



게임엔진프로그래밍응용

14. 네트워크 협동 게임 3

청강문화산업대학교 게임콘텐츠스쿨

반 경 진

공지사항

공지사항

- 2023학년도 1학기 방역 및 학사운영 방안
<https://www.ck.ac.kr/archives/193175>
- 2023학년도 1학기 국가공휴일 및 대학 행사 수업 대체 일정 공지
<https://www.ck.ac.kr/archives/193109>

온라인 수업 저작권 유의 사항

온라인 수업 저작권 유의 사항

온라인수업 저작권 유의사항 안내



**강의 저작물을 다운로드, 캡처하여
교외로 유출하는 행위는
불 법 입 니 다**

저작권자의 허락 없이 저작물을 복제, 공중송신 또는 배포하는 것은
저작권 침해에 해당하며 저작권법에 처벌받을 수 있습니다.

강의 동영상과 자료 일체는 교수 및 학교의 저작물로서 저작권이 보호됩니다.
수업자료를 무단 복제 또는 배포, 전송 시 민형사상 책임을 질 수 있습니다.

Index

1	안드로이드 빌드
2	IOS 빌드
3	기말 과제
4	Q & A

안드로이드 빌드

안드로이드 빌드

- 안드로이드 빌드
 - 안드로이드 장치의 개발자 모드 활성화
 - 안드로이드 SDK 준비
 - 유니티와 안드로이드 SDK 연동
 - 안드로이드 빌드 설정
 - 빌드하기

안드로이드 빌드

- 안드로이드 장치의 개발자 모드 활성화
OS버전이나 제조사에 따라 다를 수 있음

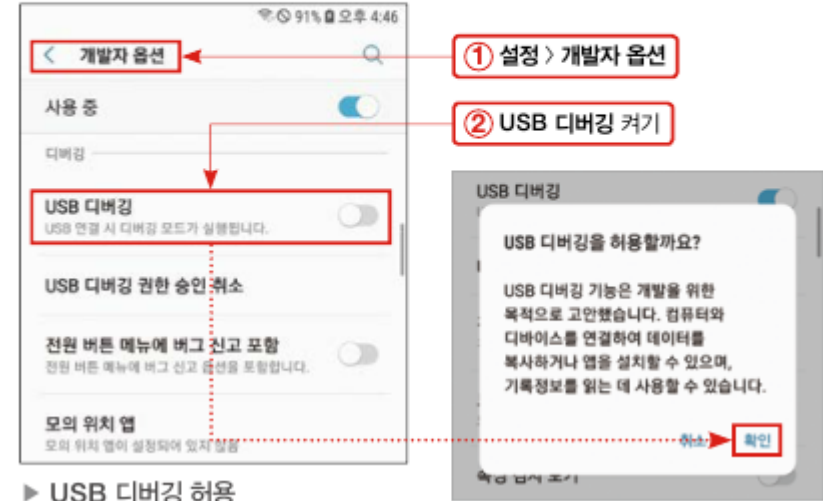
[과정 I] 개발자 모드 활성화

- ① 안드로이드 기기의 설정 > 휴대전화 정보 > 소프트웨어 정보로 이동
- ② 빌드 번호 여러 번 터치



[과정 II] USB 디버깅 활성화

- ① 안드로이드 기기의 설정 메뉴 > 개발자 옵션
- ② 개발자 옵션에서 USB 디버깅 활성화



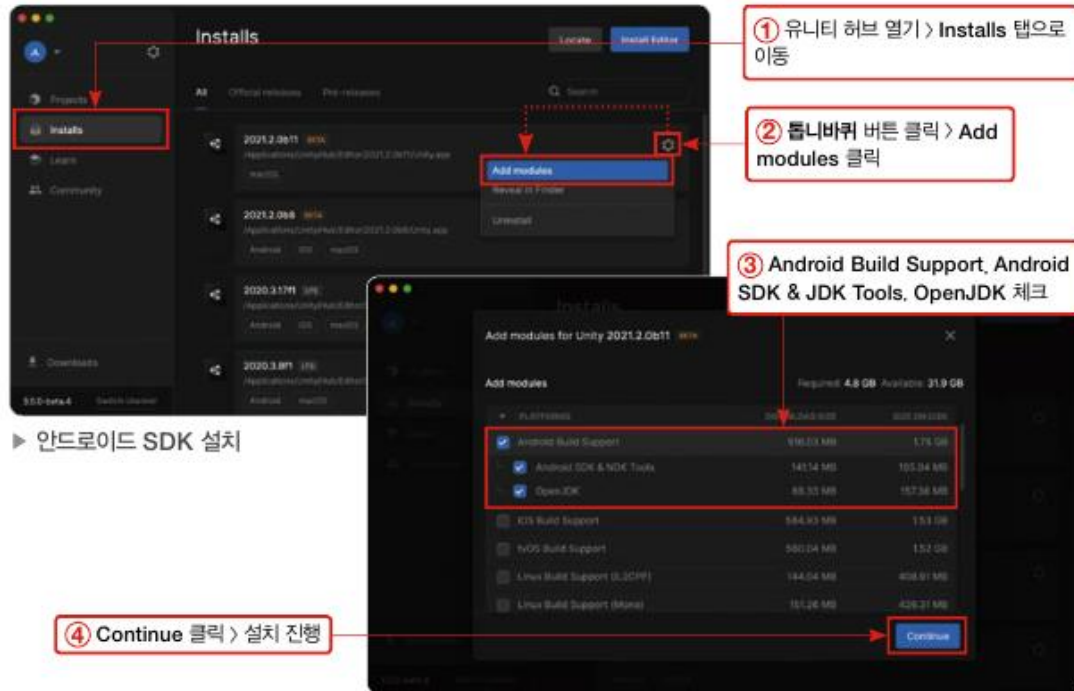
안드로이드 빌드

• 안드로이드 SDK 준비

유니티 허브를 통해 모듈 추가 또는 안드로이드 스튜디오 설치

과정 이 추가모듈 설치하기

- ① 유니티 허브 열기 > Installs 탭으로 이동
- ② 유니티 에디터 항목 옆의 톱니바퀴 버튼 클릭 > Add modules 클릭
- ③ Android Build Support, Android SDK & JDK Tools, OpenJDK 체크
- ④ Continue 클릭 > (SDK 약관 안내 등이 나오면 동의 등을 선택) 설치 진행

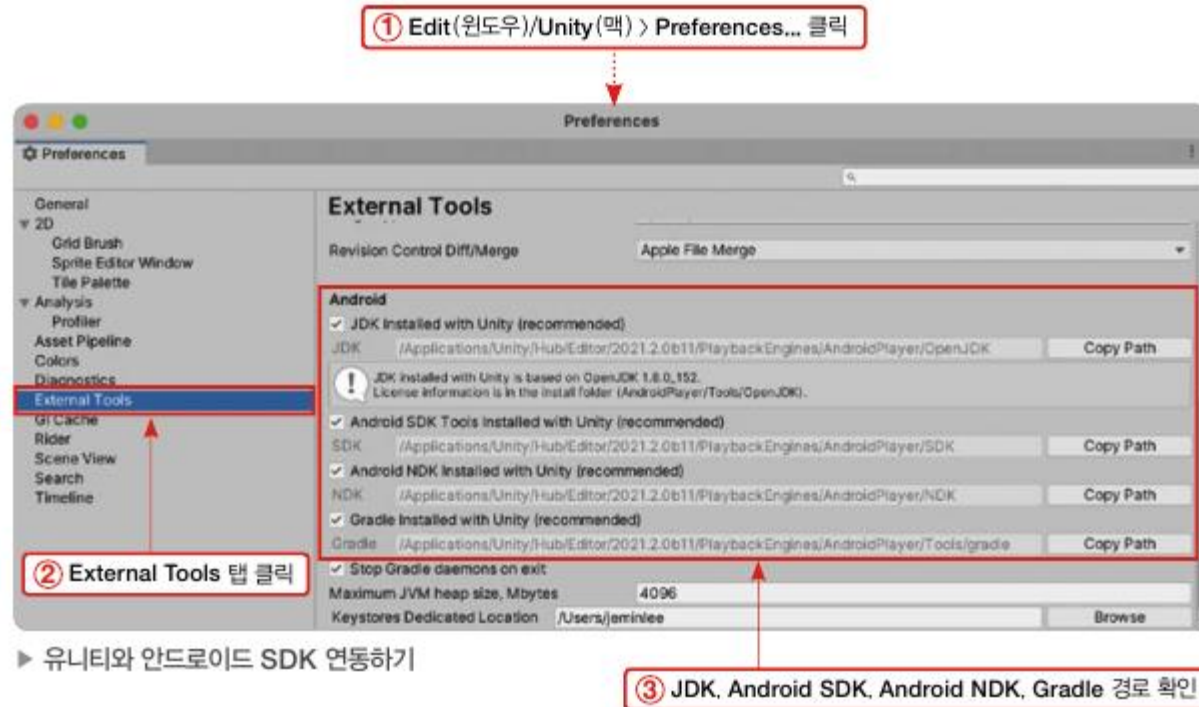


안드로이드 빌드

• 유니티와 안드로이드 SDK 연동

[과정 I] 유니티와 안드로이드 SDK 연동 확인

- ① 유니티 Preferences 창 띄우기(Edit(윈도우)/Unity(맥) > Preferences... 클릭)
- ② External Tools 탭 클릭
- ③ Android 항목에서 JDK, Android SDK, Android NDK, Gradle 경로 확인

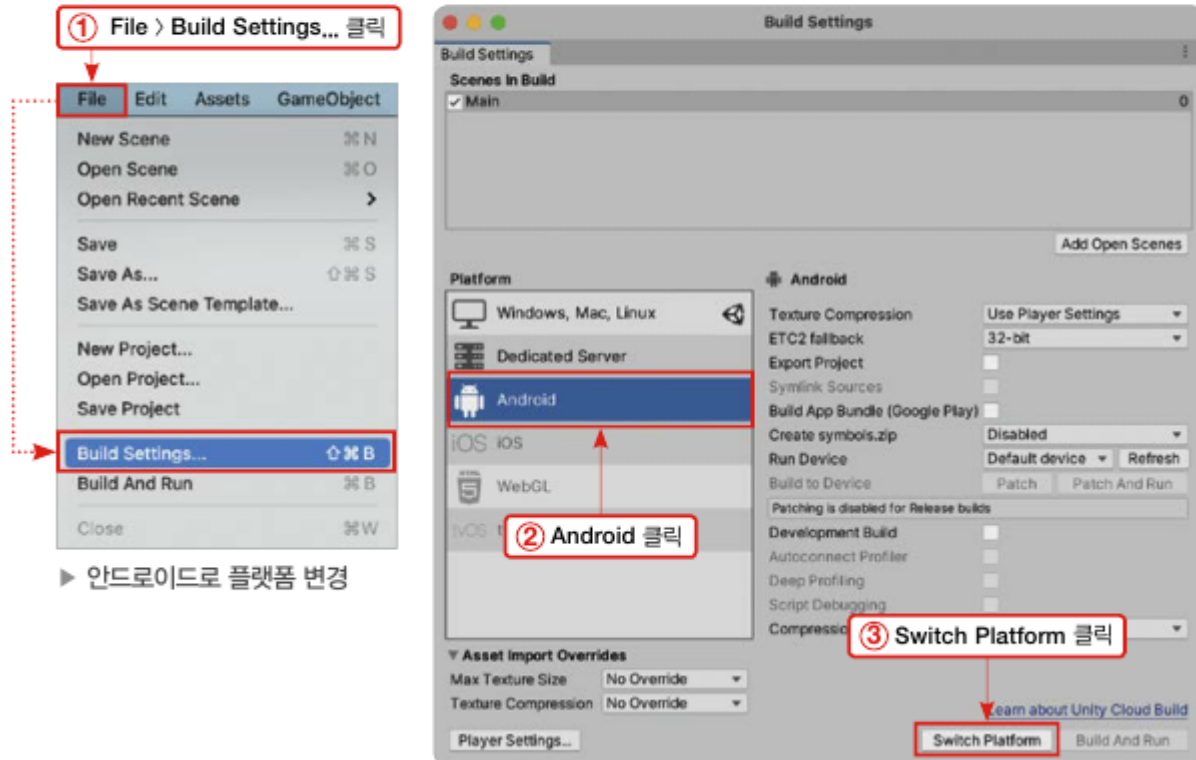


안드로이드 빌드

• 안드로이드 빌드 설정

[과정 이] 안드로이드로 플랫폼 변경

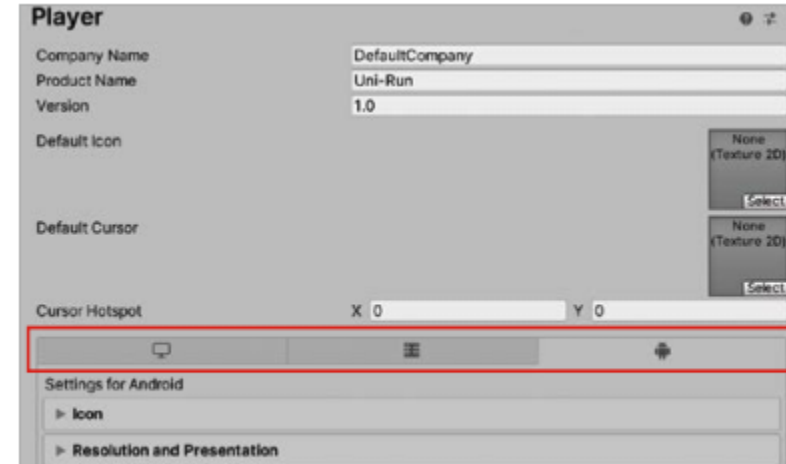
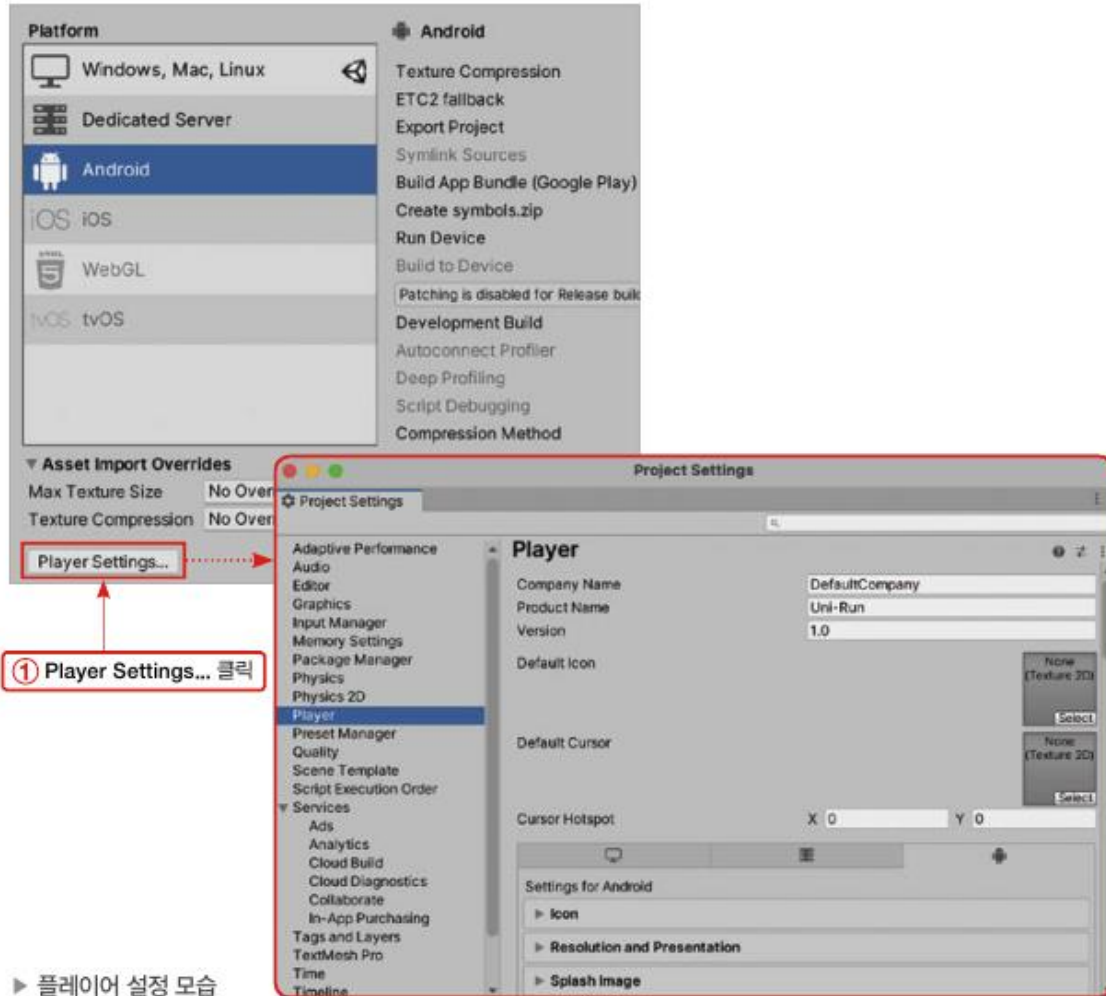
- ① 빌드 설정 열기(File > Build Settings... 클릭)
- ② Platform에서 Android 클릭
- ③ Switch Platform 클릭



안드로이드 빌드

[과정 이] 플레이어 설정으로 이동

① 빌드 설정 창에서 **Player Settings...** 클릭 > 프로젝트 설정 창이 열리고 **Player** 탭으로 이동함



▶ 플랫폼별 설정 탭

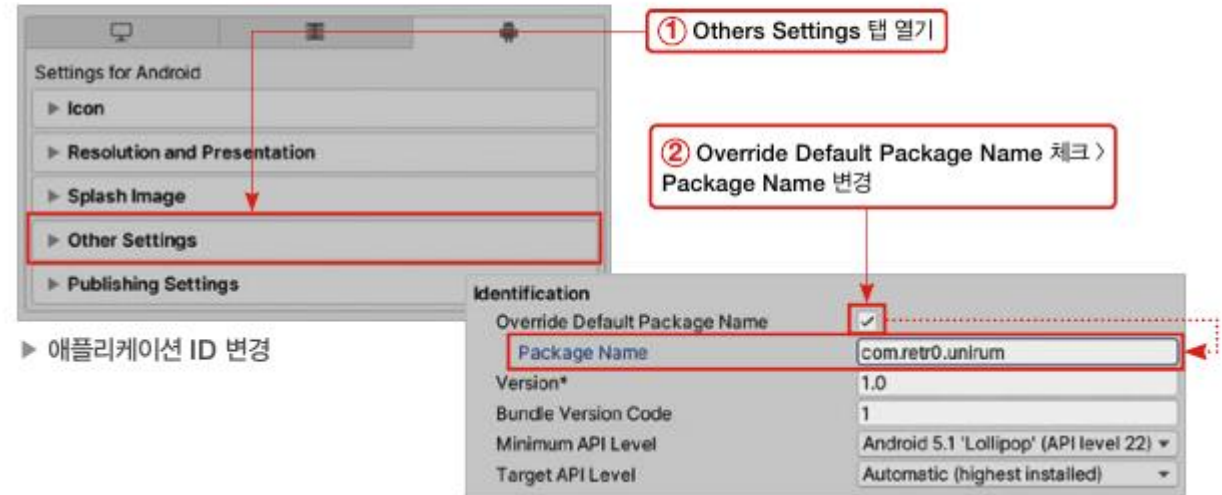
안드로이드 빌드

- 어플리케이션 ID - 앱을 구분하는데 사용

1. 영문자, 숫자, 밑줄만 사용
2. 점(.)을 이용해 두 부분 이상으로 나뉘야 함
3. 각 부분은 영문자로 시작해야 함

[과정 이] 어플리케이션 ID 변경

- ① 플레이어 설정의 Other Settings 탭 열기
- ② Override Default Package Name 체크 > Package Name 변경(Identification 항목 아래에 있음)

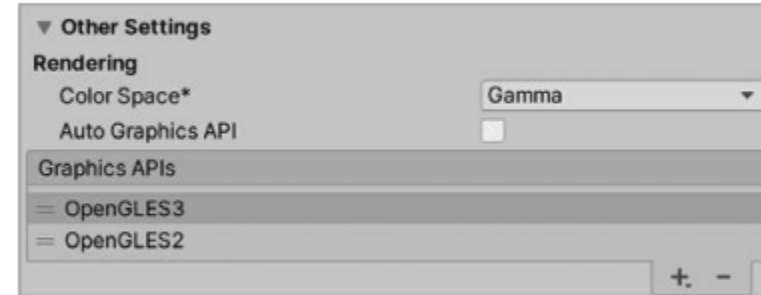


안드로이드 빌드

- Graphic APIs

사용할 그래픽스 API의 순서를 세팅

1. OpenGL ES2
2. OpenGL ES3(ES3.1 ES3.2)
3. Vulkan



▶ Rendering 설정

[과정 1] Auto Graphics API 체크

① Other Settings 탭 > Rendering 항목의 Auto Graphics API 체크



▶ Auto Graphics API 체크

안드로이드 빌드

- Android API Level

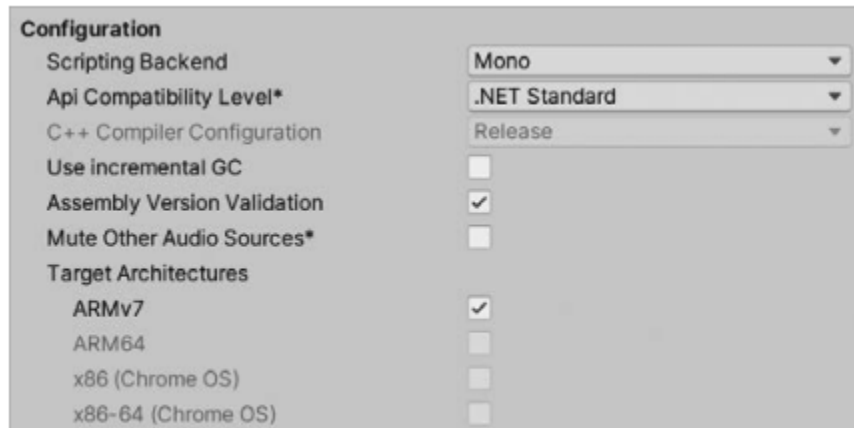
최소 요구 버전 및 타겟 버전 설정

Identification	
Override Default Package Name	<input checked="" type="checkbox"/>
Package Name	com.retro.unirum
Version*	1.0
Bundle Version Code	1
Minimum API Level	Android 5.1 'Lollipop' (API level 22) ▼
Target API Level	Automatic (highest installed) ▼

▶ API 레벨 설정

안드로이드 빌드

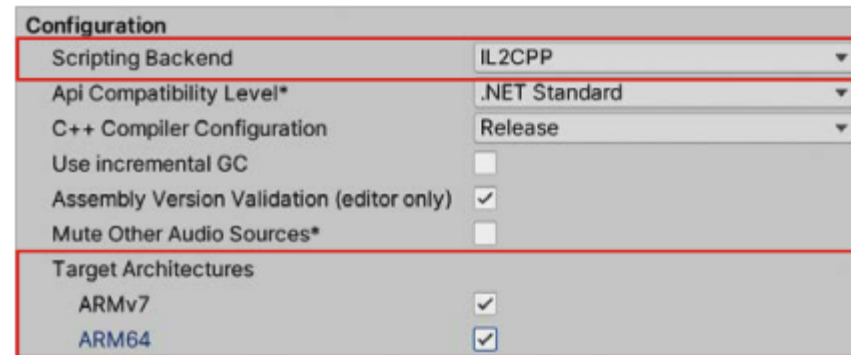
- 스크립팅 백엔드와 타킷 아키텍처 - 64비트 사용시 IL2CPP



► Configuration

[과정 이] 스크립팅 백엔드와 대상 아키텍처 변경

- ① Other Settings 탭 > Configuration 항목의 Scripting Backend를 IL2CPP로 변경
- ② target Architectures에서 ARM64 체크



► 스크립팅 백엔드와 대상 아키텍처 변경

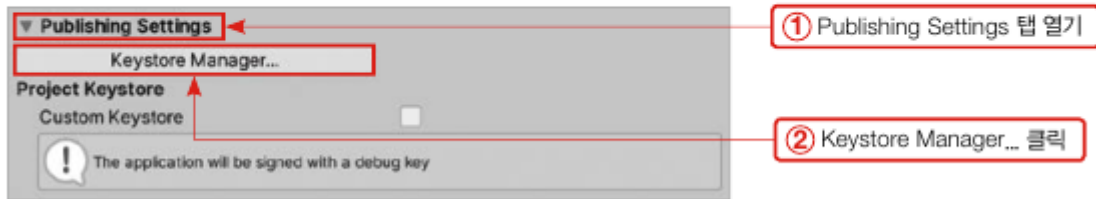
안드로이드 빌드

• 키스토어 생성

개인 키를 사용한 안드로이드 앱 서명

[과정 01] 키스토어 관리자 열기

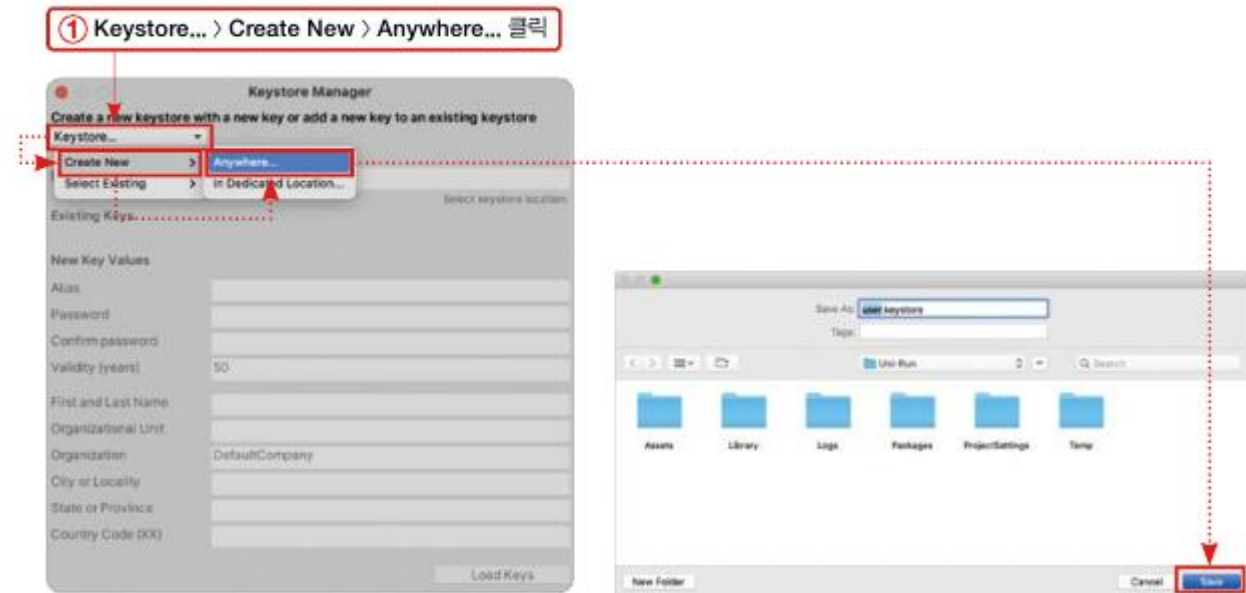
- ① 플레이어 설정의 Publishing Settings 탭 열기
- ② Keystore Manager... 클릭



▶ 키스토어 관리자 열기

[과정 02] 키스토어 파일 생성하기

- ① 키스토어 관리자 창에서 Keystore... > Create New > Anywhere... 클릭 > 적당한 장소에 키스토어 파일 저장

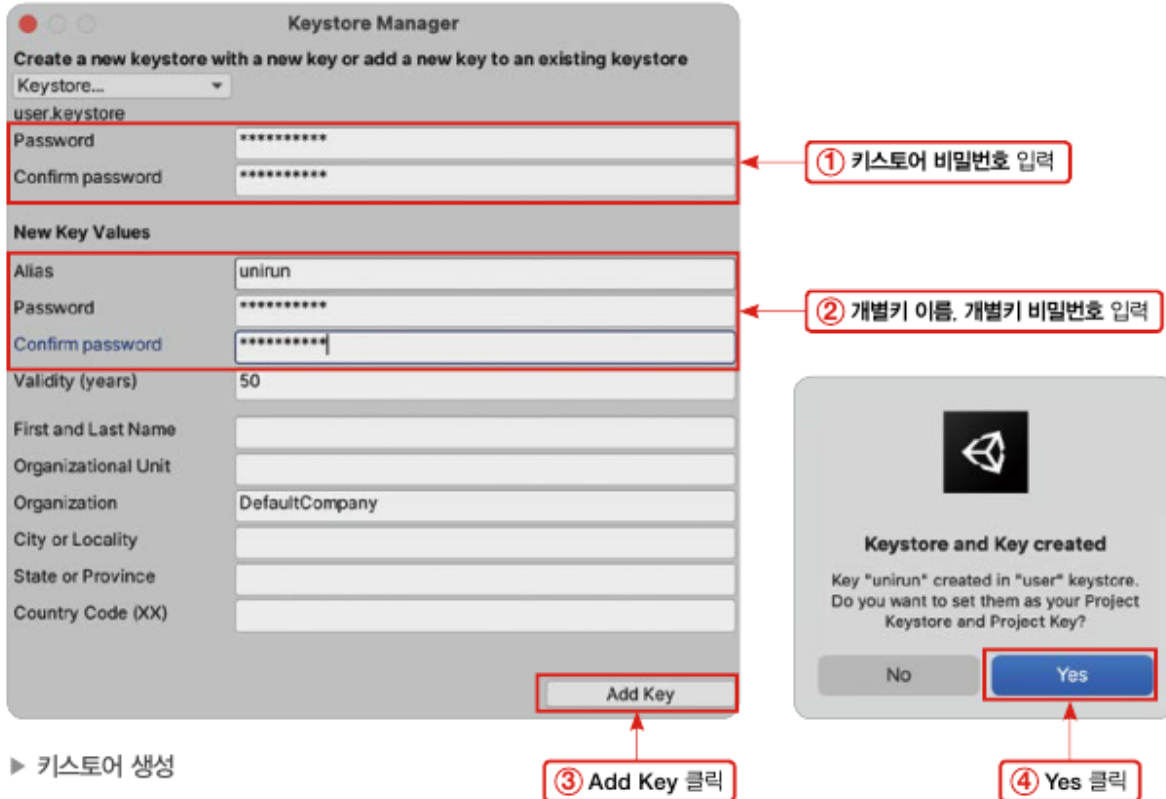


▶ 키스토어 파일 생성하기

안드로이드 빌드

[과정 03] 키스토어 생성

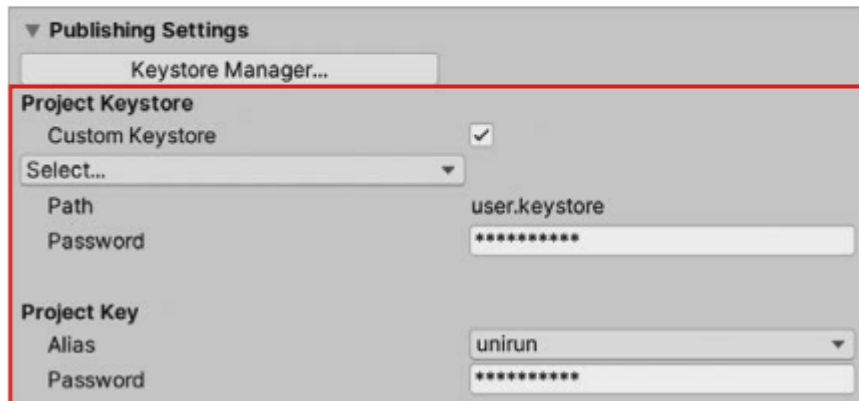
- ① Keystore 항목의 Password와 Confirm password에 사용할 키스토어 비밀번호 입력
- ② New Key Values 항목의 Alias에 사용할 개별키 이름, Password와 Confirm password에 사용할 개별키 비밀번호 입력
- ③ Add Key 클릭
- ④ 팝업 창에서 Yes 클릭(생성한 키를 프로젝트에 사용)



안드로이드 빌드

[과정 04] 키 사용

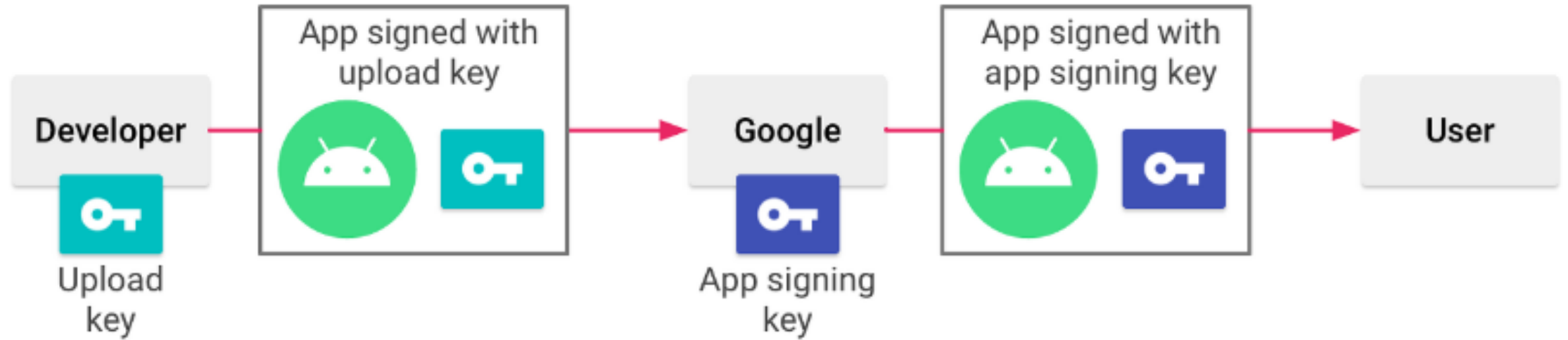
- ① Project Keystore 항목의 Password에 키스토어의 비밀번호 입력
- ② Project Key 항목의 Alias에서 개별 키 선택(저자의 경우 unirun)
- ③ Project Key 항목의 Password에 개별 키의 비밀번호 입력



The screenshot shows the 'Publishing Settings' dialog box. The 'Project Keystore' section is highlighted with a red border. It includes a 'Custom Keystore' checkbox (checked), a 'Select...' dropdown menu, a 'Path' field with the value 'user.keystore', and a 'Password' field with masked characters. The 'Project Key' section below it includes an 'Alias' dropdown menu with the value 'unirun' and a 'Password' field with masked characters.

▶ 키 사용

안드로이드 빌드

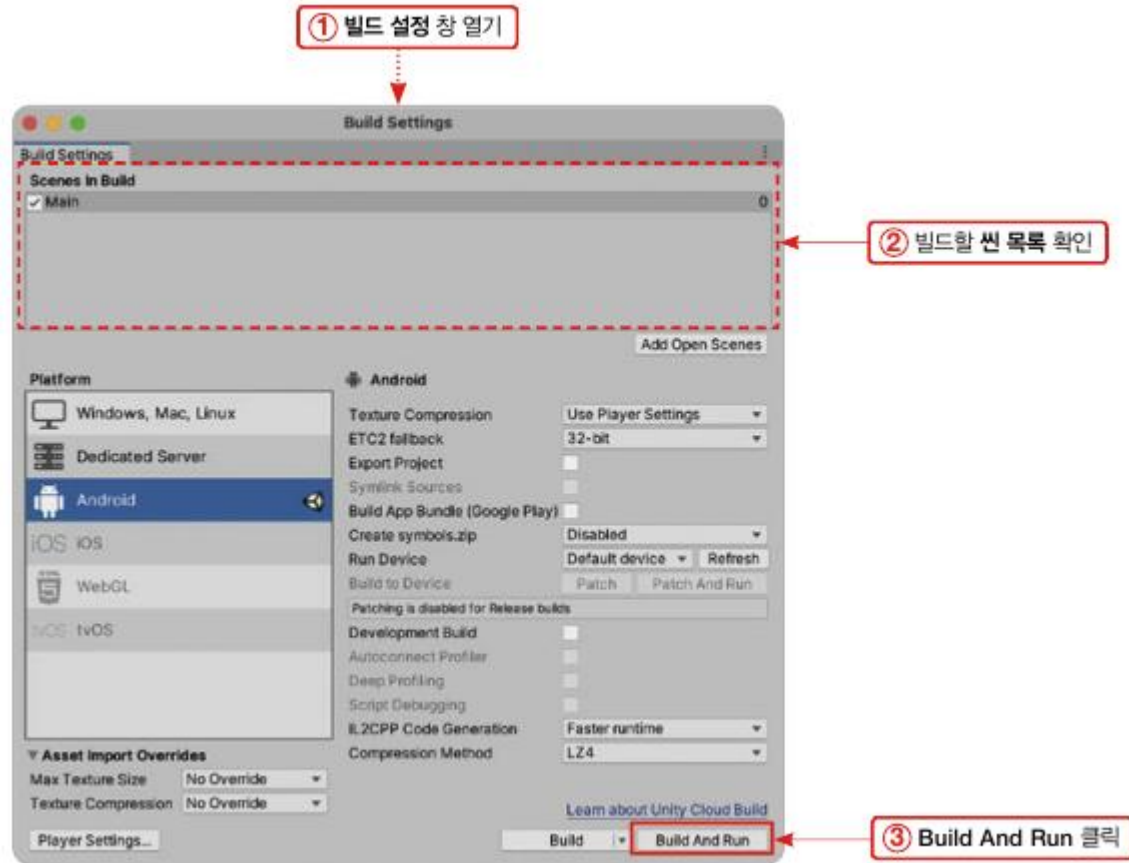


안드로이드 빌드

• 빌드 하기

[과정 이] 안드로이드 빌드 및 실행하기

- ① 빌드 설정 창 열기(File > Build Settings...)
- ② 빌드할 씬을 씬 목록에 추가했는지 확인
- ③ Build and Run 클릭



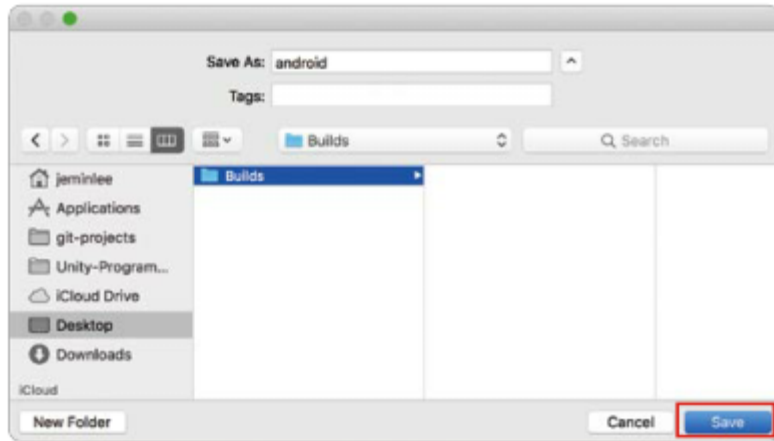
▶ 안드로이드 빌드 및 실행하기

안드로이드 빌드

• 빌드 하기

[과정 02] 빌드 이름과 위치 지정

① 빌드를 저장할 이름과 위치 지정 > Save 클릭



▶ 빌드 이름과 위치 지정

① 빌드를 저장할 이름과 위치 지정 > Save 클릭



▶ 기기에 설치된 앱

안드로이드 빌드

- 안드로이드와 유니티 사이의 통신

- 안드로이드 에서 유니티 (안드로이드 프로젝트, 안드로이드 플러그인)
UnityPlayer.UnitySendMessage("게임 오브젝트 이름", "함수 이름", "String 인자");

```
public class MainActivity extends AppCompatActivity {  
    protected UnityPlayer mUnityPlayer;
```

```
    @Override  
    public void onClick(View view) {  
        Toast.makeText(getApplicationContext(), text "왼쪽으로 이동", Toast.LENGTH_SHORT).show();  
        mUnityPlayer.UnitySendMessage( s0 "Cube", s1 "move", s2 "right");  
    }  
});
```


안드로이드 빌드

```
public class cube : MonoBehaviour {  
  
    public void move(string RorL)  
    {  
        if (RorL.Equals("left"))  
        {  
            pos_x += 0.2f;  
            Debug.Log("left move");  
        } else if (RorL.Equals("right"))  
        {  
            pos_x -= 0.2f;  
            Debug.Log("right move");  
        }  
        else  
        {  
            Debug.Log("move err : " + RorL);  
        }  
    }  
}
```

안드로이드 빌드

- 안드로이드와 유니티 사이의 통신
 - 유니티 에서 안드로이드

```
AndroidJavaObject GetJavaObject(){  
    if(javaObj == null){  
        javaObj = new AndroidJavaObject("ActivityPackageName");  
    }  
    return javaObj;  
}  
  
void Send(전달할 매개변수) {  
    GetJavaObject().Call("SendAndroid(해당 액티비티의 메서드명)", 전달할 매개변수);  
}
```

IOS 빌드

IOS 빌드

- IOS 빌드
 - 개발 환경 준비
 - 유니티 iOS 빌드 설정
 - XCode에서 iOS 빌드

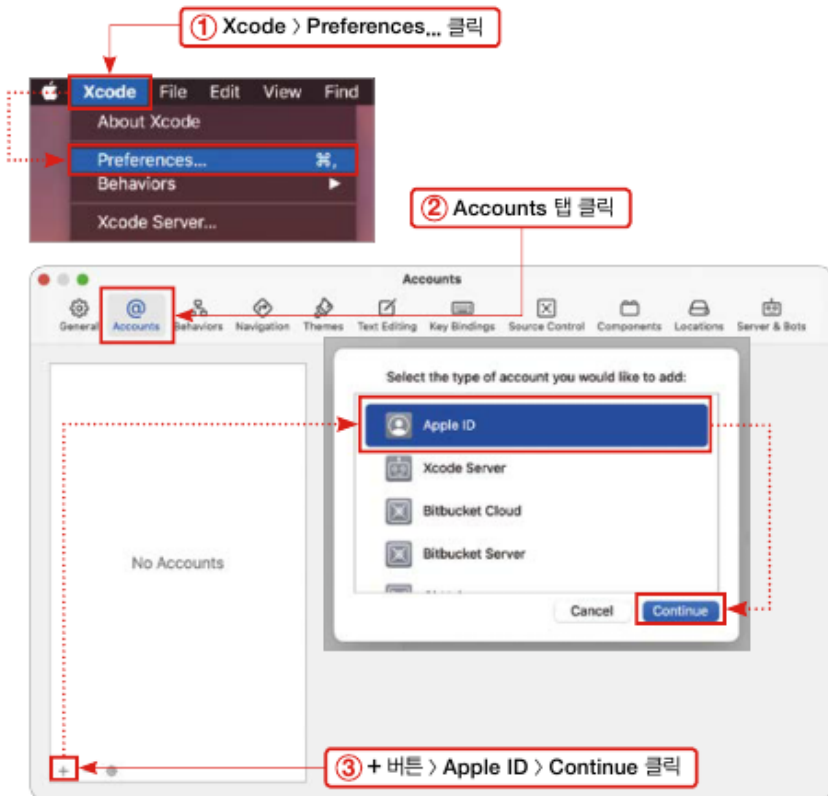
IOS 빌드

- 개발 환경 준비 (macOS Big Sur, iOS 14, XCode 12.5.1 기준)
 - 맥 장비
 - iOS기 기
 - Apple ID
 - XCode
 - 빌드할 유니티 프로젝트

IOS 빌드

[과정 1] Apple ID를 Xcode에 추가

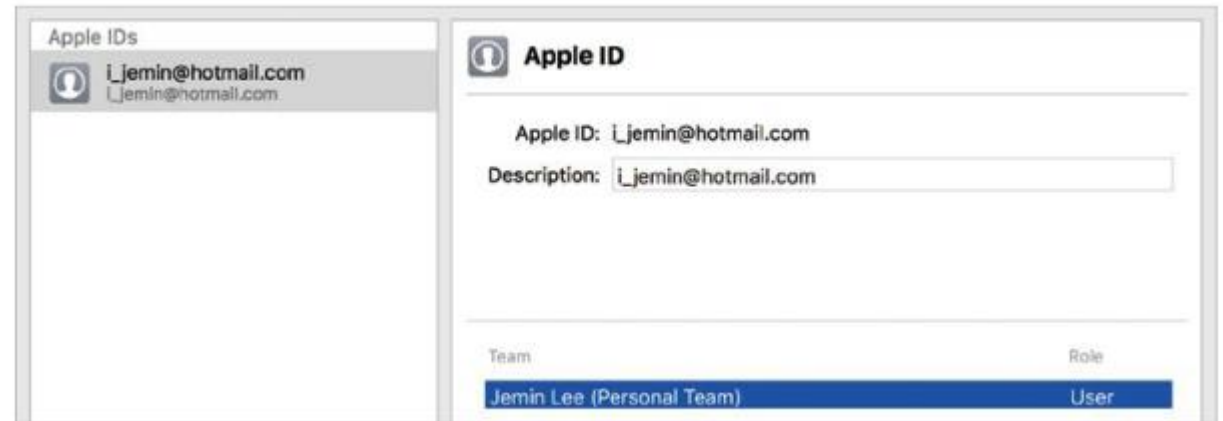
- ① Xcode를 실행하고 설정 창 열기(Xcode > Preferences... 클릭)
- ② Accounts 탭으로 이동
- ③ 좌측 하단의 + 버튼 클릭 > Apple ID > Continue 클릭
- ④ Apple ID와 패스워드 입력하고 계정 추가



▶ Apple ID를 Xcode에 추가

추가된 계정을 선택하면 Team 항목에 해당 계정이 소속된 Apple 개발자 프로그램 팀이 표시

Apple 개발자 프로그램에 등록되지 않은 Apple ID는 무료 1인 개발자 팀인 Personal Team으로 표시.



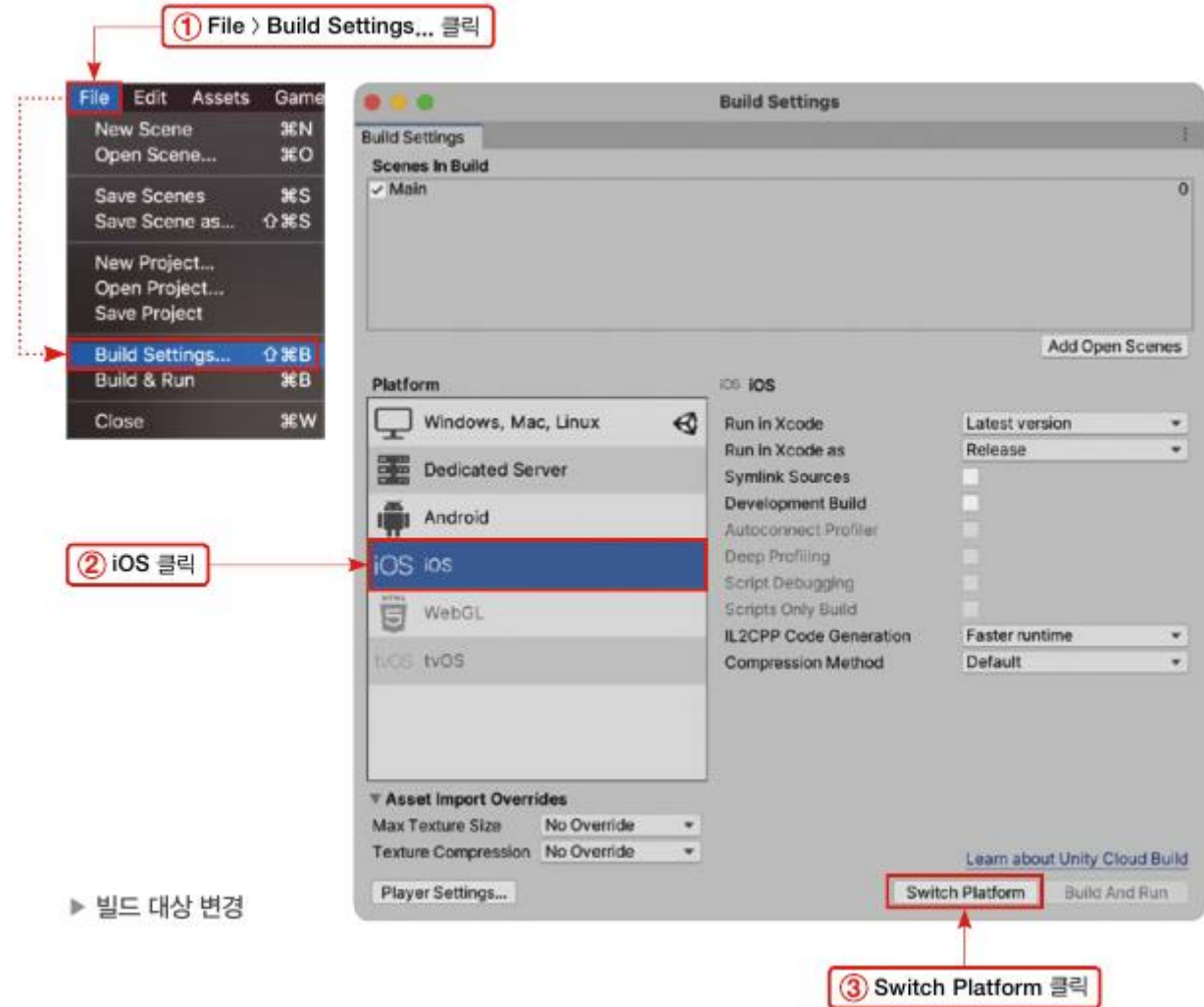
▶ 개발자 팀 정보(무료 Apple ID인 경우)

IOS 빌드

• 유니티 iOS 빌드 설정

[과정 1] 빌드 대상 변경

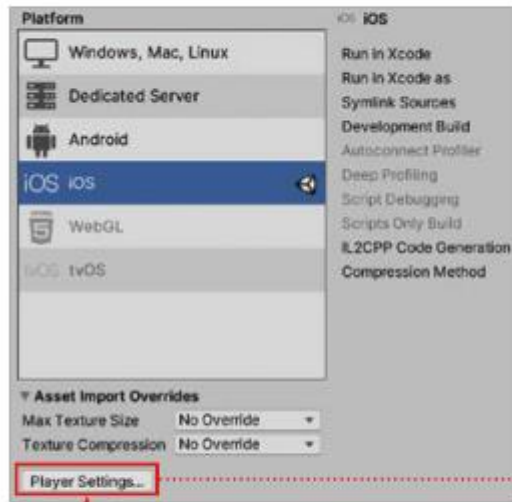
- ① 빌드 설정 열기(File > Build Settings... 클릭)
- ② Platform에서 iOS 클릭
- ③ Switch Platform 클릭



IOS 빌드

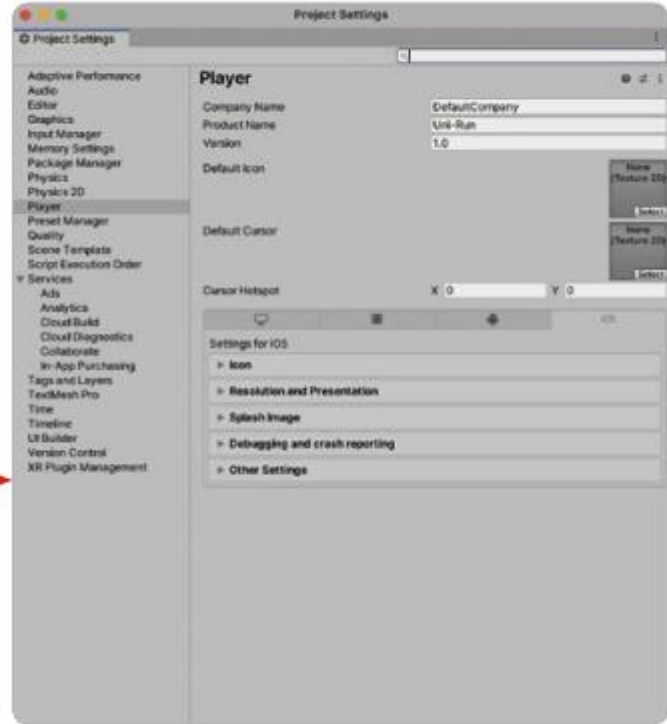
[과정 1] 플레이어 설정으로 이동

- 1 빌드 설정 창에서 **Player Settings...** 클릭



- 1 Player Settings... 클릭

▶ 플레이어 설정 모습



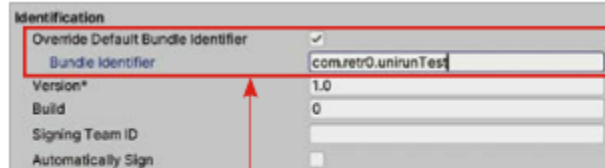
[과정 2] 번들 식별자 변경

- 1 플레이어 설정의 **Other Settings** 탭 클릭
- 2 **Override Default Bundle Identifier** 체크 > **Bundle Identifier** 변경 (Identification 항목 아래 있음)



▶ 번들 식별자 변경

- 1 Others Settings 탭 클릭

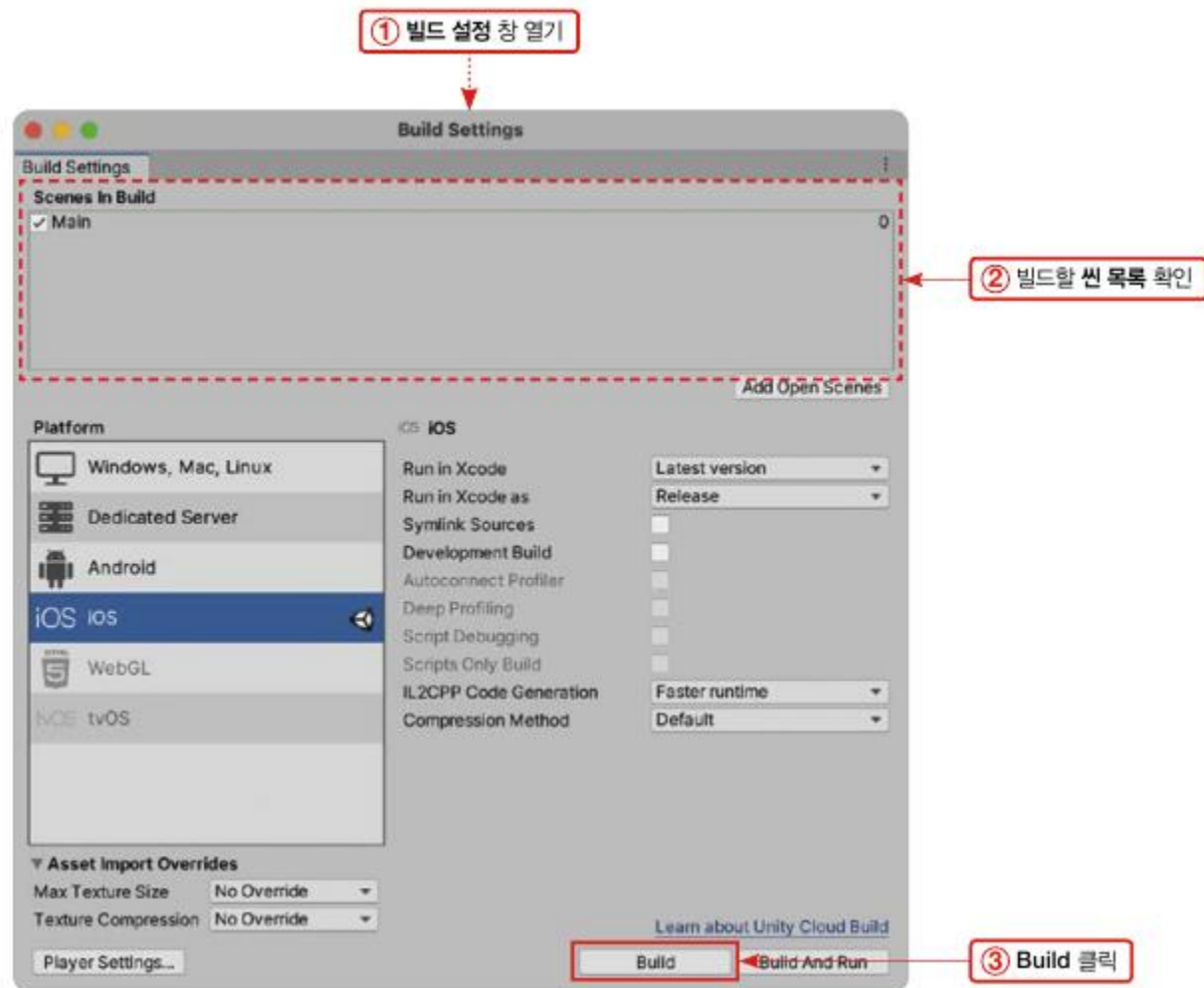


- 2 **Override Default Bundle Identifier** 체크 > **Bundle Identifier** 변경

IOS 빌드

[과정 1] Xcode 프로젝트 빌드

- ① 빌드 설정 창 열기(File > Build Settings... 클릭)
- ② 빌드할 씬을 씬 목록에 추가했는지 확인
- ③ Build 클릭

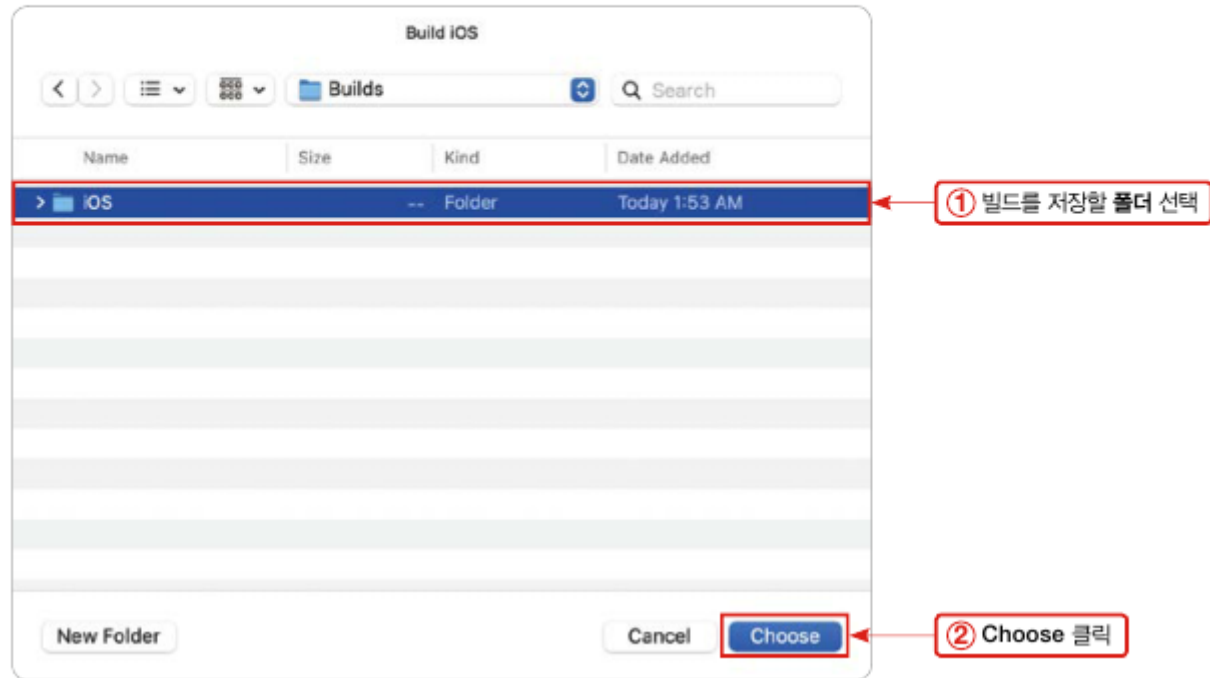


▶ Xcode 프로젝트 빌드

IOS 빌드

과정 02 Xcode 프로젝트 저장 경로 지정

- ① 탐색 창에서 빌드를 저장할 폴더 선택
- ② Choose 클릭



▶ Xcode 프로젝트 저장 경로 지정

IOS 빌드

- XCode에서 iOS 빌드

[과정 01] Xcode 프로젝트 열기

① 빌드 폴더에서 .xcodeproj 확장자를 가진 파일 실행



▶ Xcode 프로젝트 열기

IOS 빌드

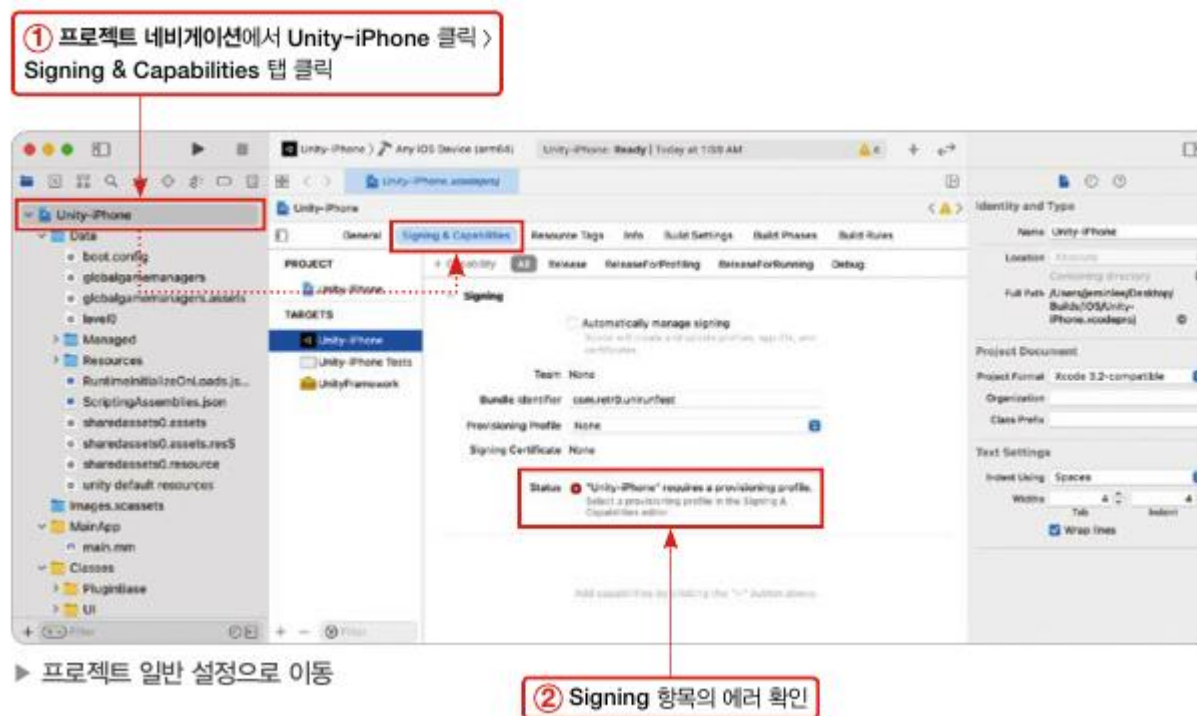
• 서명과 프로비저닝 프로파일

[과정 1] iOS 기기 연결

- ① Xcode 프로젝트를 실행한 다음 iOS 기기를 USB로 연결
- ② 만약 iOS 기기에서 Trust This Computer 팝업이 실행되면 Trust 버튼 클릭

[과정 02] 프로젝트의 서명 설정으로 이동

- ① 프로젝트 내비게이션에서 Unity-iPhone 클릭 > Signing & Capabilities 탭 클릭
- ② Status에서 Signing 항목의 에러 확인



IOS 빌드

• 서명과 프로비저닝 프로파일

[과정 03] 서명할 팀 등록

① (체크되어 있지 않다면) Automatically manage signing 체크 > Enable Automatic 버튼 클릭

② Team 항목 클릭 > 적절한 개발자 팀 선택

The image shows the Xcode 'Signing' settings panel on the left and a warning dialog on the right. The 'Signing' panel has a 'Signing' section with a checked 'Automatically manage signing' checkbox. Below it, the 'Team' dropdown is set to 'Jemin Lee (Personal Team)'. The 'Bundle Identifier' is 'com.retr0.unirunTest', the 'Provisioning Profile' is 'Xcode Managed Profile', and the 'Signing Certificate' is 'Apple Development: i.jemin@icloud.com (K7CTF...)'. A red box highlights the 'Automatically manage signing' checkbox and the 'Team' dropdown. A red arrow points from the checkbox to the warning dialog. The warning dialog has a hammer icon and the text 'Automatically manage signing will reset build settings'. It explains that allowing Xcode to automatically manage signing will reset the Code Sign Identity and Provisioning Profile build settings to default values. At the bottom of the dialog are 'Enable Automatic' and 'Cancel' buttons. A red box highlights the 'Enable Automatic' button. A red arrow points from the 'Enable Automatic' button back to the 'Team' dropdown in the 'Signing' panel. Red dashed lines connect the 'Automatically manage signing' checkbox to the warning dialog and the 'Enable Automatic' button to the 'Team' dropdown.

① Automatically manage signing 체크 > Enable Automatic 버튼 클릭

② Team 항목 클릭 > 적절한 개발자 팀 선택

Signing

☒ Automatically manage signing
Xcode will create and update profiles, app IDs, and certificates.

Team Jemin Lee (Personal Team)

Bundle Identifier com.retr0.unirunTest

Provisioning Profile Xcode Managed Profile ⓘ

Signing Certificate Apple Development: i.jemin@icloud.com (K7CTF...)

Automatically manage signing will reset build settings

Allowing Xcode to automatically manage signing will reset the Code Sign Identity and Provisioning Profile build settings to default values.

Enable Automatic

Cancel

IOS 빌드

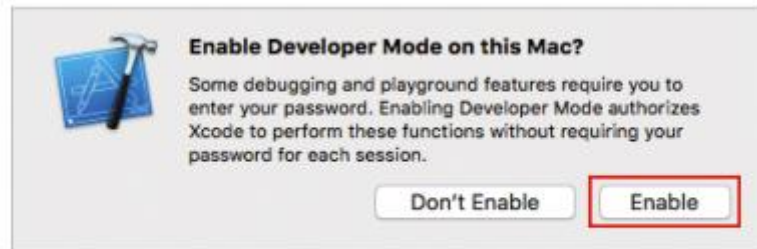
• iOS 앱 빌드

[과정 1] 빌드 대상 변경 후 빌드

- ① Xcode 상단의 Any iOS Device > 연결된 iOS 기기 클릭
- ② 상단의 Run 버튼 클릭



▶ 빌드 대상 변경 후 빌드



▶ 맥 개발자 모드 활성화 팝업



▶ 기기에 설치된 앱

기말 과제

기말 과제

- **Zombie Shooter Multiplayer**

- 멀티 Scene 구성

- 한 개 이상 기능 추가 또는 기존 기능 변경
추가 및 변경 기능 설명 PPT

- **제출물**

- 프로젝트 폴더(Assets, Packages, ProjectSettings) + 추가 기능 설명 PPT
이름(학번).zip

- **이메일 제출**

- 다음 강의 시간 (6월21일 9시) 전까지
ibanho@gmail.com

Q & A

Q & A

- 기말 과제