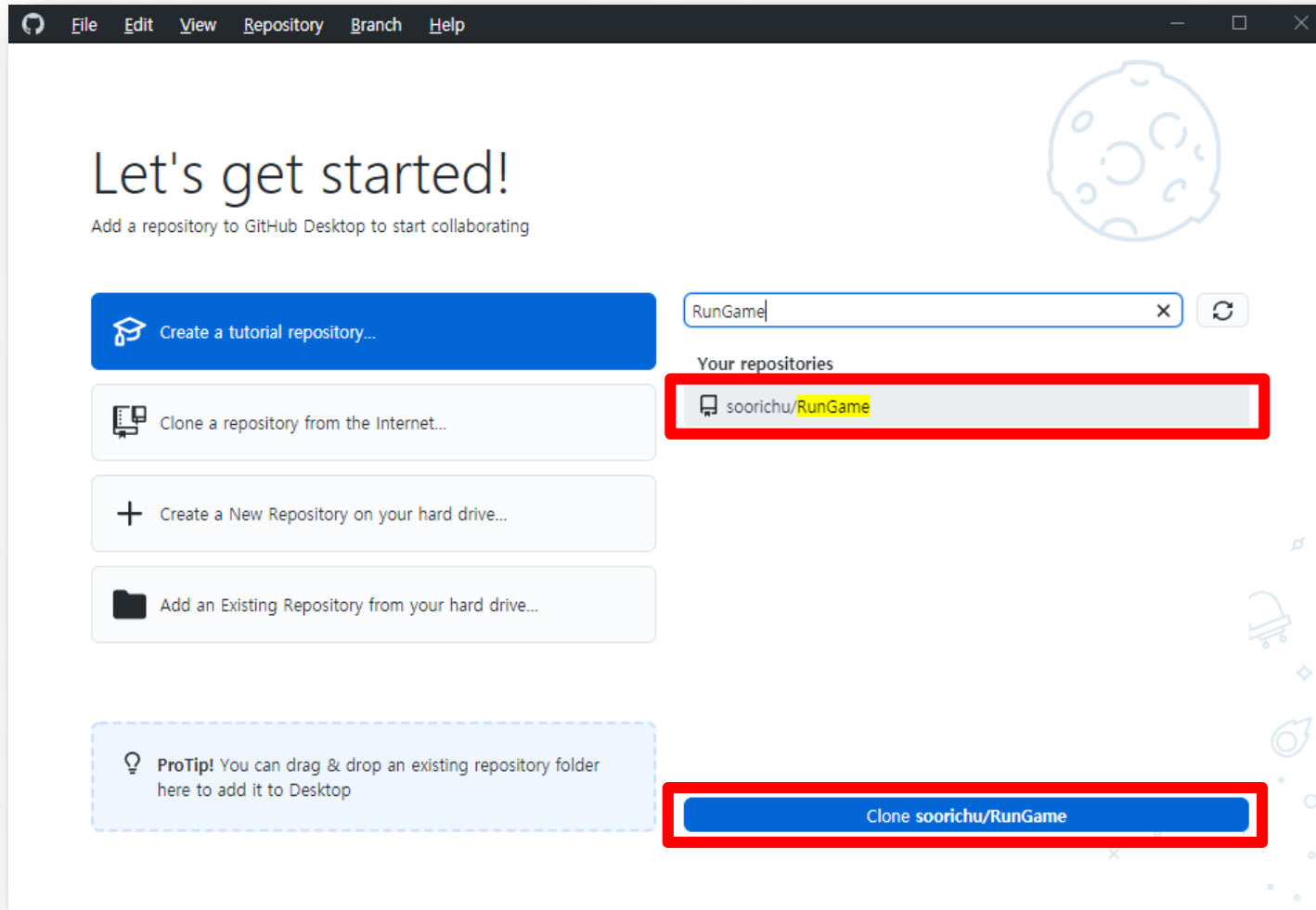
# Git Desktop Login



깃헙 로그인 진행  
크롬에서 다음과 같은 메시지가 나오면



# Clone Repository



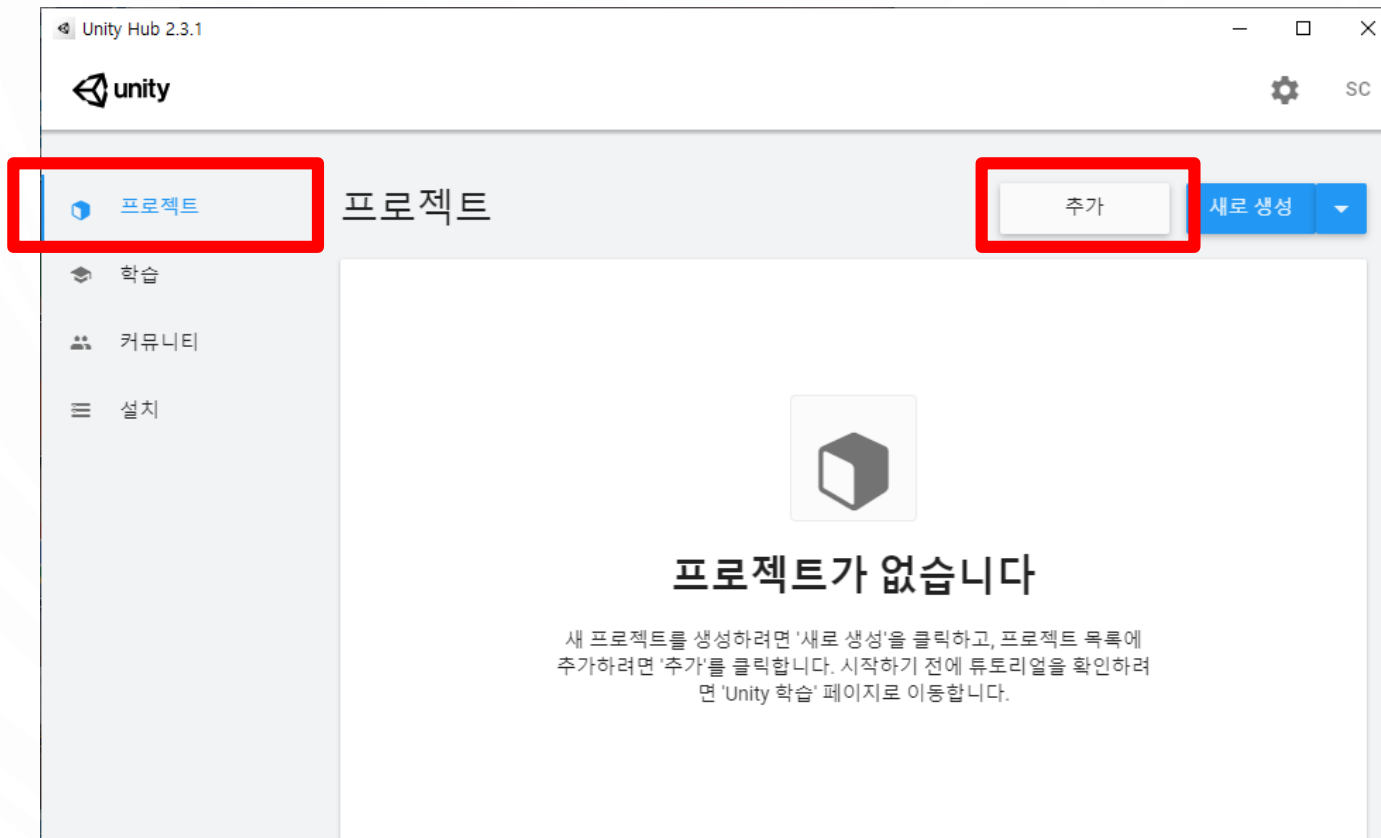
<user>/RunGame 선택

Clone <user>/RunGame 클릭

# Project Open



- Unity Hub에서 프로젝트 > 추가 > RunGame 폴더 선택

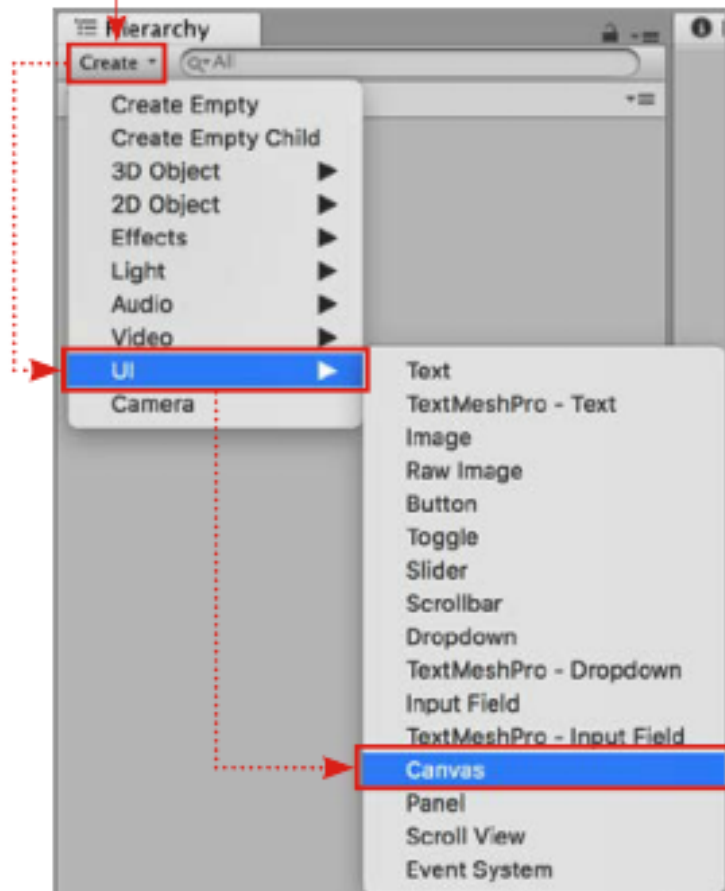


# 게임 UI 만들기

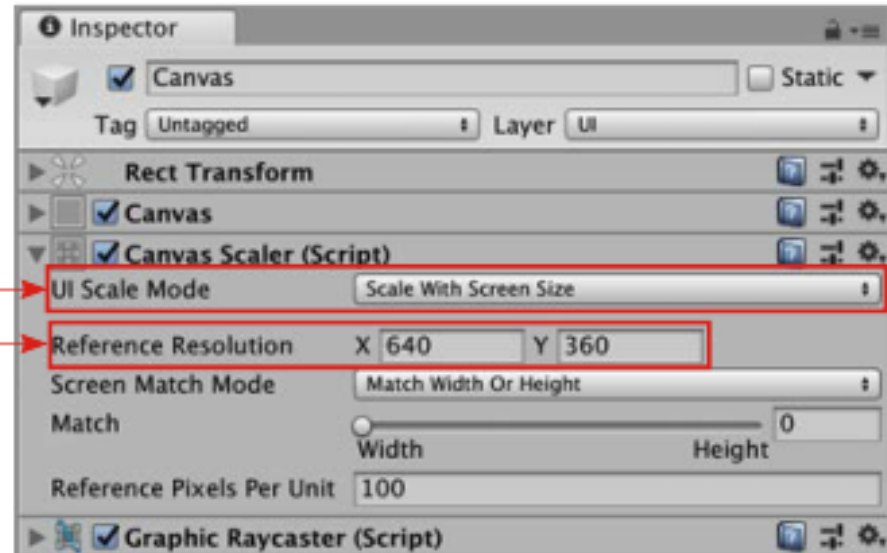
---

# Canvas 만들기

① Canvas 게임 오브젝트 생성(Create > UI > Canvas)



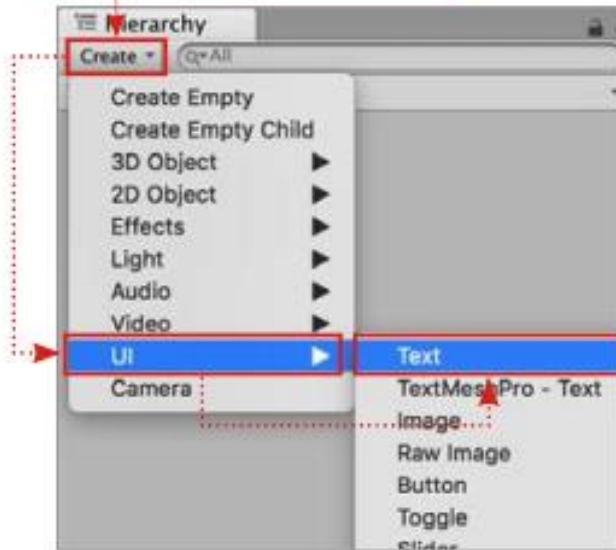
② Canvas Scaler 컴포넌트의 UI Scale Mode를 Scale With Screen Size로 변경



③ Reference Resolution을 (640, 360)으로 변경

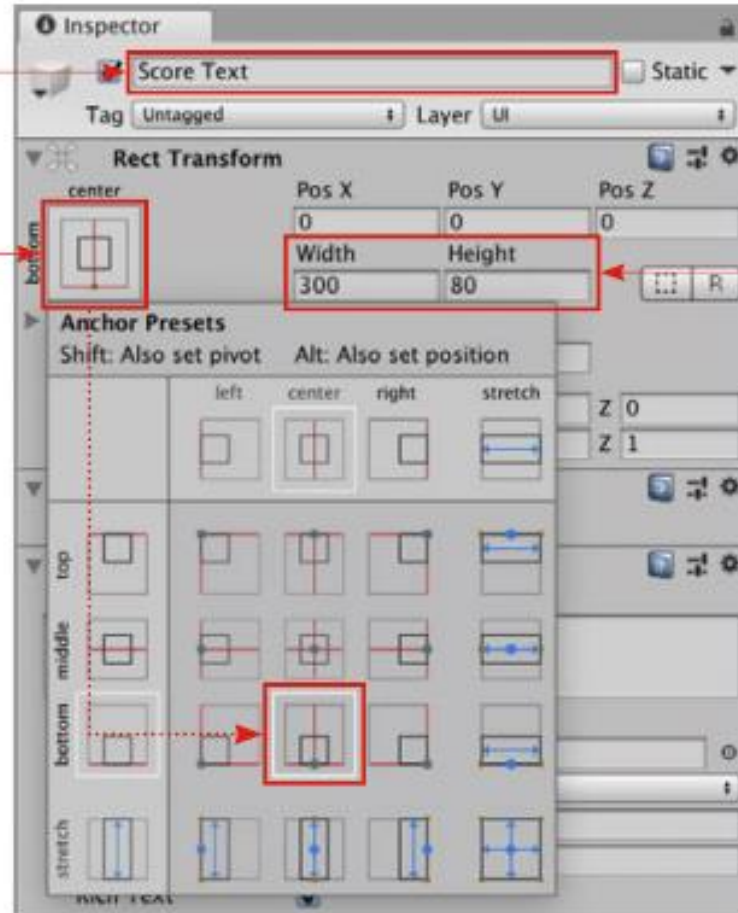
# 점수 UI 텍스트 만들기

① 새로운 Text 게임 오브젝트 생성  
(Create > UI > Text)



② 이름을 Score Text로 변경

③ Width를 300, Height를 80으로 변경



④ Anchor Preset 클릭 > [Alt+Shift]를 누른 채로 bottom-center 클릭

▶ 점수 UI 텍스트 만들기



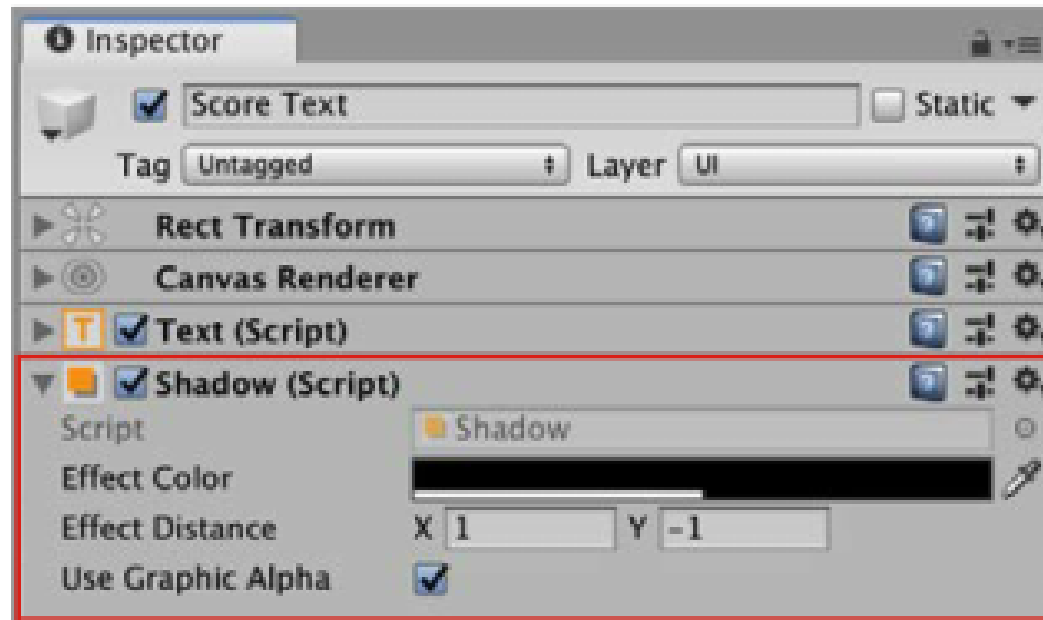
# Score Text의 컴포넌트 설정

The image shows the Unity Inspector window for a 'Score Text' component. The component is currently set to 'Untagged' and 'UI' layer. The 'Text' field displays 'Score : 0'. The 'Character' section shows the 'Font' set to 'Kenney Mini Square', 'Font Style' as 'Normal', 'Font Size' as '50', 'Line Spacing' as '1', and 'Rich Text' checked. The 'Paragraph' section shows 'Alignment' set to 'Center, Middle' (indicated by the center and middle alignment icons). The 'Color' field is currently empty. The 'Material' is set to 'None (Material)' and 'Raycast Target' is checked.

Annotations and steps:

- ① Text 필드를 Score : 0으로 변경
- ② Font 필드 옆의 선택 버튼 클릭 > Kenney Mini Square 폰트 더블 클릭
- ③ Font Size를 50으로 변경
- ④ Alignment를 Center, Middle로 변경
- ⑤ Color 필드 클릭 > 폰트 컬러를 (255, 255, 255)로 변경

# Score Text에 그림자 추가



▶ Score Text에 그림자 추가

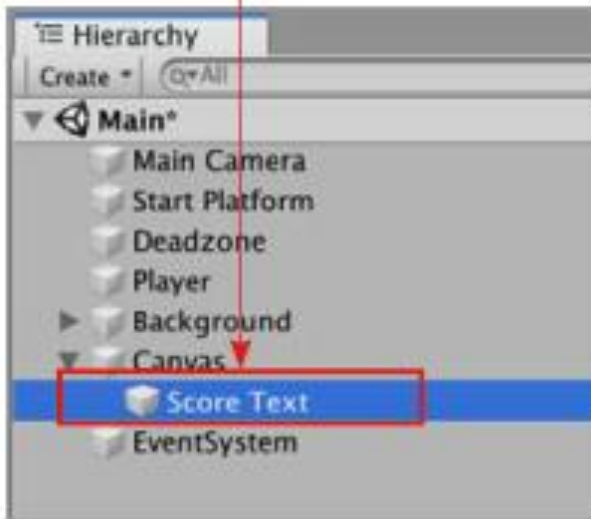


▶ 완성된 Score Text 게임 오브젝트의 모습

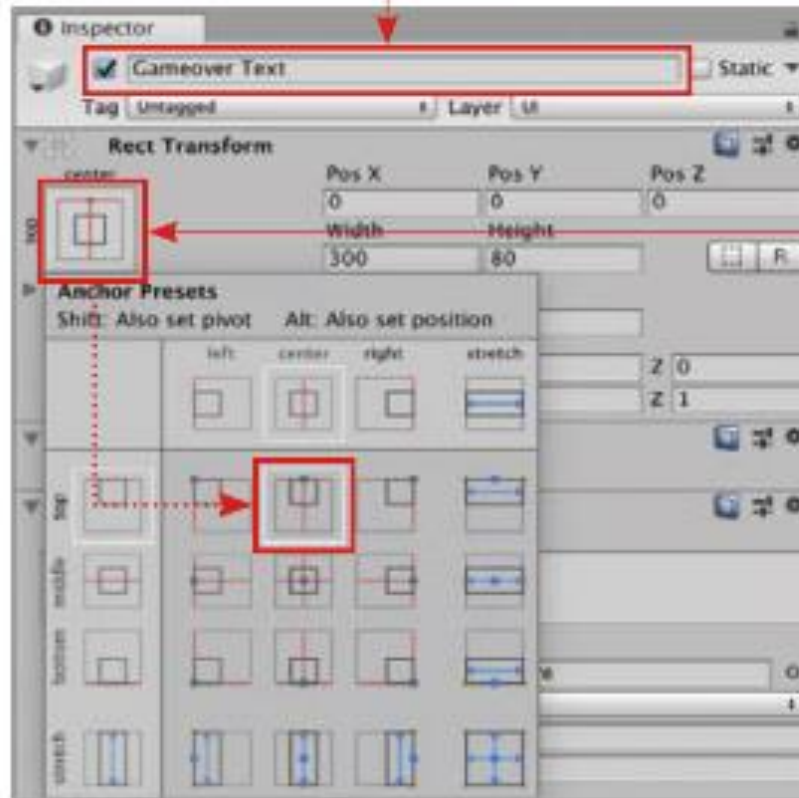
① Shadow 컴포넌트 추가(Add Component > UI > Effects > Shadow)

# GameOver Text 생성

① Score Text 선택 > [Ctrl+D]로 복제



② Score Text (1)의 이름을 Gameover Text로 변경

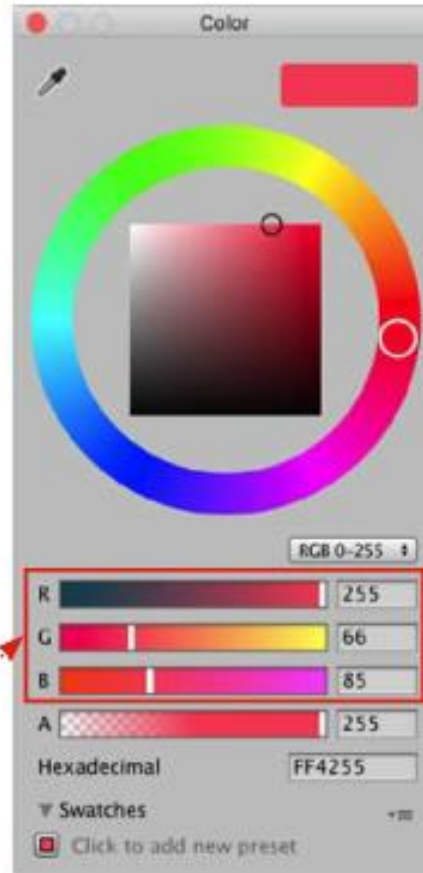
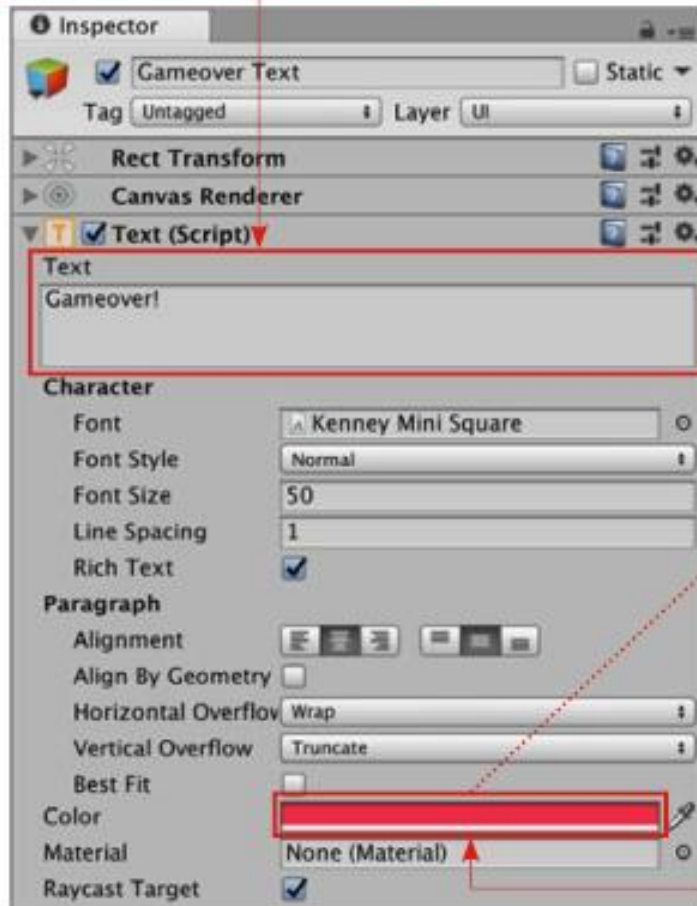


③ Anchor Preset 클릭 > [Alt+Shift]를 누른 채 top-center 클릭

▶ 게임오버 텍스트 생성

# GameOver Text 컴포넌트 설정

① Text 컴포넌트의 Text 필드 값을 Gameover!로 변경



② Color 필드 클릭 > 폰트 컬러를 (255, 66, 85)로 변경

# Restart Text 생성

① Gameover Text 선택 > [Ctrl+D]로 복제

② 이름을 Restart Text로 변경

③ Pos Y값을 -40으로 변경

④ Text 필드를 Jump To Restart로 변경

⑤ Font Size를 33으로 변경

⑥ Restart Text를 Gameover Text의 자식으로 만들기 (Restart Text를 Gameover Text로 드래그&드롭)

The image shows two screenshots of the Unity interface. The top screenshot shows the Hierarchy panel with 'Gameover Text' selected. The bottom screenshot shows the Hierarchy panel with 'Restart Text' selected. The Inspector panel shows the properties of the selected object, including 'Rect Transform' and 'Text (Script)'. Red boxes and arrows highlight the specific changes made in each step.

# GameOver Text 비활성화

완성된 UI 모습



GameOver Text와 Restart Text는 게임 진행 중에는 숨겨야 하기 때문에 Gameover Text를 비활성화합니다. (Restart는 Gameover에 종속되어 있음.)



# GAMEMANAGER 스크립트

---

# GameManager 스크립트(1)

```
1  using UnityEngine;
2  using UnityEngine.SceneManagement;
3  using UnityEngine.UI;
4
5  // 게임 오버 상태를 표현하고, 게임 점수와 UI를 관리하는 게임 매니저
6  // 씬에는 단 하나의 게임 매니저만 존재할 수 있다.
7  public class GameManager : MonoBehaviour {
8      public static GameManager instance; // 싱글톤을 할당할 전역 변수
9
10     public bool isGameOver = false; // 게임 오버 상태
11     public Text scoreText; // 점수를 출력할 UI 텍스트
12     public GameObject gameOverUI; // 게임 오버시 활성화 할 UI 게임 오브젝트
13
14     private int score = 0; // 게임 점수
15
16     // 게임 시작과 동시에 싱글톤을 구성
17     void Awake() {
18         // 싱글톤 변수 instance가 비어있는가?
19         if (instance == null)
20         {
21             // instance가 비어있다면(null) 그곳에 자기 자신을 할당
22             instance = this;
23         }
24         else
25         {
26             // instance에 이미 다른 GameManager 오브젝트가 할당되어 있는 경우
27
28             // 씬에 두개 이상의 GameManager 오브젝트가 존재한다는 의미.
29             // 싱글톤 오브젝트는 하나만 존재해야 하므로 자신의 게임 오브젝트를 파괴
30             Debug.LogWarning("씬에 두개 이상의 게임 매니저가 존재합니다!");
31             Destroy(gameObject);
32         }
33     }
```

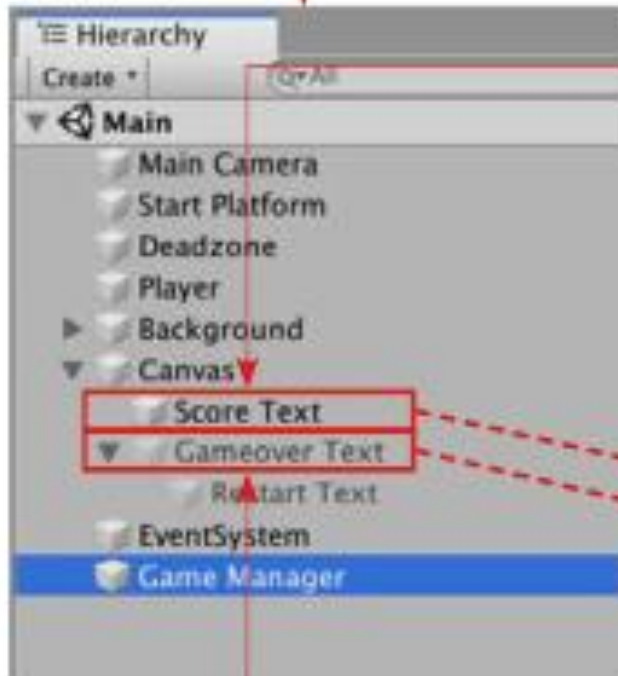


# GameManager 스크립트(2)

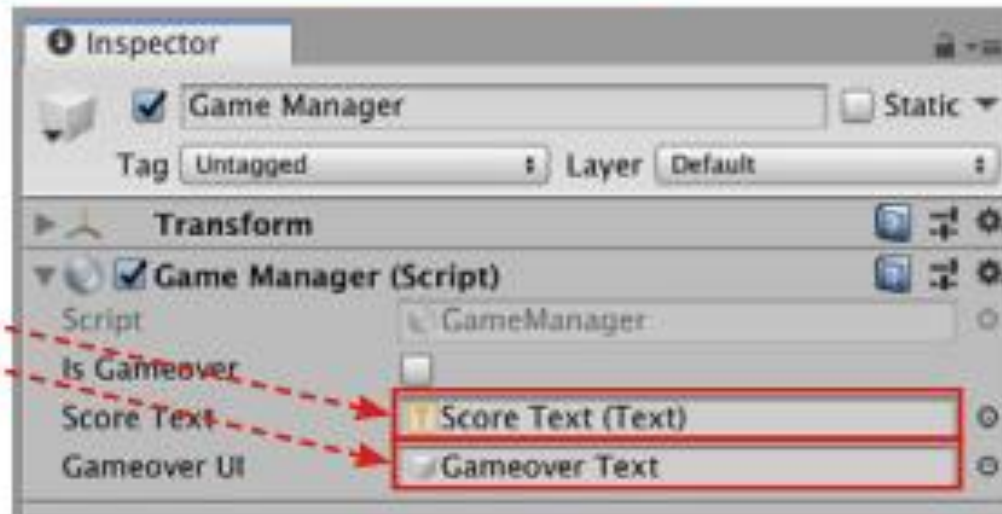
```
34
35 void Update() {
36     if (isGameOver && Input.GetMouseButtonDown(0))
37     {
38         // 게임 오버 상태에서 마우스 왼쪽 버튼을 클릭하면 현재 씬 재시작
39         SceneManager.LoadScene(SceneManager.GetActiveScene().name);
40     }
41 }
42
43 // 점수를 증가시키는 메서드
44 public void AddScore(int newScore) {
45     // 게임 오버가 아니라면
46     if (!isGameOver)
47     {
48         // 점수를 증가
49         score += newScore;
50         scoreText.text = "Score : " + score;
51     }
52 }
53
54 // 플레이어 캐릭터가 사망시 게임 오버를 실행하는 메서드
55 public void OnPlayerDead() {
56     // 현재 상태를 게임 오버 상태로 변경
57     isGameOver = true;
58     // 게임 오버 UI를 활성화
59     gameOverUI.SetActive(true);
60 }
61 }
```

# GameManager 컴포넌트 설정

① 하이어라키 창에서 GameManager 게임 오브젝트 선택



② Score Text 게임 오브젝트를 Score Text 필드로 드래그&드롭



③ Gameover Text 게임 오브젝트를 Gameover UI 필드로 드래그&드롭

# PlayerController 스크립트 수정

---

```
private void Die() {  
    // 애니메이터의 Die 트리거 파라미터를 셋  
    animator.SetTrigger("Die");  
  
    // 오디오 소스에 할당된 오디오 클립을 deathClip으로 변경  
    playerAudio.clip = deathClip;  
    // 사망 효과음 재생  
    playerAudio.Play();  
  
    // 속도를 제로(0, 0)로 변경  
    playerRigidbody.velocity = Vector2.zero;  
    // 사망 상태를 true로 변경  
    isDead = true;  
  
    // 게임 매니저의 게임 오버 처리 실행  
    GameManager.instance.OnPlayerDead();  
}
```

# ScrollingObject 스크립트 수정

---

```
using UnityEngine;

// 게임 오브젝트를 계속 왼쪽으로 움직이는 스크립트
public class ScrollingObject : MonoBehaviour {
    public float speed = 10f; // 이동 속도

    private void Update() {
        // 게임 오버가 아니라면
        if (!GameManager.instance.isGameOver)
        {
            // 초당 speed의 속도로 왼쪽으로 평행 이동
            transform.Translate(Vector3.left * speed * Time.deltaTime);
        }
    }
}
```

# 테스트하기

---

*RUNGAME이 잘 실행되는지 테스트하고 오류가 있으면 선생님께 질문해주세요!*

# 테스트하기

---

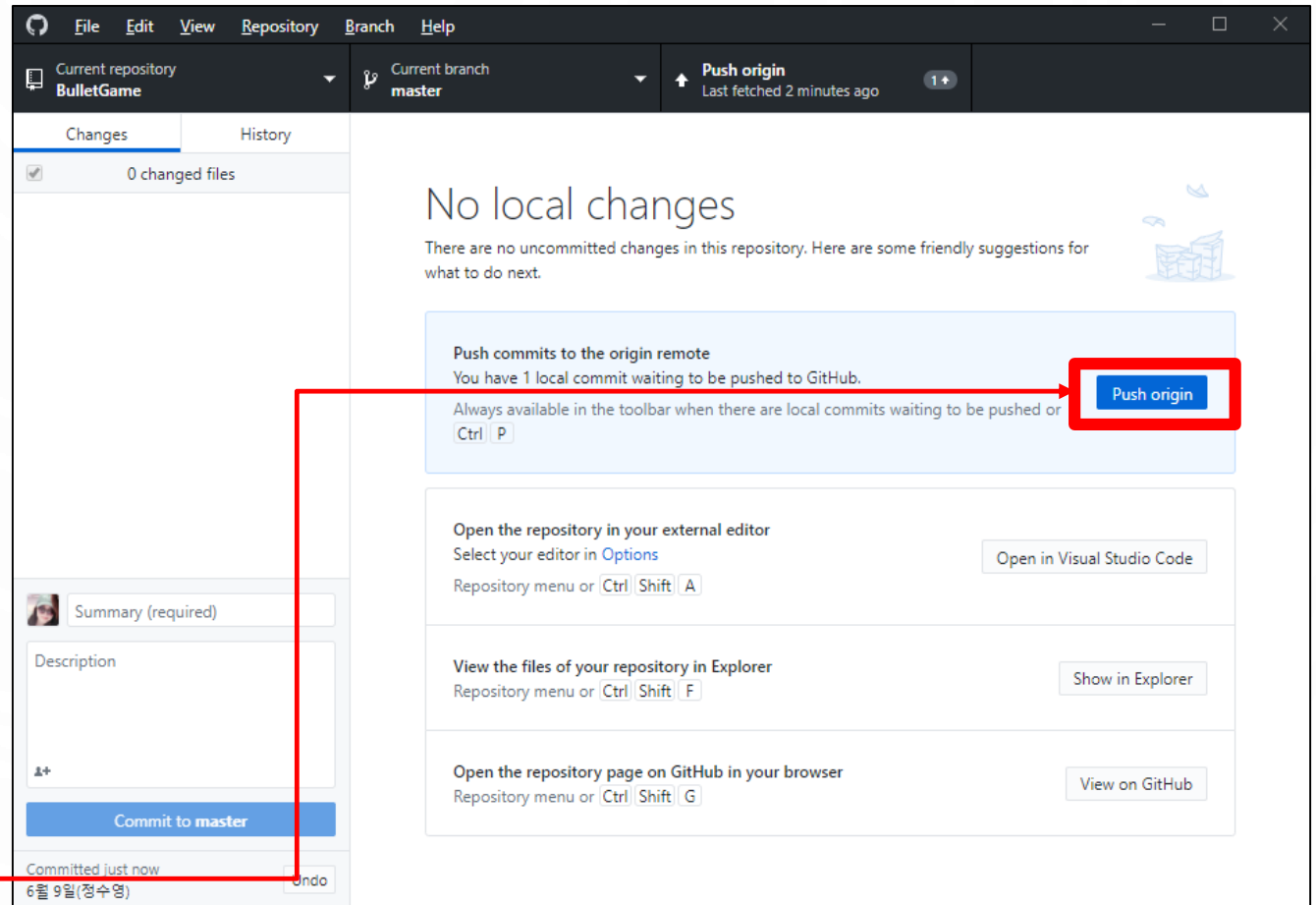
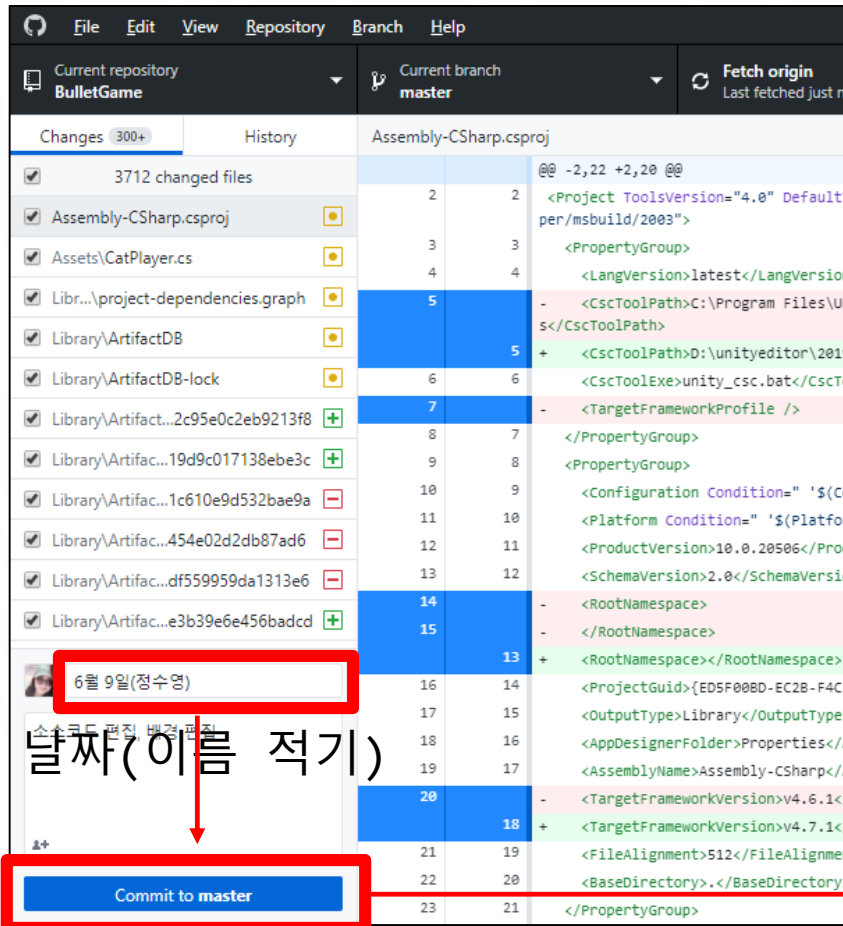


# GITHUB PUSH

---

# 깃허브 올리기

- Unity 프로젝트 저장 후 모두 닫고 Git Desktop 열기





# 에러 날 때!

The image shows the GitHub Desktop application window. The 'File' menu is highlighted with a red box, and a red arrow points from it to the 'Options' dialog box. In the 'Options' dialog, the 'Git' tab is selected and highlighted with a red box. The 'Name' field contains 'Soori' and the 'Email' field contains 'soorichu@gmail.com'. The 'Save' button is also highlighted with a red box.

Options

Accounts

Integrations

Git

Appearance

Advanced

Name

Soori

Email

soorichu@gmail.com

Save

Cancel

Let's get started!

Add a repository to GitHub Desktop to start collaborating

Clone a repository from the Internet...

Create a New Repository on your hard drive...

Add an Existing Repository from your hard drive...

그냥 Name과 Email이 잘 채워진 것만 확인하고  
Save 버튼 누르세요.  
이제 다시 Commit > Push 시도해보세요. ^^

# END

---

참고 : 레트로의 유니티 게임 프로그래밍 에센스