

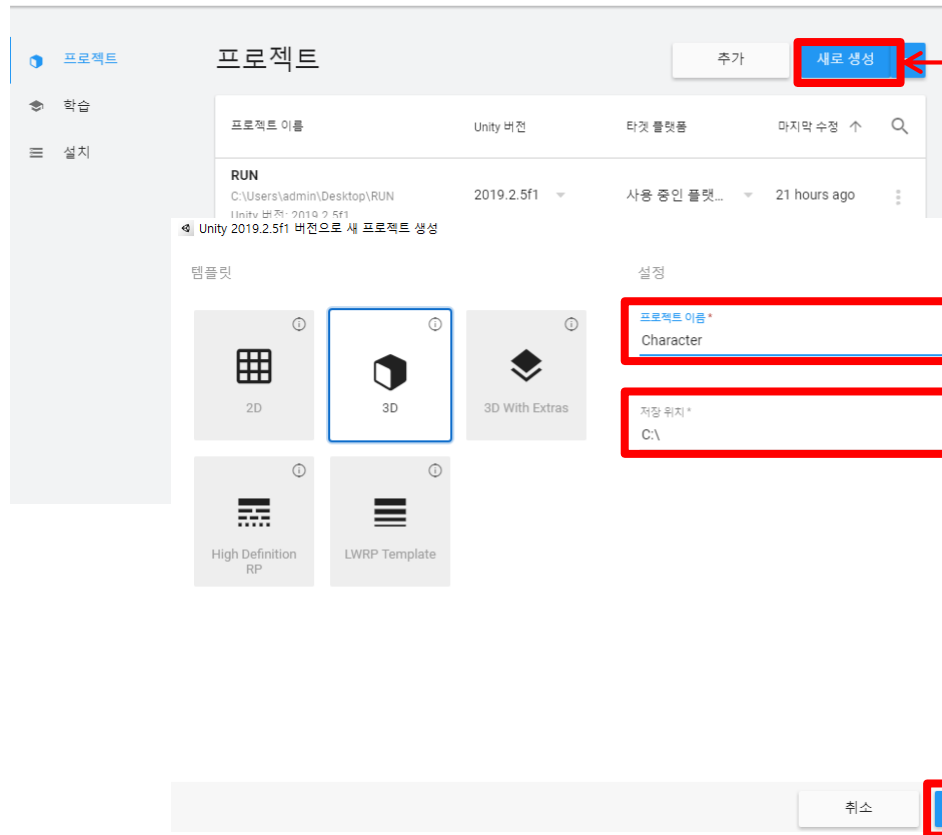
간단한 예제 3

캐릭터 움직이기

2019.2.5f1 버전에서 테스트 완료

프로젝트 만들기(Character)

Unity Hub 2.1.3



새로 생성 클릭

유니티 허브 실행 후 새로 생성 선택

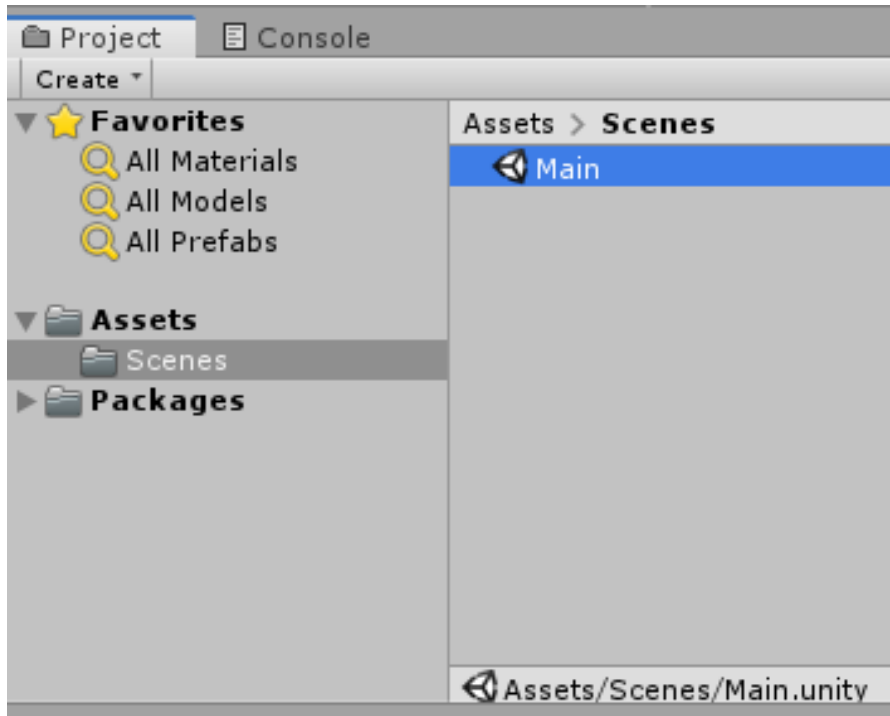
프로젝트 이름은 영문자로 설정
하길 추천합니다.

프로젝트 이름 입력하기
→ Character

저장 경로 지정하기(임의의 폴더)

생성 클릭

씬(Main)

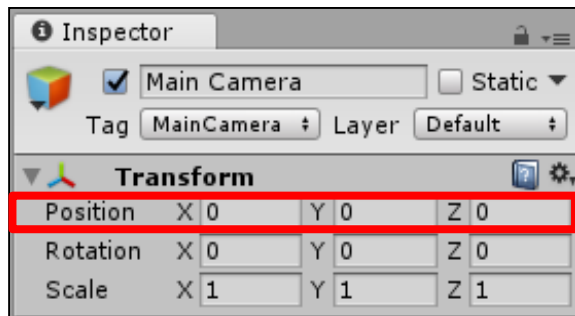
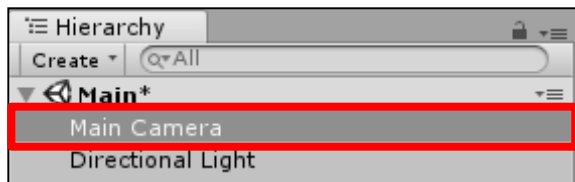


[Assets > Scenes]에 있는
SampleScene의 이름을 Main으로 변경

시점 확인하기

1. Hierarchy – Main Camera 선택

2. Inspector – Transform – Position X:0, Y:0, Z:0



3. 씬 기즈모를 이용하여 씬의 방향 변경

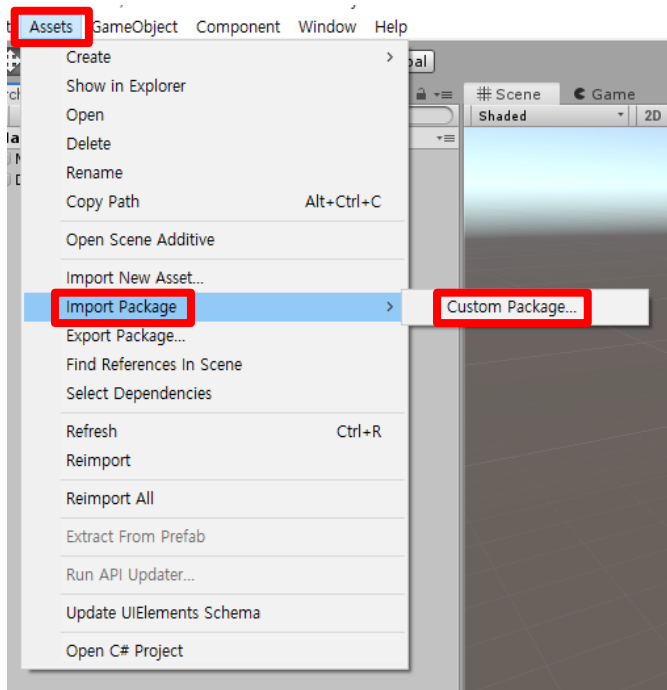


원뿔 부분을 클릭하여 씬의 방향을 회전시킴

X가 오른쪽, Y가 위쪽에 오도록 변경

게임 리소스 가져오기

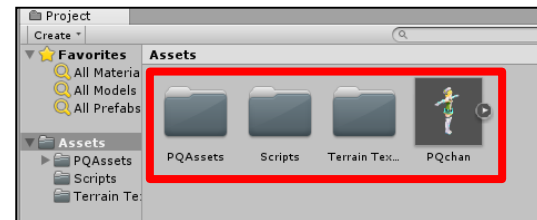
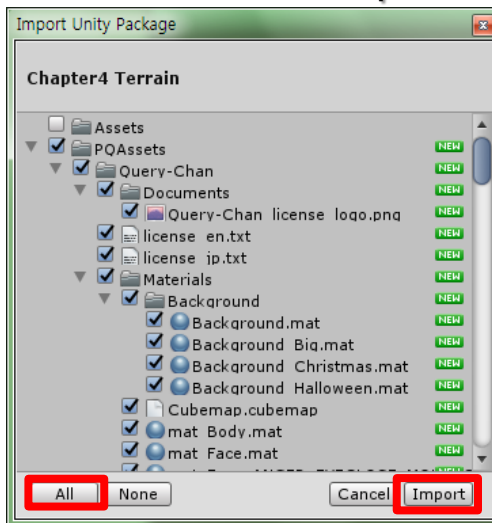
1. Assets – Import package – Custom Package...



2. Chapter4 Terrain.unitypackage 선택

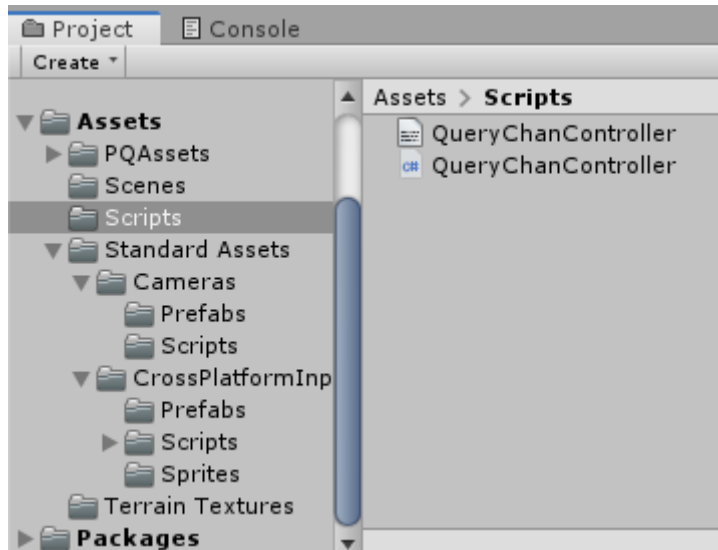
이름	수정한 날짜	유형	크기
Chapter4 Terrain.unitypackage	2019-11-13 오전 10:11	Unity package file	22,594KB
HiyokoTobashiPack.unitypackage	2019-11-13 오전 10:11	Unity package file	12,963KB

3. All 클릭 후 Import 클릭



스크립트 오류 수정하기

1. Assets > Scripts에서 Create > C# Script를 선택해 'QueryChanController'이름의 C# 스크립트 생성 후 열기



스크립트 오류 수정하기

2. 모든 코드를 지운 후에 아래의 코드를 붙여 넣고 저장

```
using UnityEngine;

public class QueryChanController : MonoBehaviour
{
    public float speed = 3.0f;
    public float jumpPower = 6.0f;
    private Vector3 direction;
    private CharacterController controller;
    private Animator anim;

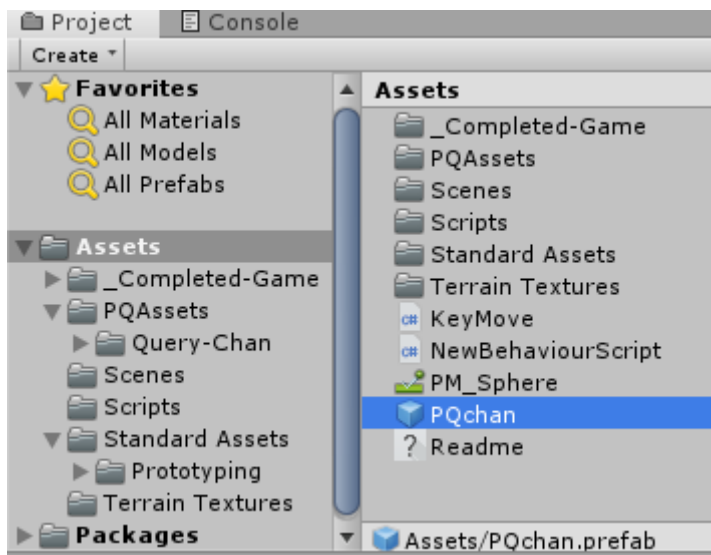
    void Start()
    {
        controller = GetComponent<CharacterController>();
        anim = GetComponent<Animator>();
    }

    void Update()
    {
        if (controller.isGrounded)
        {
            float inputX = Input.GetAxis("Horizontal");
            float inputY = Input.GetAxis("Vertical");
            Vector3 inputDirection = new Vector3(inputX, 0, inputY);
            direction = Vector3.zero;

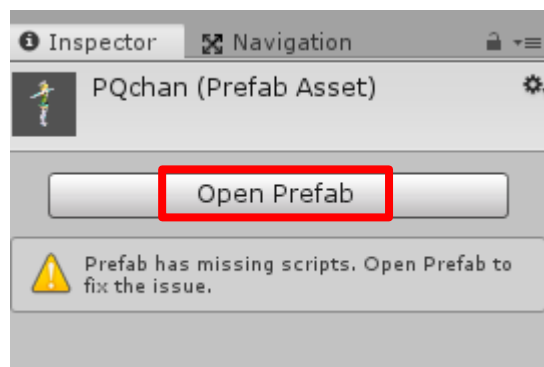
            if (inputDirection.magnitude > 0.1f)
            {
                transform.LookAt(transform.position + inputDirection);
                direction += transform.forward * speed;
                anim.SetFloat("Speed", direction.magnitude);
            }
            else
            {
                anim.SetFloat("Speed", 0);
            }
            if (Input.GetButton("Jump"))
            {
                anim.SetTrigger("Jump");
                direction.y += jumpPower;
            }
        }
        controller.Move(direction * Time.deltaTime);
        direction.y += Physics.gravity.y * Time.deltaTime;
    }
}
```

스크립트 오류 수정하기

3. Assets에 있는 PQchan 선택

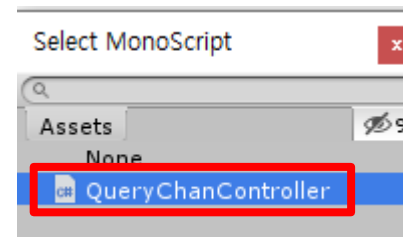
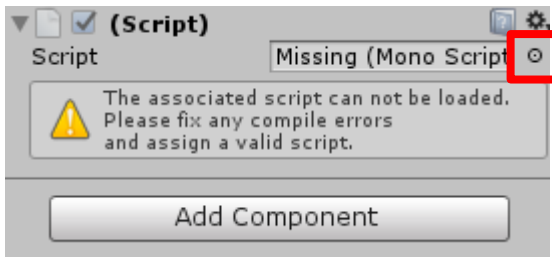


4. Inspector에서 Open Prefab 클릭



스크립트 오류 수정하기

5. Inspector창의 가장 아래 (Script)의 Script 항목
'Missing (Mono Script)' 옆에 있는 동그라미 아이콘을 클릭하고
방금 만든 C# 스크립트를 선택한다.



6. Hierarchy창의 왼쪽 화살표 버튼을 클릭해서 씬으로 되돌아간다.



스테이지 만들기 (1/4)

1. 터레인 추가하기

1.1. Hierarchy – Create – 3D Project - Terrain

2. 터레인 크기 변경하기

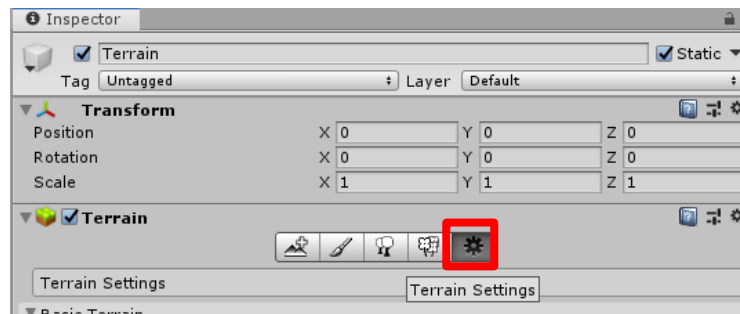
2.1. Terrain 선택 - Inspector – Terrain Settings(톱니바퀴 아이콘)

2.2. Inspector – Terrain – Mesh Resolution - Terrain Width : 100, Terrain Length : 100

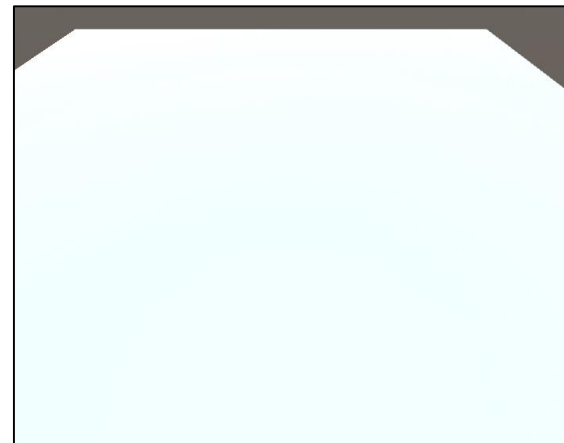
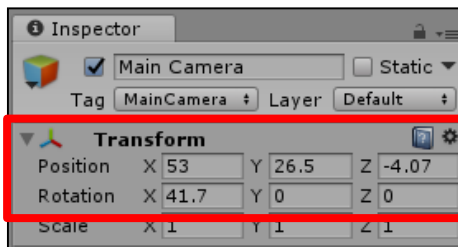
3. 카메라 방향 변경하기

3.1. Main Camera 선택 - Inspector – Transform - Position X:53, Y:26.5, Z:-4.07

3.2. Inspector – Transform - Rotation X:41.7, Y:0, Z:0



Mesh Resolution (On Terrain Data)	
Terrain Width	100
Terrain Length	100
Terrain Height	600
Detail Resolution Per Patch	32
Detail Resolution	1024

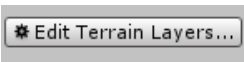


스테이지 만들기 (2/4)

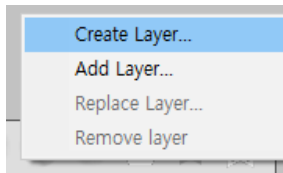
4. 터레인에 텍스처 입히기

4.1. Inspector – Terrain – Paint Terrain(붓 아이콘) –
Raise or Lower Terrain – Paint Texture

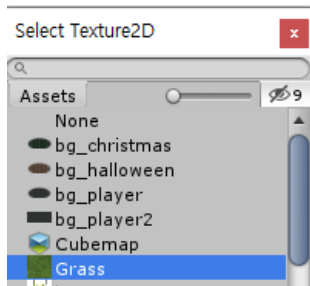
4.2. Edit Terrain Layers...



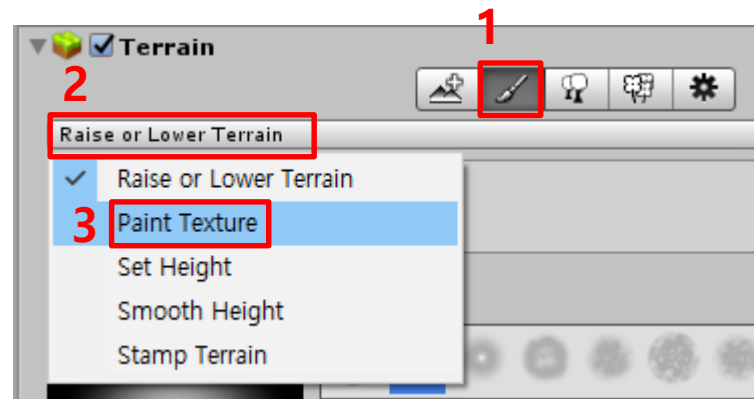
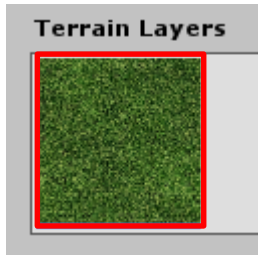
4.3. Create Layer...



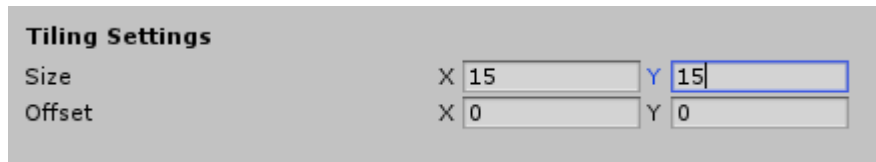
4.4. Grass 선택



4.5. Terrain Layers에서
Grass 텍스처 클릭

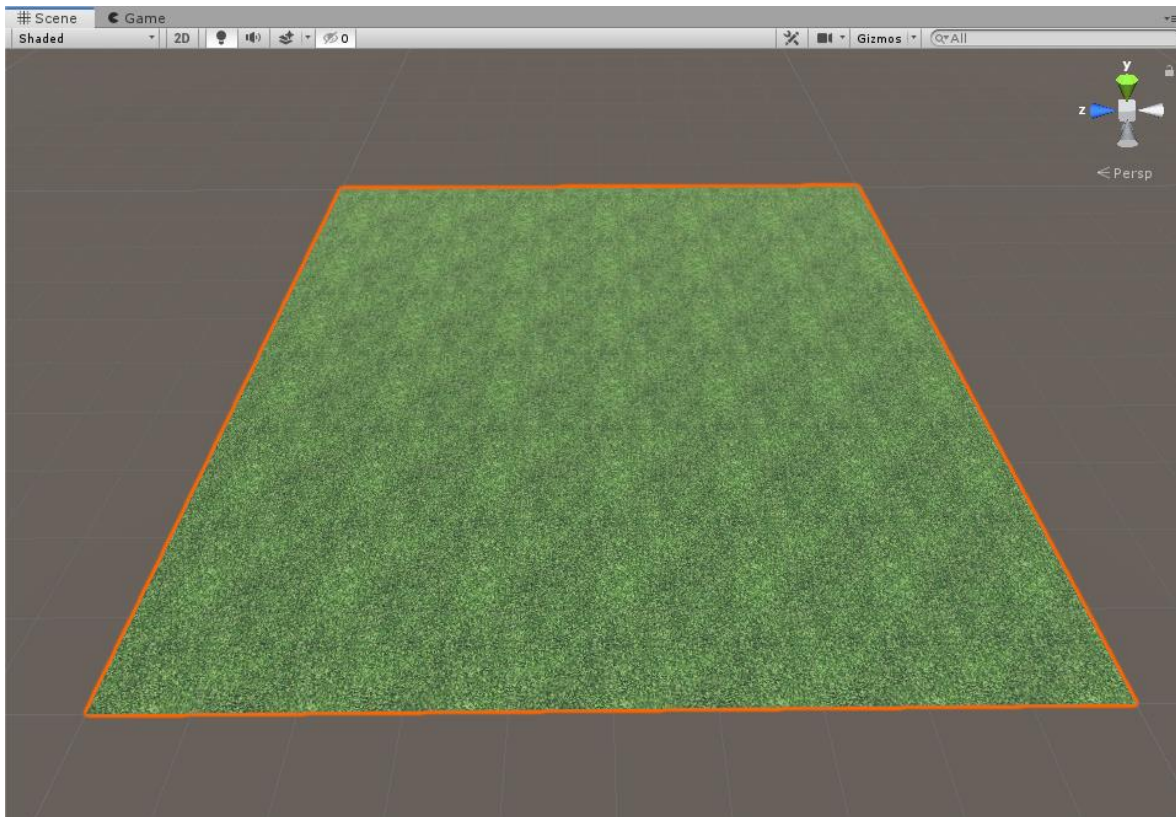


4.6. Tiling Settings에서 Size를 X:15 Y:15로 설정



스테이지 만들기 (2/4)

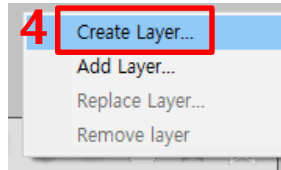
- 4.7. 키보드의 '['와 ']'키로 브러시의 크기를 조절할 수 있다.
']'키를 눌러 브러시 크기를 최대로 키운 후 터레인 전체를 칠한다.



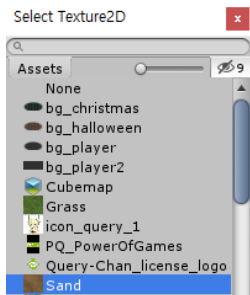
스테이지 만들기 (3/4)

5. 터레인에 흙 텍스처 추가 및 반영하기

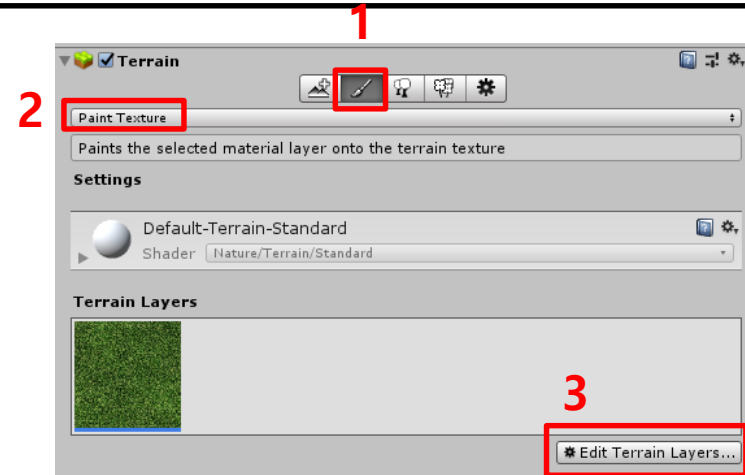
5.1. Inspector – Terrain – Paint Terrain(붓 아이콘) – Paint Texture
– Edit Terrain Layers... – Create Layer



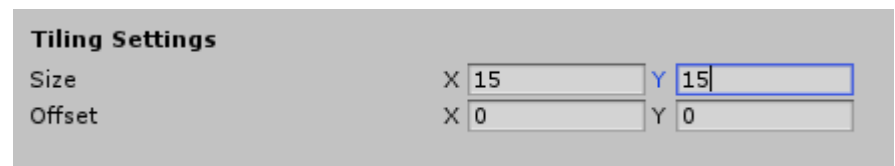
5.2. Sand 선택



5.3. Terrain Layers에서 Sand 텍스처 선택



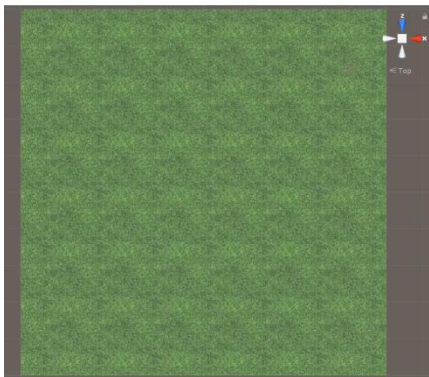
5.4. Tiling Settings에서 Size를 X:15 Y:15로 설정



스테이지 만들기 (3/4)

5.5. Hierarchy – Terrain – 시점변경

텍스처 반영을 쉽게 하기 위해 Alt+드래그, Ctrl+Alt+드래그



5.6. Inspector – Brushes에서 원하는 브러시 선택



5.7. '['와 ']'키로 브러시의 크기를 알맞게 조절하면서 터레인에 텍스처 칠하기

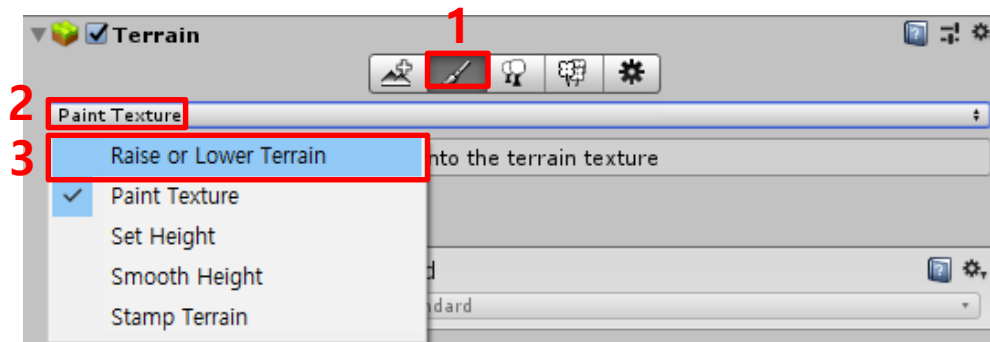
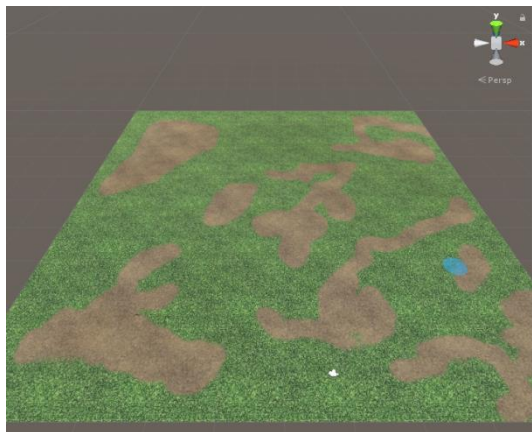


스테이지 만들기 (4/4)

6. 산 만들기

6.1. Inspector – Terrain – Paint Terrain(붓 아이콘) – Paint Texture – Raise or Lower Terrain

6.2. Alt+드래그로 시점 전환



6.3. Inspector – Brushes 선택 브러시에 따라 높아지는 모양도 달라진다.

6.4. Inspector – Opacity 선택 작게 설정하면 완만한 산을 만들 수 있다.

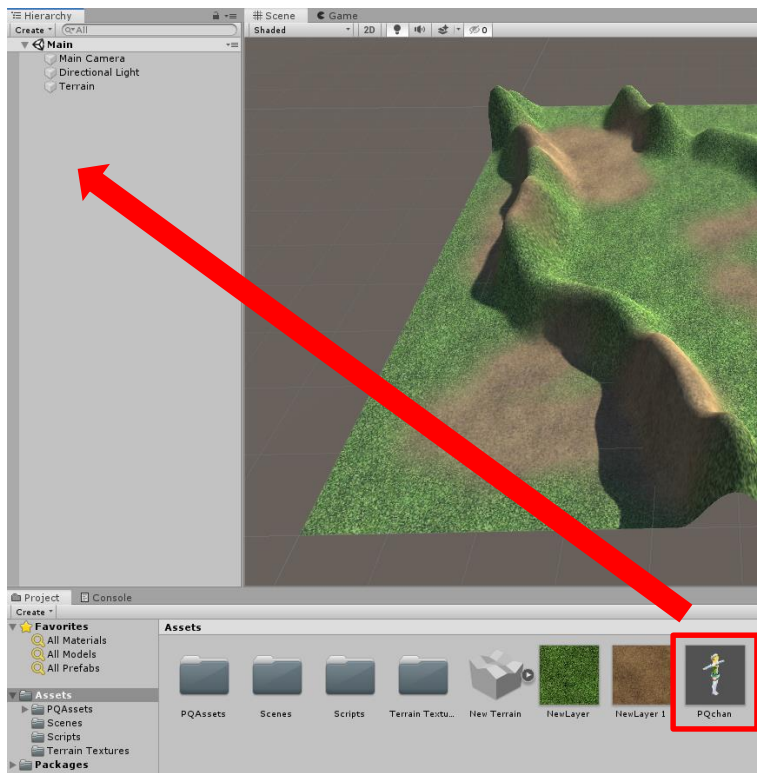
6.5. 터레인 드래그 앤 드롭



캐릭터

1. 캐릭터 씬에 배치하기

1.1. Project – Assets – PQChan을 Hierarchy로 드래그



2. 캐릭터 위치 이동하기

2.1. Hierarchy - PQChan

화살표나 사각형을 드래그하여 스테이지 중앙에 위치시킴



3. 게임 실행 및 캐릭터 움직이기



게임 실행 후 화살표 키로 캐릭터 움직이기

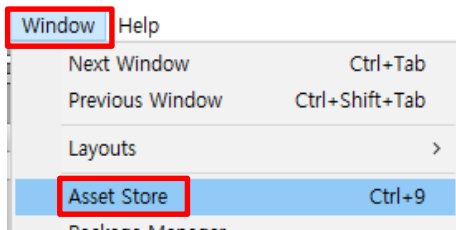
카메라 설정 (1/2)

1. Main Camera 제거하기

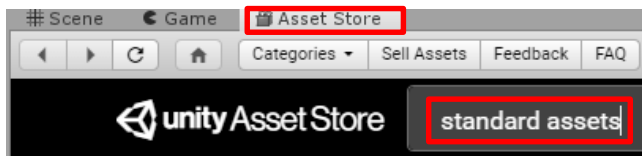
1.1. Hierarchy – Main Camera 우 클릭 - delete

2. 에셋 импорт하기

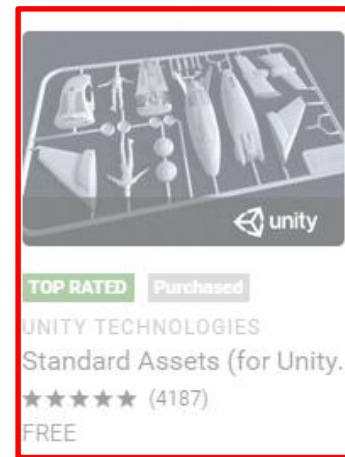
2.1. Window – Asset Store



2.2. Asset Store 탭 선택 후 'standard assets' 검색



2.3. 'Standard Assets (for Unity...' 클릭



2.4. Download를 클릭한 후 Import 클릭

**Standard Assets (for Unity
2017.3)**



Unity Technologies

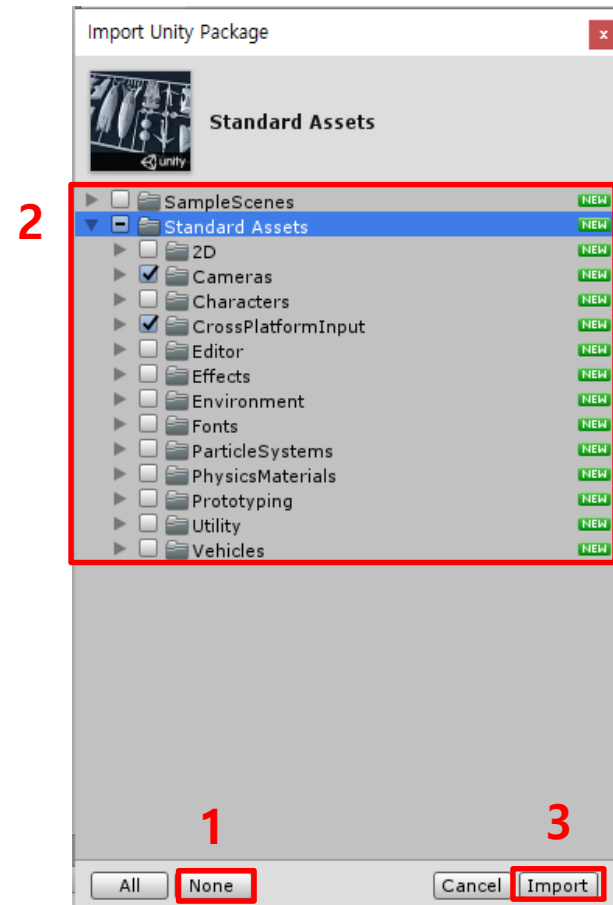
★★★★★ 5 | 709 Reviews

FREE

Import

카메라 설정 (1/2)

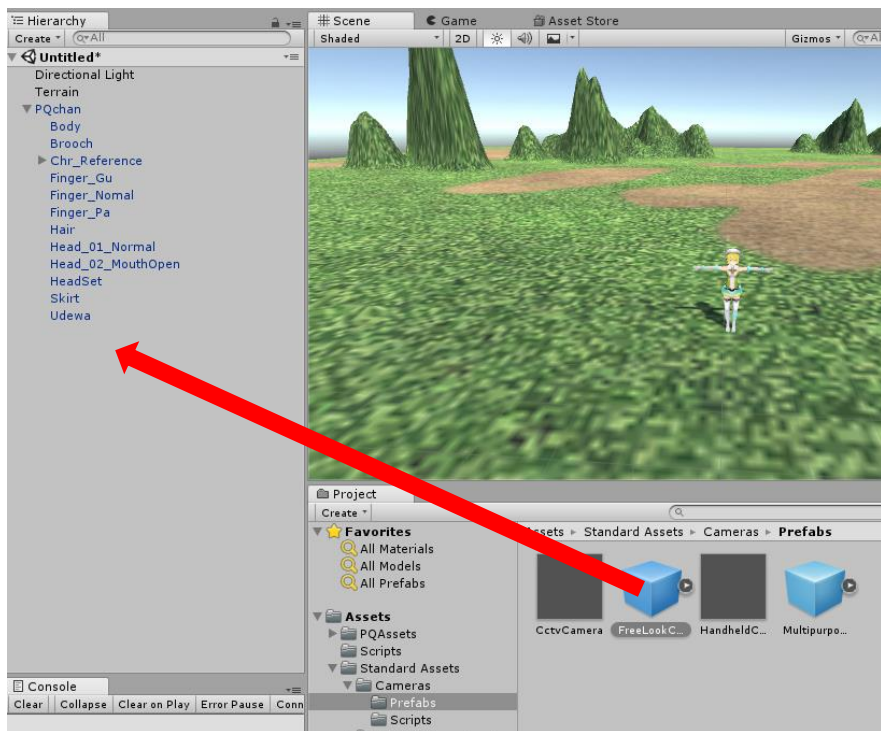
2.5. 먼저 왼쪽 아래의 None을 클릭해서 모든 체크를 푼 다음
'Cameras'와 'CrossPlatformInput'만 선택하고 Import 클릭



카메라 설정 (2/2)

3. 추적 카메라 추가하기

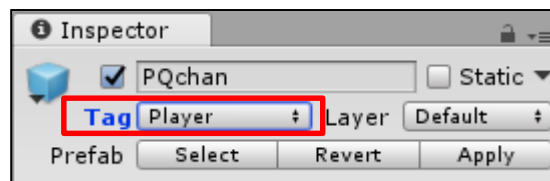
3.1. Assets – Standard Assets – Cameras – Prefabs – FreeLookCameraRig를 Hierarchy로 [드래그](#)



4. 캐릭터 따라다니도록 설정

4.1. Hierarchy - PQChan

4.2. Inspector – Tag : [Player](#) 선택



5. 게임 실행 및 캐릭터 움직이기



Reference

(초보자를 위한) 유니티 5 입문, 아라카와 다쿠야, 아사노 유이치 지음, 윤준 옮김,
한빛미디어