

Moile GunShooting 프리팹 사용법

Git Hub 주소(Scripts)

<https://github.com/rainbowkjh/UnityGame>

Scene 생성 시 필수 프리팹

1.Player, 2.MobileUICanvas, 3.GameManager, 4.MemoryPool, 5.ParsingData, 6.DialogManager, 7.StageManager, 8.LoadingManager, 9.BGMMManager

1 . Player

스크립트 역할(스크립트가 없는 오브젝트는 생략) 및 사용 옵션 설명(옵션은 빨간색)

각 컴포넌트의 연결 객체

2. MobileUICanvas

스크립트 역할(스크립트가 없는 오브젝트는 생략) 및 사용 옵션 설명(옵션은 빨간색)

각 컴포넌트의 연결 객체

3. GameManager

4. MemoryPool

5. ParsingData

6. PlayerMove

7. WeaponeCtrl

8. DialogManager

9. StageManager

10. LoadingManager

11. BMGManager

12. Enemy

1. PLAYER

Player 프리팹을 맵에 생성

(기본적으로 Player 프리팹에서 연결 되어 있는 부분은 설정되어 있다, 그 외 다른 오브젝트와 상호작용하는 부분만 연결해주면 된다. 주로 플레이와 관련된 UI 오브젝트와 연결 함)

CtrlManager

모바일 조이스틱, 마우스 조작 선택

모바일 선택 시 해당 스크립트를 활성화, 마우스를 비활성화 시키는 방식

MouseCtrl

마우스 조작

MobileCtrl

모바일 조이스틱 조작

PlayerCtrl

Hp, MaxHp – 플레이어의 체력을 설정해준다 //저장 시스템이 생기면서 DataManager에서 관리

Aim UI – 에임의 애니메이션을 호출한다

FirePosTr – 사격 시작 지점

IsDrive – 차량 또는 헬기에 탑승 시 true를 해준다(무기 교체 버튼 비활성화)

CameraRigTr – 1인칭 카메라(플레이어가 바라보는 방향으로 회전 시킴)

DeadCamera - 플레이어의 체력이 0이 되면 활성화 시키면서 쓰러지는 애니메이션을 실행

1인칭으로 카메라가 쓰러지는 애니메이션을 만들었고 그걸 실행 시킴

IsEventPlayer – 이벤트 연출을 할 때 조작 및 체력 등 상황에 맞게 제어를 하기 위한 옵션

MarkerCamTr – 미니맵에 마커를 보여줌(카메라)

각 캐릭터들 밑에 마커를 만들었고, 플레이어 밑에 마커를 바라보도록 만든 카메라

AniCtrl

애니메이션 실행 함수를 관리 한다

상황에 맞게 필요한 애니메이션을 호출 시켜 사용

WeaponManager

무기 장착 위치를 관리 한다

사용 무기를 활성화 시키고 미 사용 무기를 비활성화 시킨다

PlayerUI

게임에 관련되 UI 를 호출하여 출력한다

HitDmg

공격을 받으면 HP 를 감소

플레이어의 경우(적에게 공격 당하면) 카메라를 흔들어 타격 효과를 준다

VoicePlay

캐릭터의 음성을 관리 – 플레이어는 아직 음성 파일이 없다

FadeOut

플레이어의 HP 가 0 이 되면 화면을 어둡게 한다

화면을 천천히 밝게하는 함수도 있다

ItemManager

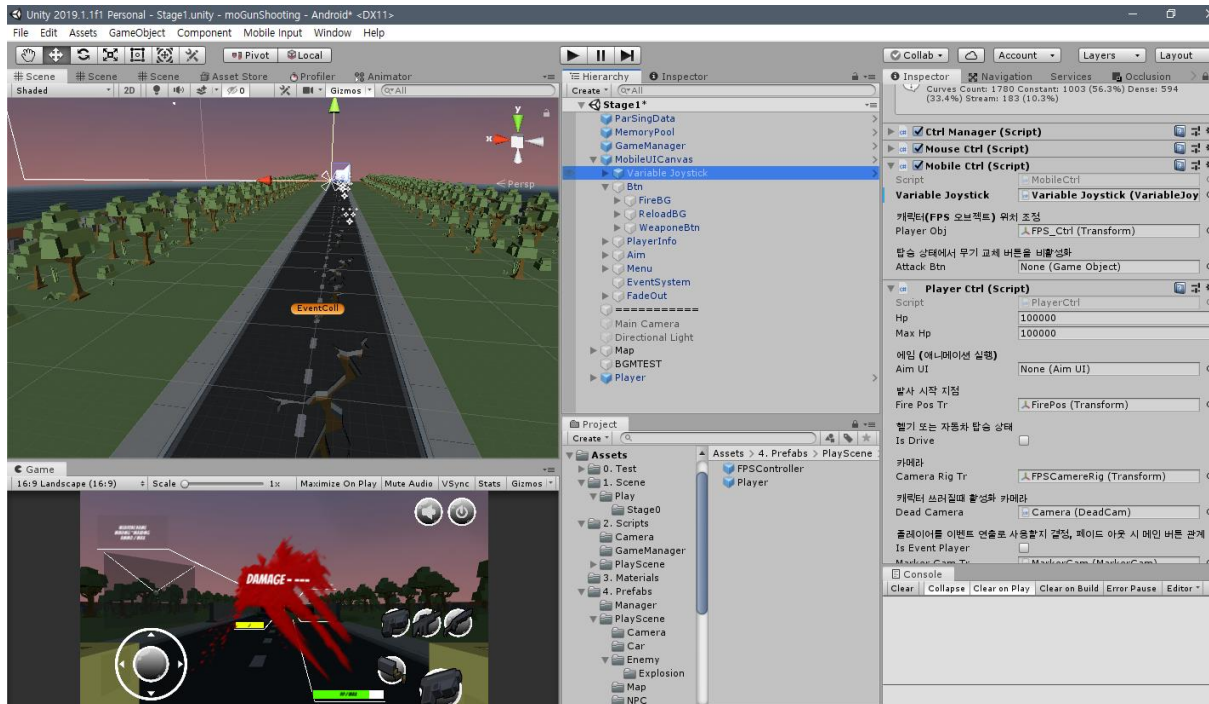
수류탄과 회복 아이템 정보를 관리 한다

GrenadeDmg 현재 수류탄 데미지,

RecoveryHP 현재 회복량

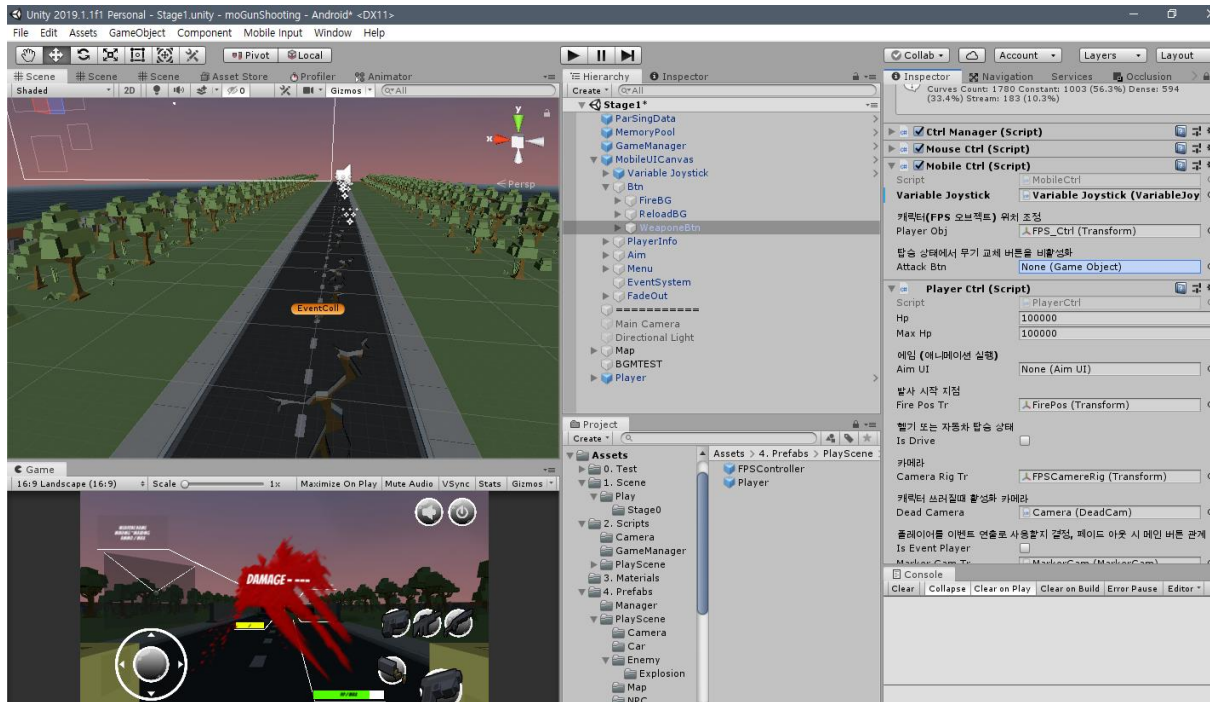
nUPgradePoint 현재 업그레이드 포인트

MobileCtrl



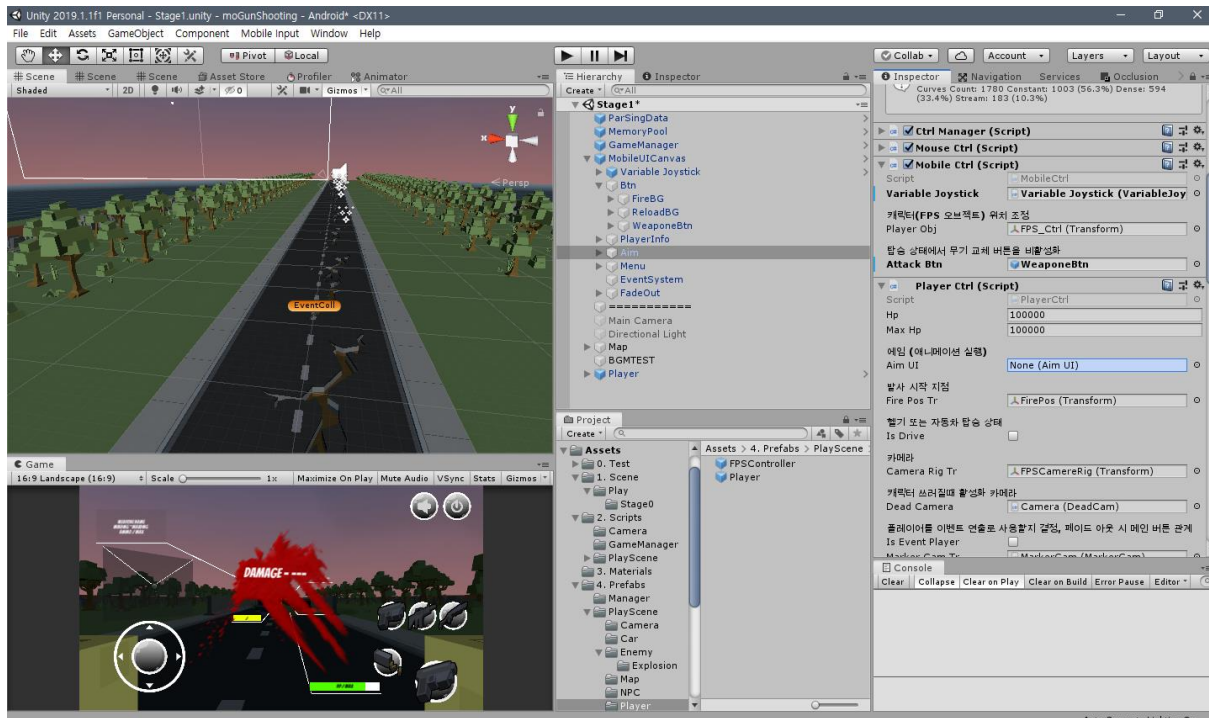
모바일 조이스틱 설정

- 캐릭터의 시야를 조작 한다
- Variable Joystick – MobileUICanvas -> Variable Joystick



차량 탑승 시 무기 교체 버튼을 비활성화 시킨다.(탑승 무기만 사용)

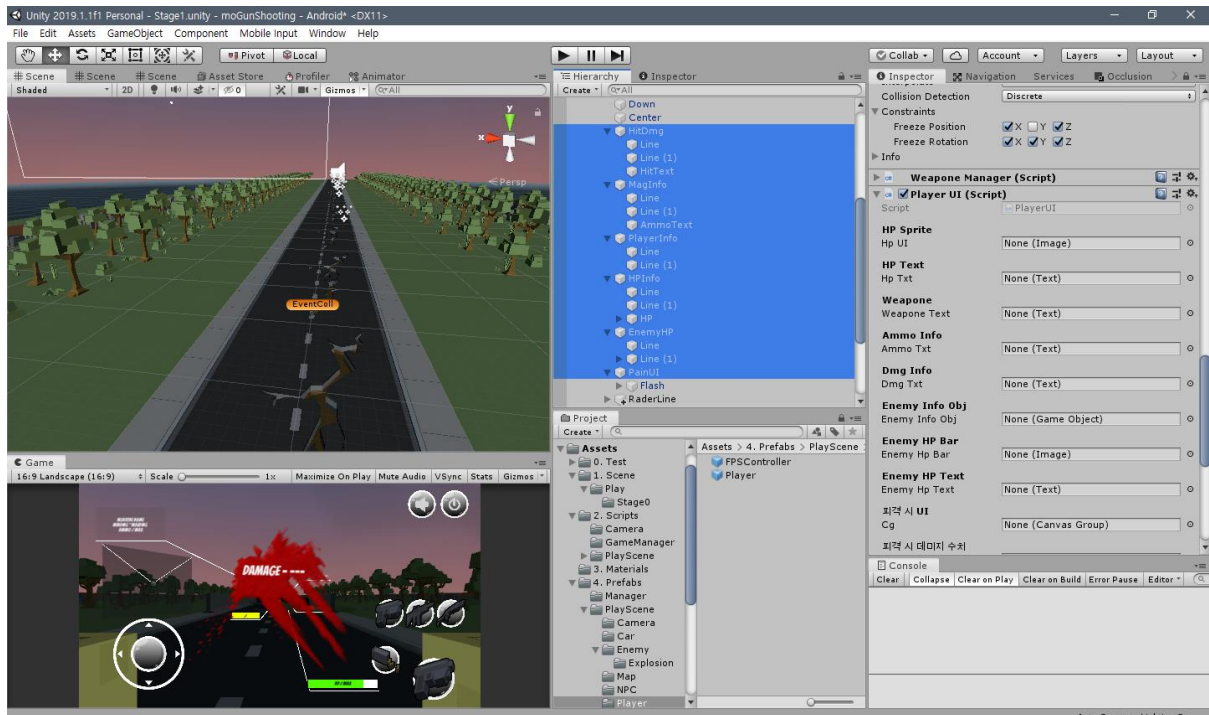
Attack Btn – MobileUICanvas -> Btn -> WeaponeBtn



PlayerCtrl

플레이어의 조준 애니메이션

- 대기, 사격 시 에임 벌어짐 효과(이동 중 숨겨지는 효과 추가 예정)
- Aim UI – MobileUICanvas -> Aim



PlayerUI

게임에 필요한 정보를 출력해 주는 UI 설정

Hp UI – MobileUICanvas(오브젝트) -> Aim -> HPInfo -> HP ->HPBar

HP Txt - ... HP -> HPInfo

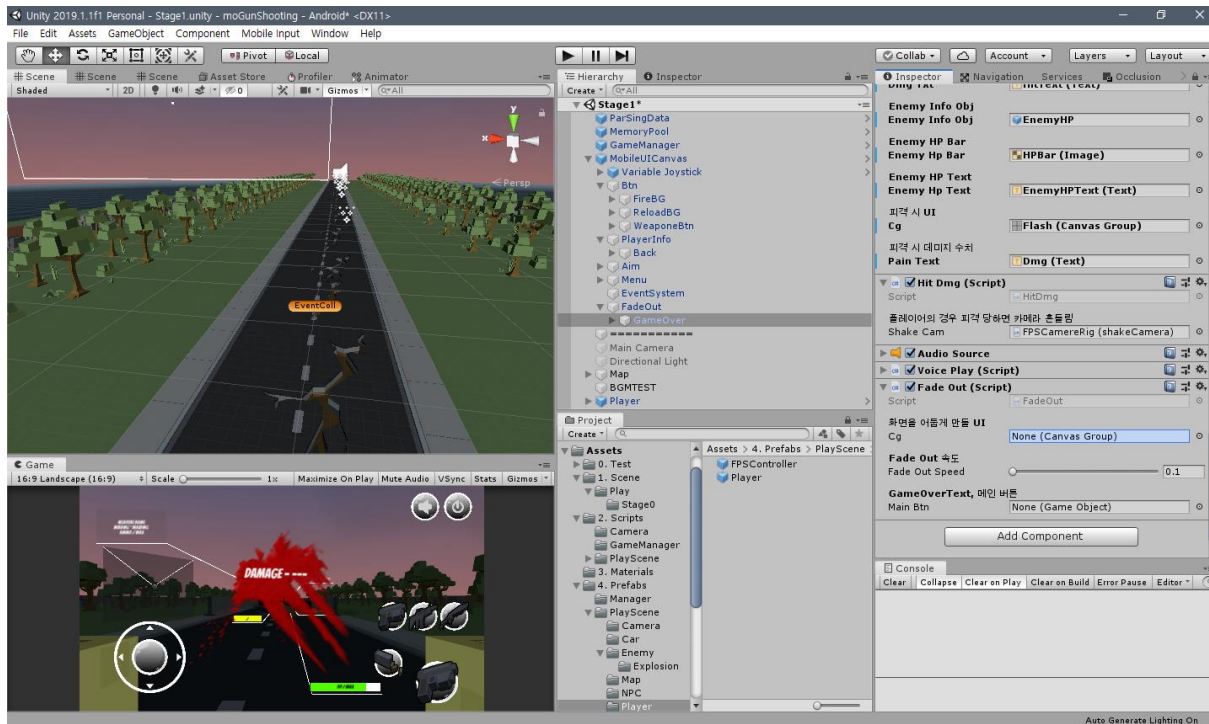
Weapon Text (무기 정보 출력)

Weapon Text – MobileUICanvas -> PlayerInfo -> Back -> WeaponInfo -> WeaponText

피격 시 UI

cg - MobileUICanvas(오브젝트) -> Aim ->. PainUI->Flash

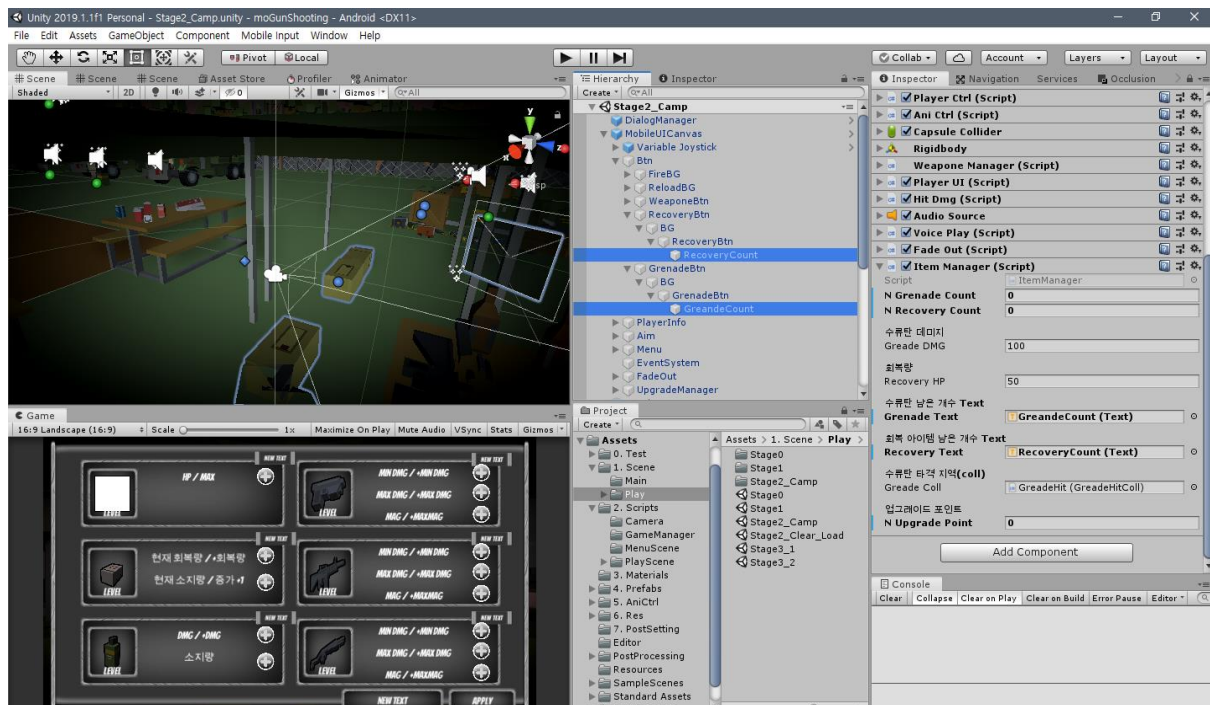
PainText - ...PainUI -> Dmg



Fade Out

- 플레이어가 쓰러지면 화면을 어둡게 한다
- 이벤트로 쓰러질때는 화면만 어두워지고
- 실제 플레이 중 HP가 0이 되면 어두워진 후 메인으로 가는 버튼 활성화
- Cg – MobileUICanvas -> FadeOut -> GameOver
- MainBtn - ...GameOver -> MainBtn

ItemManager



아이템의 정보를 관리

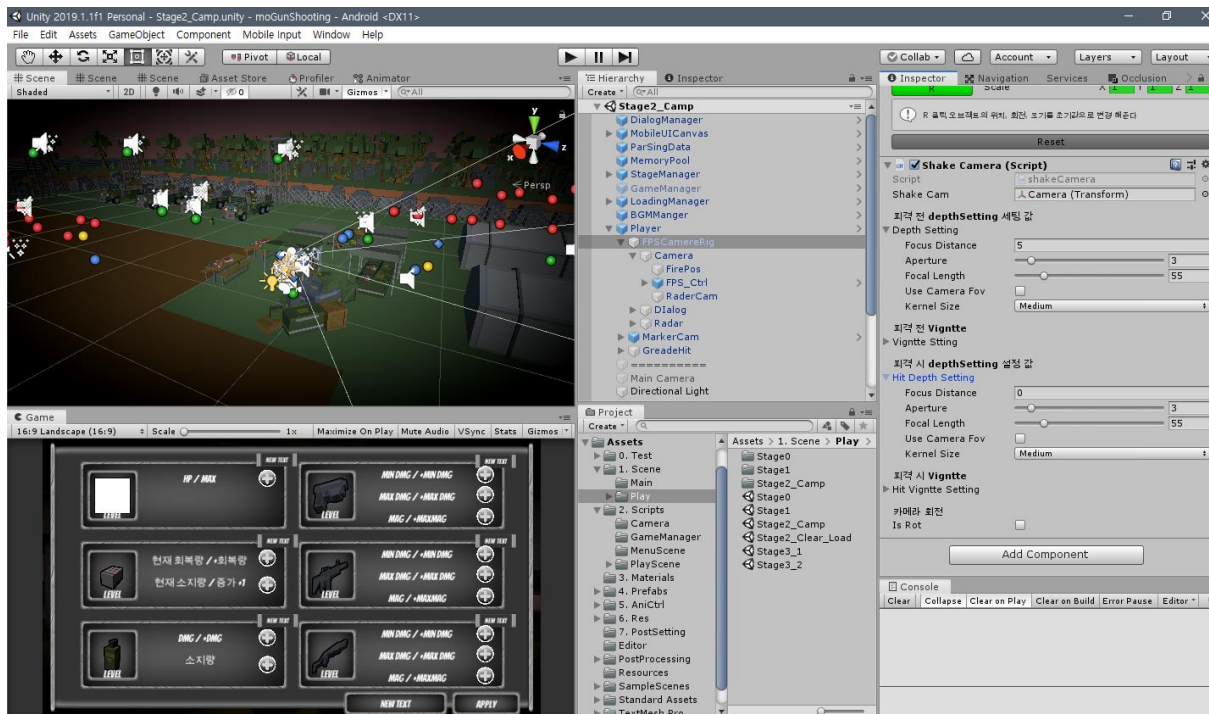
수류탄 데미지, 회복 아이템 회복량, 남은 개수 확인 UI, 업그레이드 포인트, 수류탄 데미지 범위(콜라이더)

- 아이템 남은 수량은 UI 오브젝트에서 가져온다
- 수류탄 타격 지역은 플레이어 하위 오브젝트에 있는 콜라이더를 가져온다.

ShakeCam

피격 등 타격 효과를 주기 위해 카메라의 상위 오브젝트에 적용 시킨다(CameraRig)

추가로 포스트 프로세싱으로 타격 효과를 줌



피격 전 세팅 값과 피격 되었을 때 세팅 값

Depth Setting

피격 시 Depth Setting

Focus Distance 5

Focus Distance 0

Aperture 3

Aperture 3

Focal Length 55

Focal Length 55

Vignette Setting

Vignette Setting

Mode Classic Color Black

Smoothness 1

Center x 0.5, y 0.5 Intensity 0.45

Smoothness 0.01 Roundness 1

2. MobileUICanvas(게임의 정보를 보여주는 UI, 하위 오브젝트에 각 기능의 스크립트가 존재한다)

스크립트가 있는 것만 정리

나머지는 각 역할을 하는 오브젝트로 다른 곳에서 함수로 제어하여 사용한다

Variable Joystick

Variable Joystick(OBJ) - **Variable Joystick(Scripts)**

모바일 조이스틱 (유니티 에셋)

Btn(OBJ)

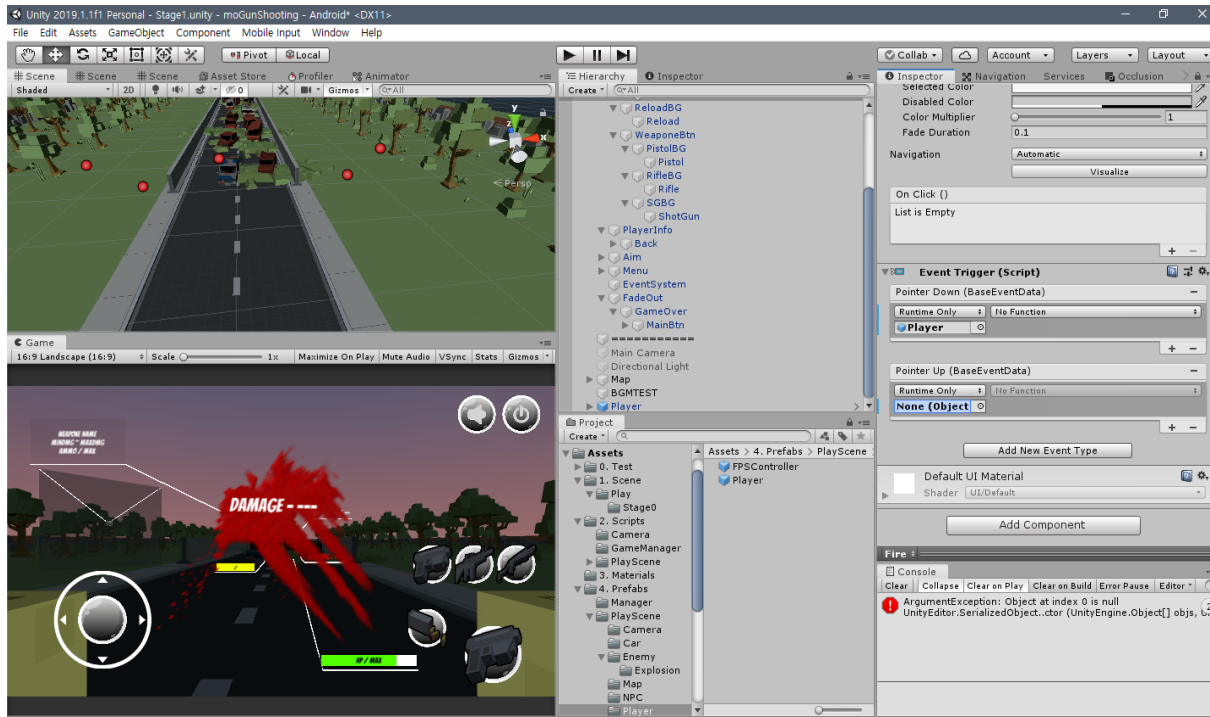
무기 교체, 공격, 재장전 등 버튼(스킬?? 또는 수류탄 등 추가 기능 추가 가능)

이벤트 트리거로 함수 호출(이벤트 트리거가 없는건 버튼 컴포넌트로 하면 된다)

Menu – **MenuBtnManager(Scripts)**

우측 상단에 메뉴 버튼

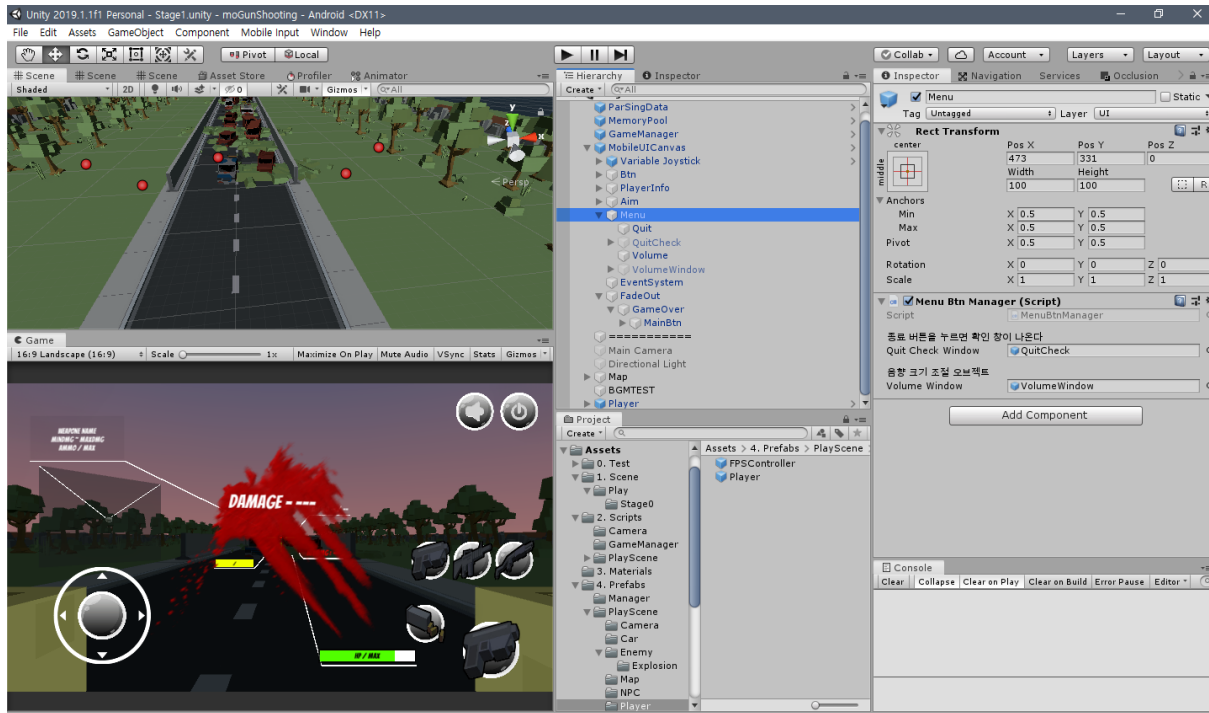
Btn(공격, 재장전, 무기 교체) – 공격만 예로 작성



누르는 즉시 반응하도록 버튼 컴포넌트가 아닌 이벤트 트리거에서 호출함

캐릭터의 모바일 조작에 있는 공격 함수를 가져오기 때문에 Player 오브젝트를 연결 시킴

MenuBtnManager



QuitCheckWindow – Menu(OBJ) -> QuitCheck

VolumeWindow – Menu -> VolumeWindow

3. GameManager

.GameManager

IsPause – 게임의 정지 상태(모든 캐릭터가 정지)

Volume – 게임의 배경음, 효과음 볼륨을 관리 한다

DataManger

게임의 저장, 로드 함수를 관리(따로 설정 해주는 것은 없다)

4. MemoryPool

.MemoryPool

플링을 사용할 오브젝트를 관리

5. ParsingData

무기의 데이터를 파싱하여 리스트를 만든다

각 무기에서 이름으로 검색하여 파일의 데이터를 적용 시킨다

6. PlayerMove

MovePos(Scripts)

이동 시킬 Pos 오브젝트를 만들고 스크립트를 적용시킨다

NextPos – 다음 위치

IsPlayerPos – 이동 좌표가 플레이어의 좌표인지 적의 좌표인지 구분(에디트 기능에 해당 – 플레이어는 파란색 상자, 적은 빨간색 상자로 에디터에 표시 되며, 다음 위치까지 녹색 선으로 가리켜 준다)

IsEvent – 전투 발생(true 이면 적을 생성 시키는 스크립트를 호출한다 (생성 시킬 적의 좌표를 묶어놓은 오브젝트에 있는 스크립트)

MoveSpeed – 다음 위치까지 이동속도를 변경 시켜준다

EnemyArea - 생성 시킬 적의 좌표를 묶어놓은 오브젝트

TagName – 플레이어 외 다른 객체의 감지를 한다

MoveTimer – 플레이어가 접근하면 해당 시간 만큼 대기 후 이동한다 (IsEvent 와 같이 사용)

EndPos(Scripts)

MovePos 는 다음 위치를 넣어 주고 이동 시키는 역할이면

EndPos 는 다음 위치가 없고 도착하면 정지 시킨다

IsPlayerPos – 위와 같은 기능

TagName – 위와 같은 기능

7. WeaponeCtrl

각 무기에 적용 시킨다

효과음, 파티클 등 관리

WeaponeCtrl

WeaponeDataName - 파싱 데이터의 리스트에서 해당 무기의 이름을 검색하여 데이터를 적용 시킨다

Sfx - 무기의 효과음 배열

MuzzleEffect - 사격 이펙트

HitEffectObj - 피격 지점 파티클

(오브젝트 안에 파티클들이 있어 오브젝트로 호출하여 파티클을 배열에 담아 실행 시킨다)

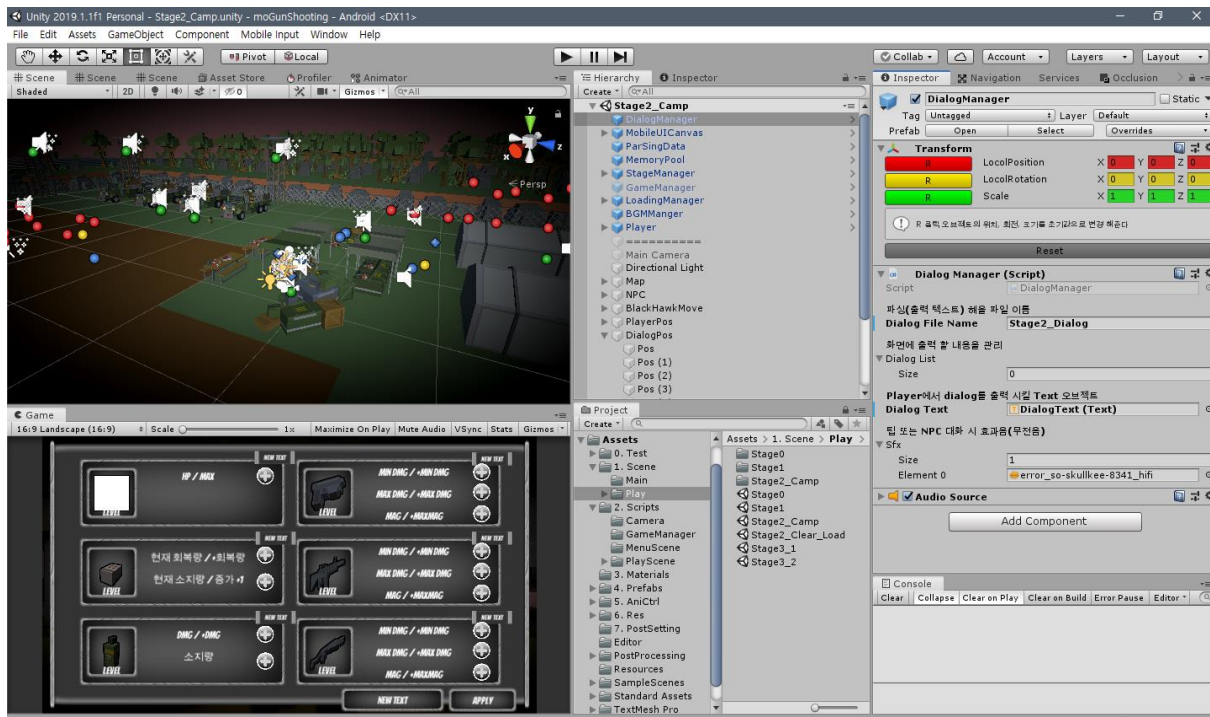
ShakeCam - 사격 시 카메라 흔들림 효과

8.DialogManager

대화창의 내용을 관리한다

XML 로 파일을 만들고 파싱하여 데이터를 담고 있다가

DialogPos(CallDialog)에서 플레이어를 감지하면 해당 인덱스 값을 넘겨 출력 시킨다.



DialogFileName – XML 파일 이름을 적어준다(해당 파일의 내용을 파싱함)

DialogList – 파싱 된 데이터를 리스트에 담음

Dialog Text – Player – FPSCameraRig – Dialog – Canvas – BG – DialogText

Sfx – 텍스트 출력 효과음

CallDialog(DialogPos)

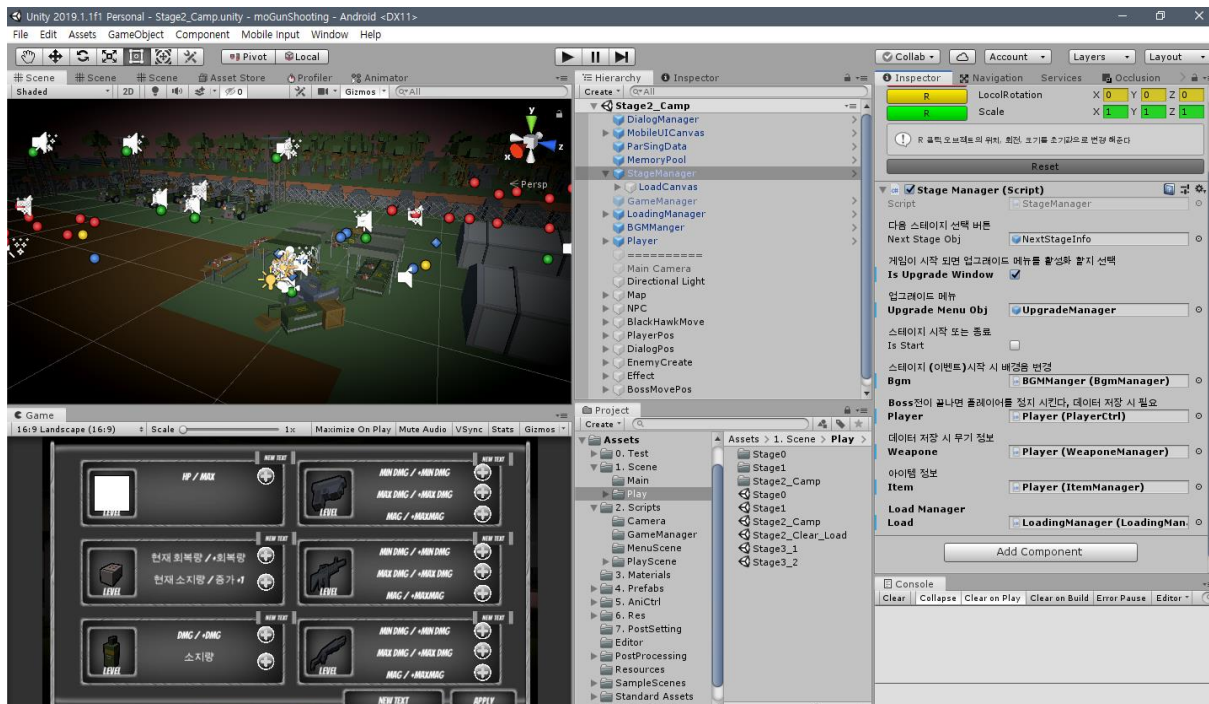
DialogIndex – 텍스트를 출력 시킬 위치에 콜라이더를 생성하고 출력 시킬 리스트의 문자열의 인덱스 값을 지정해준다

DialogManager – DialogManager 를 연결해주면 된다

9. StageManager

스테이지 관리

스테이지 별로 업그레이드가 가능한 스테이지인지, 보스 캐릭터가 있는지에 따라 옵션을 선택 해주어야 한다. Stage1 처럼 둘다 해당하지 않을때는 둘다 선택하지 않으면 된다



NextStageObj – StageManager –LoadCanvas – NextStageInfo(클리어 시 다음 스테이지 선택 버튼)

IsUpgradeWindow – 스테이지 시작 시 업그레이드 창이 활성화 된다

UpgradeMenuObj – 업그레이드 활성화 창 /UI 에 있는 UpgradeManager

IsStart – 보스전 제어(보스 처치 시 false 가 되면서 다음 스테이지 버튼이 활성화

(업그레이드 창을 닫을 때 전투 이벤트를 바로 발생 시킬때 사용 하기도한다)

NPC 들이 공격을 멈춘다

Bgm – BGMMManager

Player, Weapone, Item – Player

Load - LoadingManager