**1. Unity Xcode 빌드 전 준비.**

**- link.xml 생성.**

프로젝트에서 아무 파일 우클릭선택 -> iOS -> Create link.xml 클릭

통신이 새로 추가되었을 경우엔 이 기능을 반드시 실행해주세요.(추가 안 되었을 경우엔 안 해도 됩니다)



link.xml은 cpp로 변환할 때 반드시 컴파일 해야할 클래스를 지정해주는 파일입니다.

현재 JSON변환시 사용하는 클래스와 통신에 사용되는 클래스를 지정하도록 되어있습니다.

관련 기능에 수정이 필요한 경우 EodBundleEditor.cs파일의 GetClasses 함수를 수정하시면 됩니다.

**- EodProtocolEventList.cs 생성**

프로젝트에서 아무 파일 우클릭선택 -> iOS -> Create NetResponse Dictionary 클릭

통신이 새로 추가되었을 경우엔 이 기능을 반드시 실행해주세요.(추가 안 되었을 경우엔 안 해도 됩니다)



해당 파일은 cpp로 변환 할 때 JSON에서 역직렬화 할 때 배열이 들어갈 경우, List<T>를 만든 후에 배열로 변환 작업을 거칩니다. 이 때, cpp에 미리 List<배열로들어가는 클래스>를 명시적으로 지정해주지 않았으면, 해당 기능이 컴파일 되지 않아서 에러를 냅니다.

따라서 명시적으로 통신에 사용되는 배열 클래스를 명시적으로 생성하는 구문이 있는 더미 용 함수가 필요한데, 그걸 자동생성해주는 기능입니다.

관련 기능에 수정이 필요한 경우 EodBundleEditor.cs파일의 GenerateDummyListClass함수를 수정하시면 됩니다.

더미 클래스가 담긴 스크립트의 위치는 Scripts/Network/EodProtocolEventList.cs 입니다.

**- XCode 빌드 세팅**

Edit => Project Settings => Player 로 플레이어 세팅에 들어갑니다.



iOS탭에서 Other Settings에 다음과 같이 세팅이 필요합니다.

\* Scripting BackEnd -> IL2CPP

스크립트를 C++형식으로 변환합니다. iOS에서 리플렉션(JSON 등) 기능을 쓸려면 필수적으로 해야합니다.

\* Target minimum iOS Version -> 9.0

게임만 실행하는데에는 6.0으로도 충분하지만 GPGS와 Google Analytics 관련 기능을 사용하려면 9.0이상의 iOS를 요구할 필요가 있습니다.

\* Architecture -> Universal

현재 앱스토어에서 x64를 지원하지 않으면 심사 거부를 하므로 반드시 ARM64를 지원하게 세팅해줘야합니다.

다만 ARMv7까지 포함해서 용량이 상당히 커지므로, 아이폰5s이후의 64비트 기기만 지원할 예정이라면 ARM64로 세팅해서 빌드하면 됩니다.

\* XCode 빌드

Build => BuildSettings클릭.

빌드 윈도우에서 빌드플랫폼을 iOS로 설정한 후, 기타 세팅은 안드로이드와 동일하게 선택한 후 Build 버튼을 누르면 됩니다.

\* 빌드 후 XCode 자동세팅 관련

Xcode에서 각 종 플러그인을 연결해주기 위해(sqlite3, facebook sdk, navercafe 등) 세팅해야 할 것을 대부분 자동화 했습니다.

해당 기능에 관해서는 Scripts/Editor/Build/ EodFacebookPostProcessBuild.cs에 있으므로, 해당 파일의 주석을 참조하여, 추후 추가되는 플러그인에 맞춰 수정하시면 됩니다.

**2. XCode 빌드 절차**



XCode로 빌드된 프로젝트에서 **Unity-iPhone.xcworkSpace** 파일을 더블클릭해서 엽니다.

Xcodeproj로 열면 GPGS관련 코드가 연결되지 않아서 빌드 중에 에러를 뿜습니다.

\* 빌드 중에 GPGSLogins.h를 열 수 없다는 에러가 뜬다면 podinstall파일을 더블클릭하여 실행 해주세요. 빌드 후에 자동으로 실행하게 했지만, 가끔 안 될 경우가 있습니다.

**- XCode Signing 세팅**



스크린샷의 번호순서대로 클릭을 해주세요. 좌상단의 폴더아이콘 -> Unity-iPhone -> General -> Signing -> Team

여기에서 Nox Games Inc.를 선택해서 빌드해야 합니다.

\* 만약 에러가 발생하면, Nox쪽에서 애플개발자센터에 AppleID를 개발자로 등록안한 것입니다. 녹스에 자신의 AppleID(이메일주소)를 보내서 등록해달라고 요청한뒤, Add an Account…를 클릭하여 AppleID로그인 해서 등록해줘야합니다.

**- XCode 권한 세팅**



실제 게임 서비스에서 사용할 부가 기능을 선택해서 On으로 선택해줍니다. 서비스시 사용하게 될 예상 기능은 다음과 같습니다.

\* Push Notifications(푸쉬알람)

\* In-App Purchase(인앱 결제)

 **- 빌드 시작(내부 테스트용)**

우선 빌드용 맥에 테스트용 기기를 연결해줍니다.



좌상단에서 Unity-iPone옆에 아이폰 아이콘을 클릭하시면 빌드 후에 넣을 기기를 선택 가능합니다.

빌드한 게임을 넣을 디바이스를 선택하고 좌측의 ▶버튼을 누르면 빌드가 진행됩니다.

**- 빌드(외부 테스트 배포용)**

해당 절차는 복잡하므로 별도의 설명된 페이지를 링크 걸겠습니다.

Ad-hoc방식을 통하여 공유하는 법을 추천합니다.

<http://devlecture.tistory.com/entry/%ED%8C%8103-iOS-%EC%95%B1-%EB%82%B4%EB%B6%80-%EB%B0%B0%ED%8F%AC-adhoc-OTA-%EA%B0%80%EC%9D%B4%EB%93%9C-Dropbox-%ED%99%9C%EC%9A%A9>

3. 기타 사항

- 어셋 번들은 Platform이 iOS로 세팅된 컴퓨터에서 기존 안드로이드와 동일하게 만드시면 됩니다. 맥이 아닌 윈도우 컴퓨터에서도 iOS용 어셋 번들이 정상적으로 빌드되는 것을 확인 하였으므로, 맥의 성능이 나쁜 경우에는 성능 좋은 윈도우 PC에서 빌드해도 됩니다.



- 통신이 추가될 경우

[Newtonsoft.Json.JsonObject(Newtonsoft.Json.MemberSerialization.Fields)]

통신에 사용할 클래스 위에 JSON오브젝트로 쓸 것이라고 Attribute로 선언해야 합니다. 신규 통신에서 에러가 발생한다면 이것 때문에 문제가 되는 것입니다

마찬가지로 통신에서 주고받을 때 사용할 클래스들(ex. Player, Worth등)도 반드시 선언 해줘야 합니다.

- 어셋번들 자동 업데이트



베이스쪽 어셋번들은 BaseUpdate\_iOS.bat, 버전데이터는 VersiondataUpdate\_iOS.bat를 실행시켜서 올리시면 됩니다. 사용법은 기존 안드로이드용과 동일하며, 프로젝트 Output->FTPAuto폴더 안에 있는 WinSCP자동화사용법.txt를 참고하셔서 사용하시면됩니다.