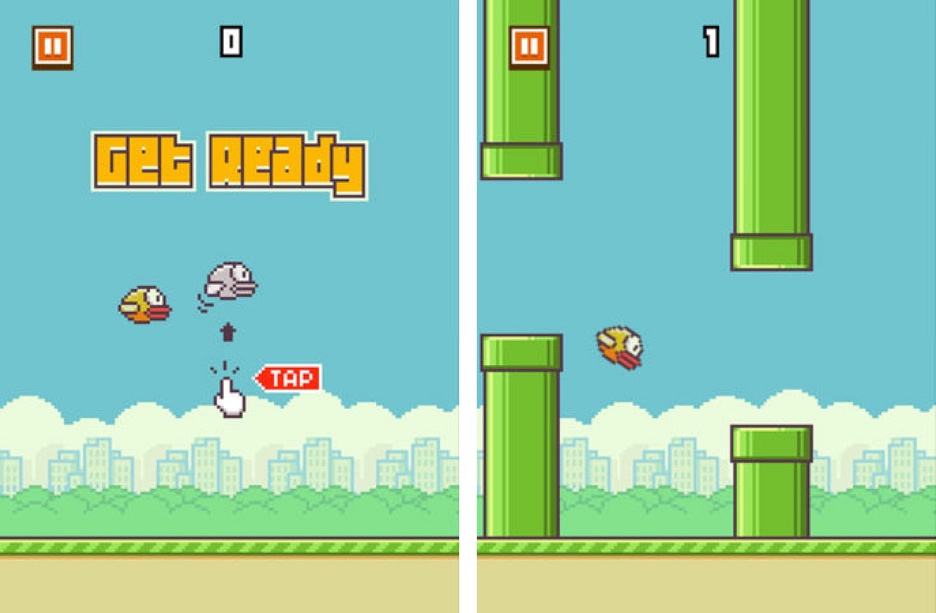
Flappy Bird 게임에 CloudBread 붙여보기

# Flappy Bird 게임 만들기

Flappy Bird 라는 게임은 매우 단순한 Running 게임 입니다.



원본 게임은 현재 앱 스토어에는 제공되지 않고 있지만 지속적으로 패러디 게임들이 개발되고 있습니다. 뿐만 아니라 개발 하기 매우 쉽기 때문에 처음 게임을 개발을 해보시는 분들에게 부담 없이 만들어 볼 수 있을 최적의 게임이라고 생각합니다.

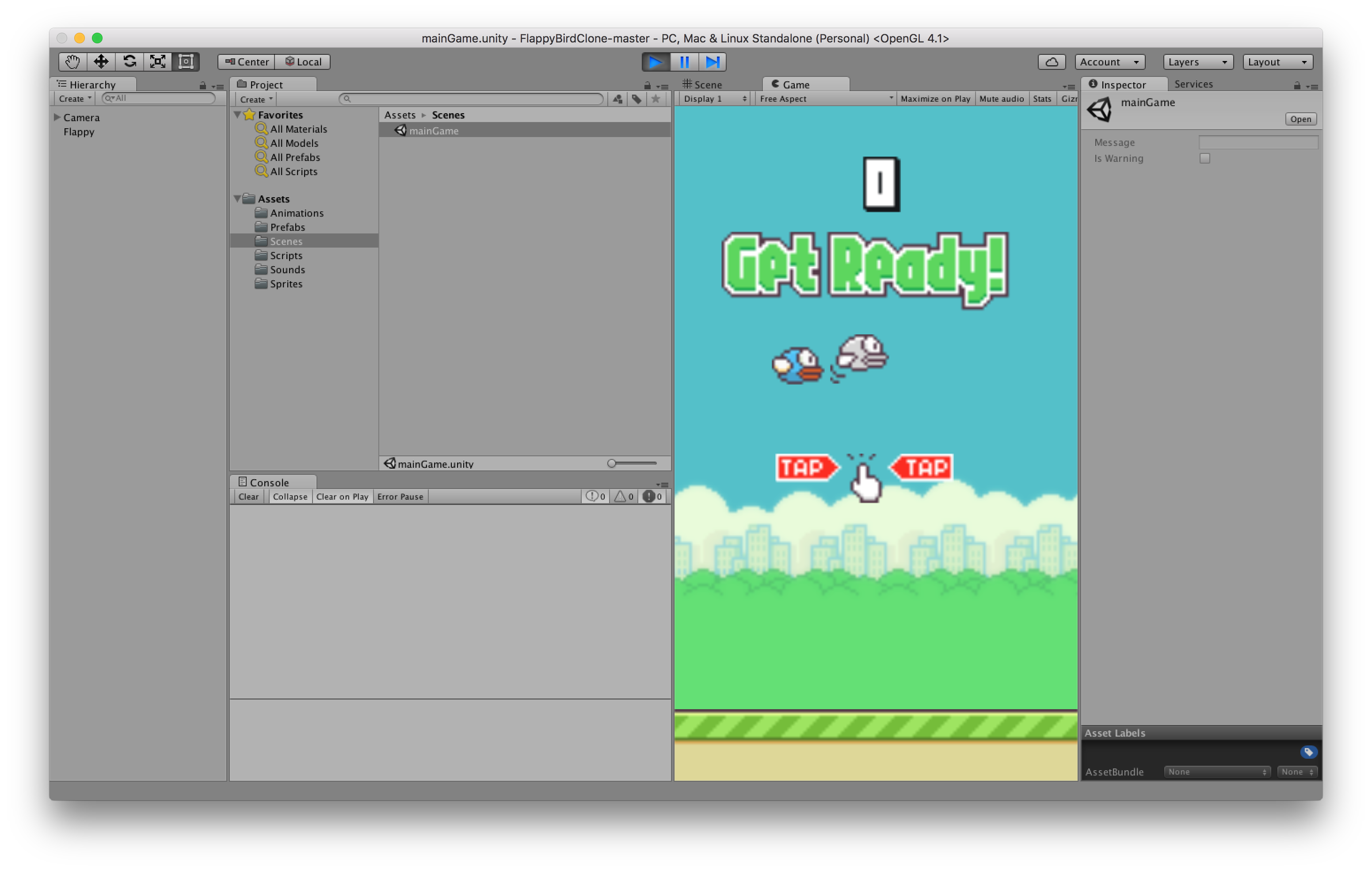
실제 게임을 만드는 과정은 다음의 블로그를 참고하시면, Unity 3D를 사용하여 개발을 진행할 수 있습니다.

[*https://dgkanatsios.com/2014/07/02/a-flappy-bird-clone-in-unity-source-code-included-3/*](https://dgkanatsios.com/2014/07/02/a-flappy-bird-clone-in-unity-source-code-included-3/)

시간 관계상 다음 주소에서 필요한 소스코드를 다운받도록 하겠습니다.

[*https://github.com/dgkanatsios/FlappyBirdClone*](https://github.com/dgkanatsios/FlappyBirdClone)

# Unity 에서 Flappy Bird 실행하기

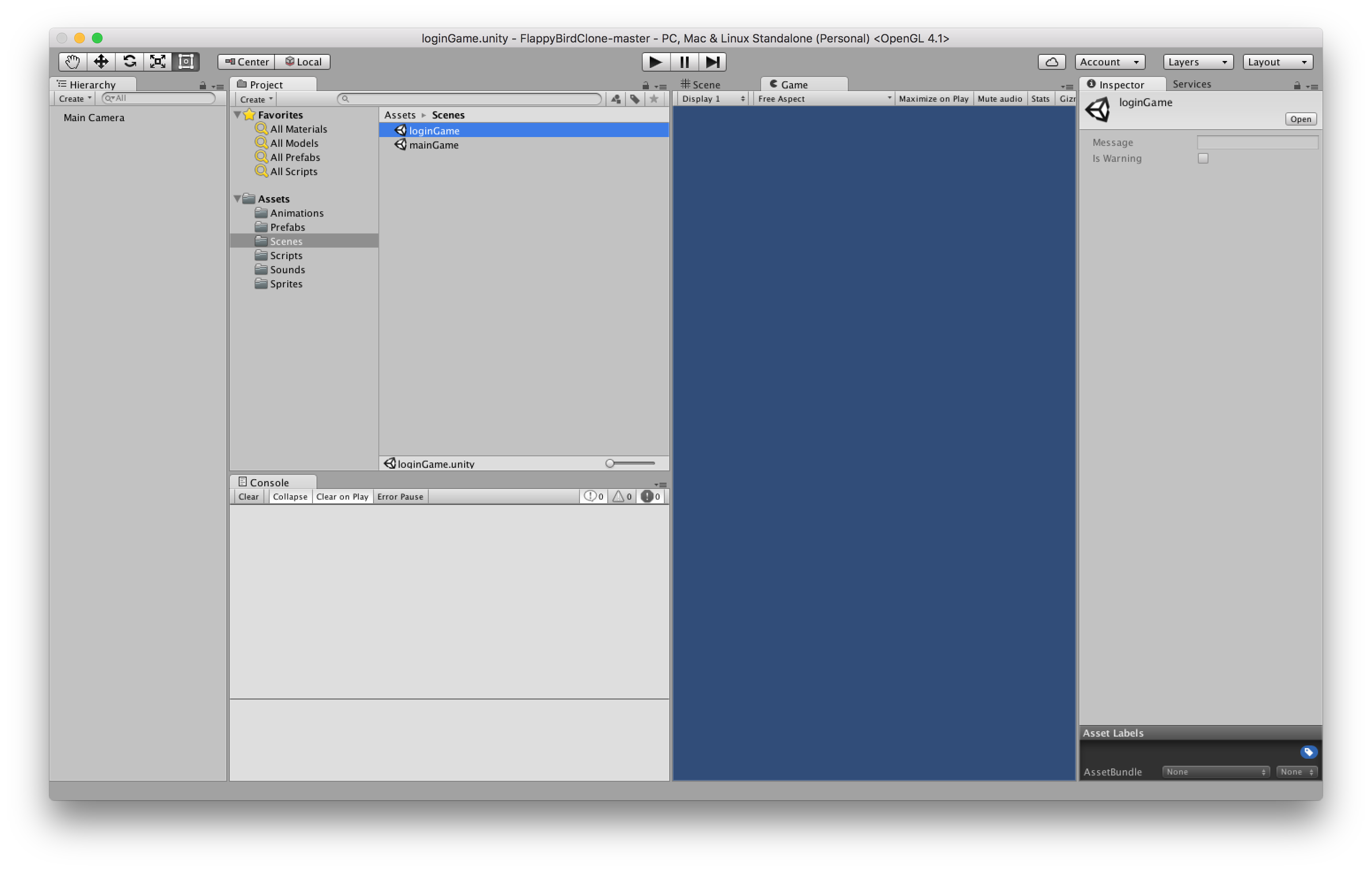


**Asset - Scenes - mainGame** 을 열면 위와 같이 게임을 실행 할 수 있다.

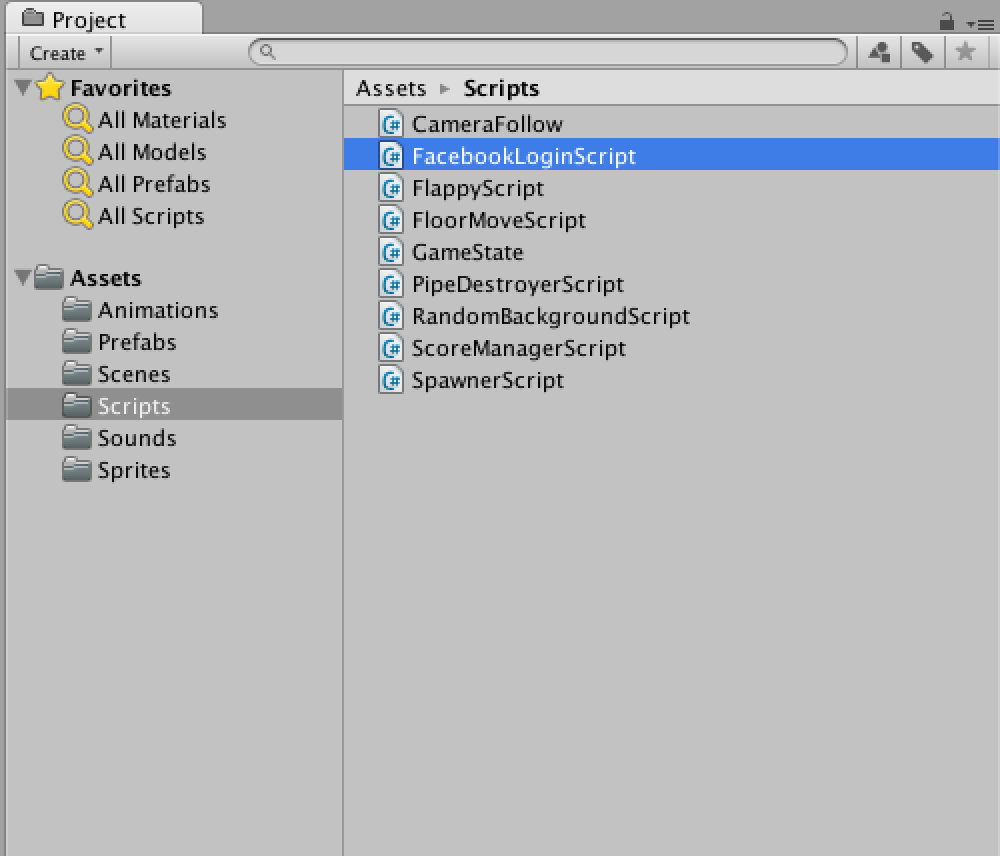
# Unity 에 로그인 화면 만들기

|  |
| --- |
|  |

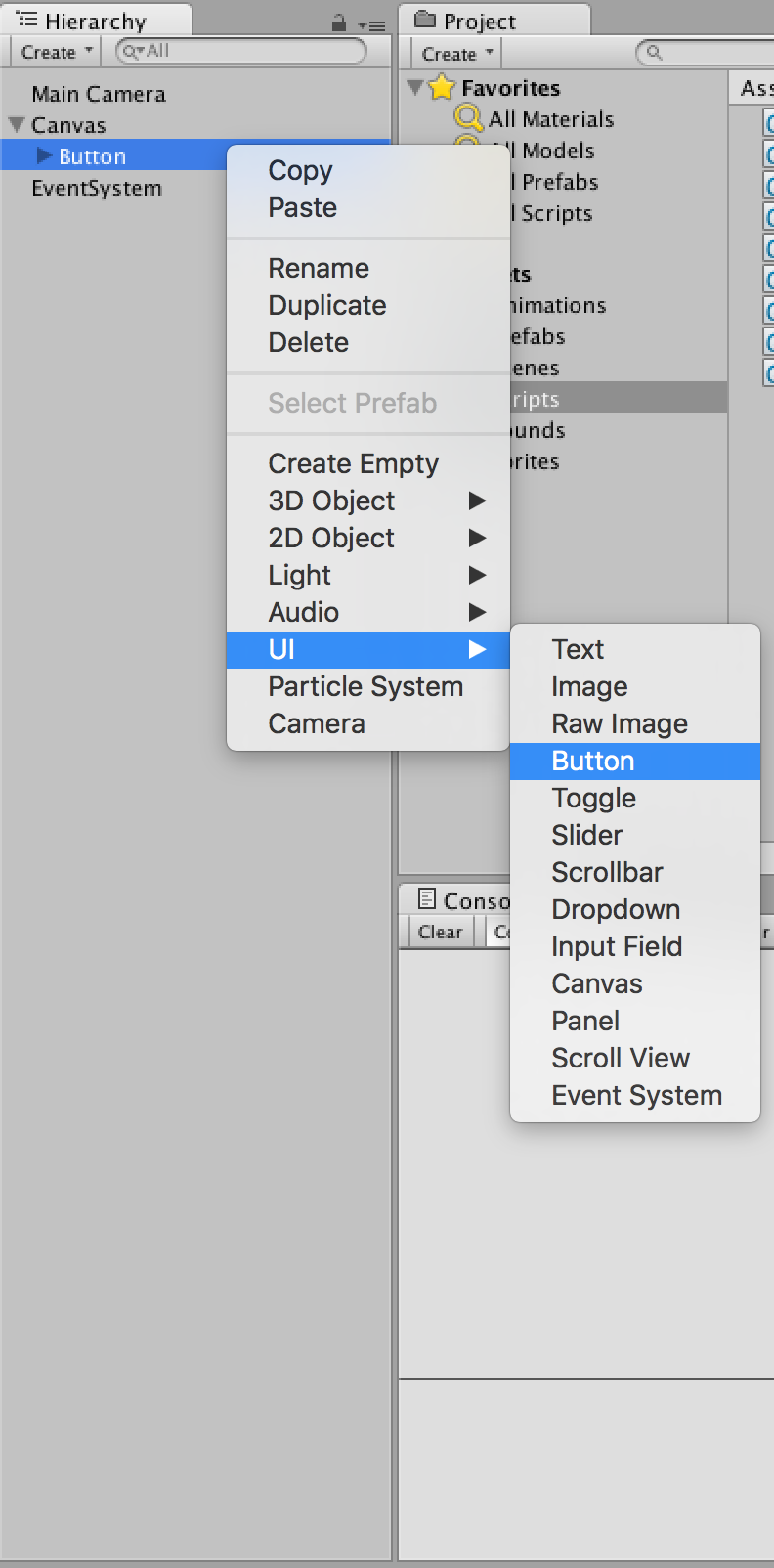
**마우스 오른쪽 버튼 클릭 - Create - Scene** 을 클릭하여 loginGame 씬 생성



**Assets – Scrips** 에 **FacebookLoginScript.cs (C# Script)** 생성하기

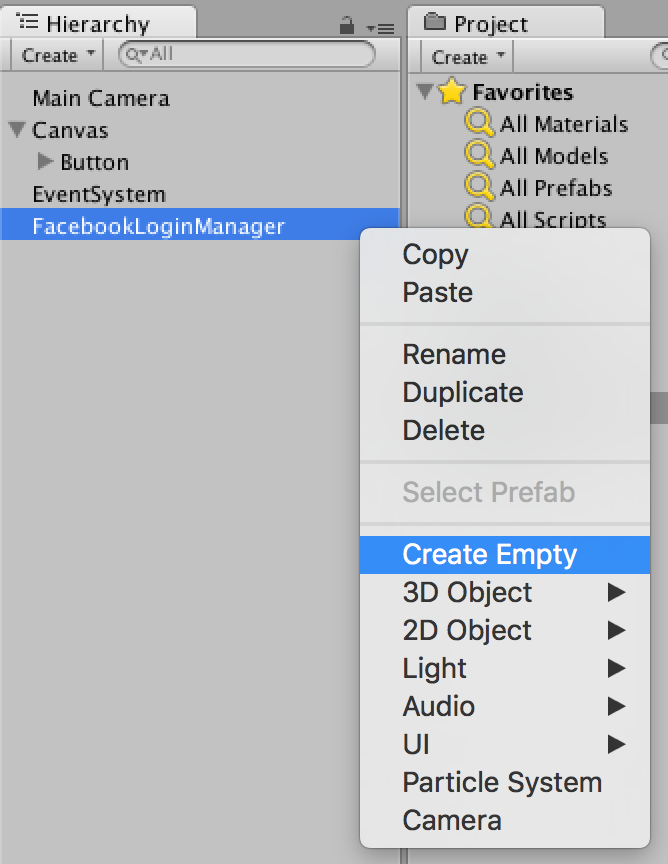


loginGame 씬에서 **마우스 오른쪽 – UI – Button** 클릭 해서 버튼 만들기

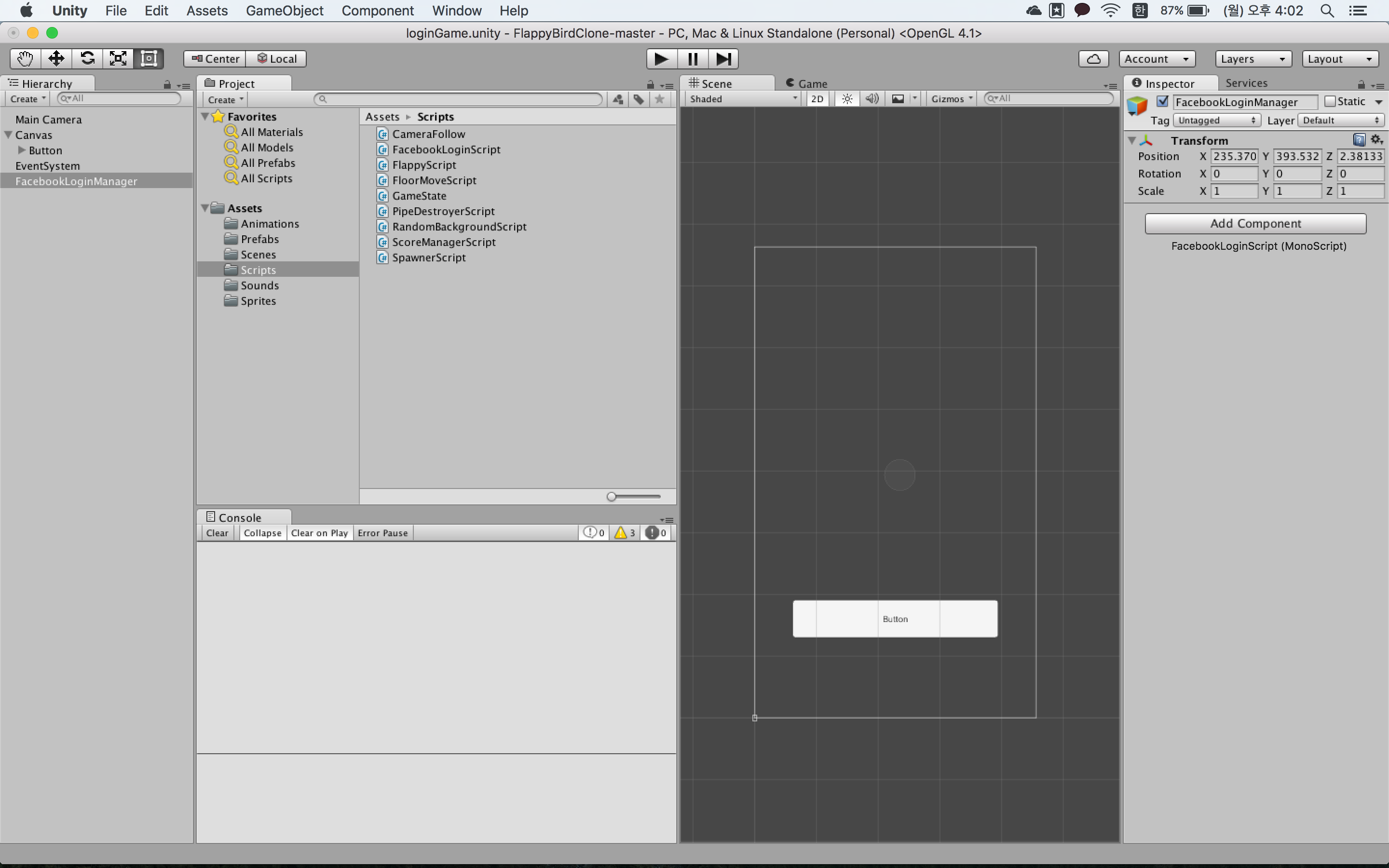
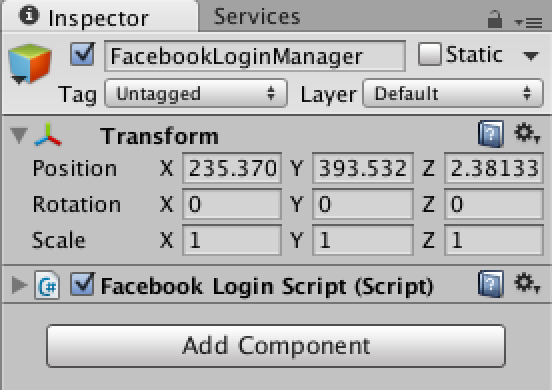


**마우스 오른쪽 – Create Empty** 클릭해서

FacebookLoginManager라는 Gameobject 만들기

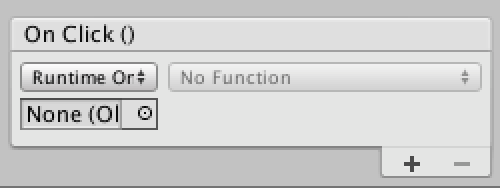
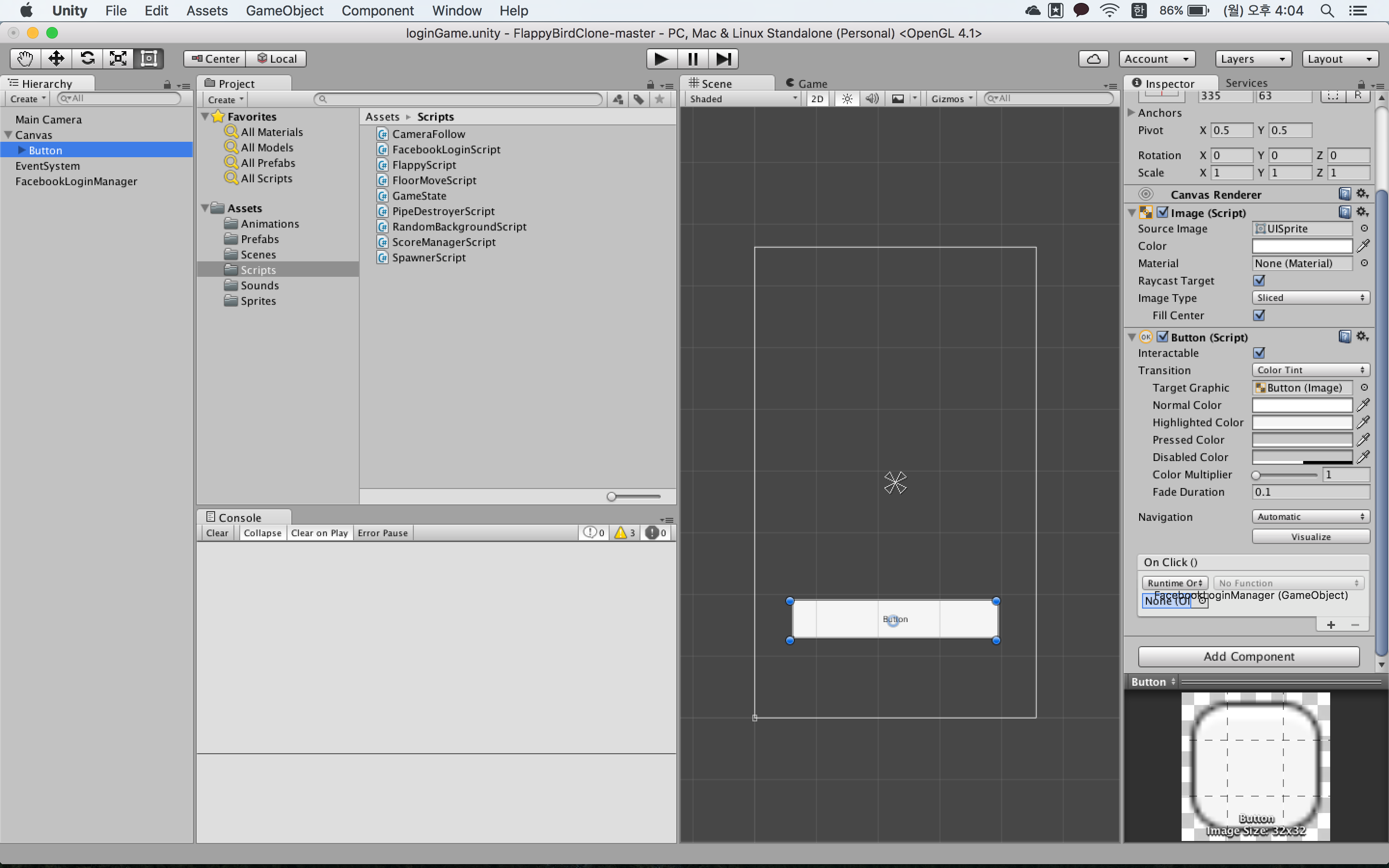
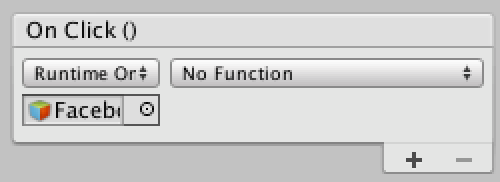


아까 만든 FacebookLoginScript를 드래그 해서 FacebookLoginManager에 추가하기

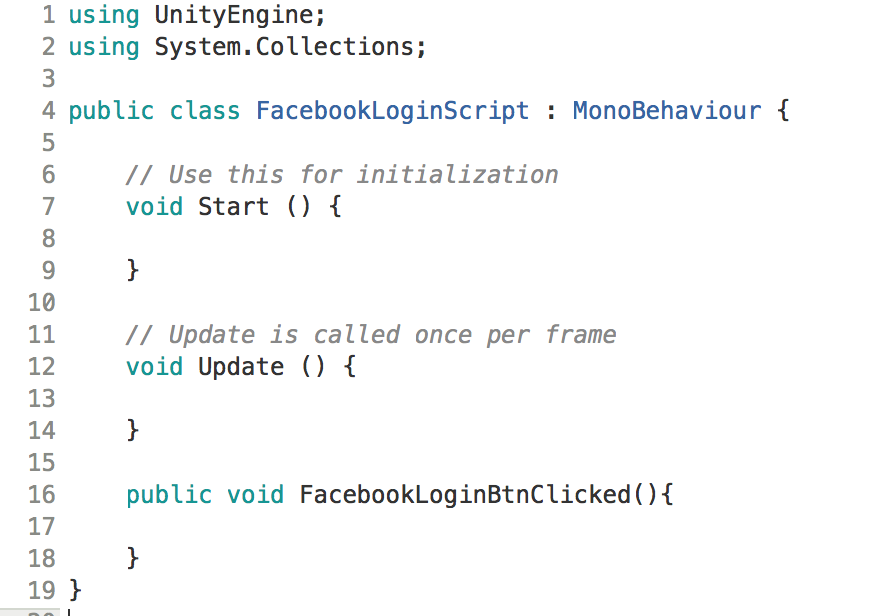
 

아래 순서대로 새로 만든 버튼 클릭 – Inspector - On Click () - + 버튼 클릭

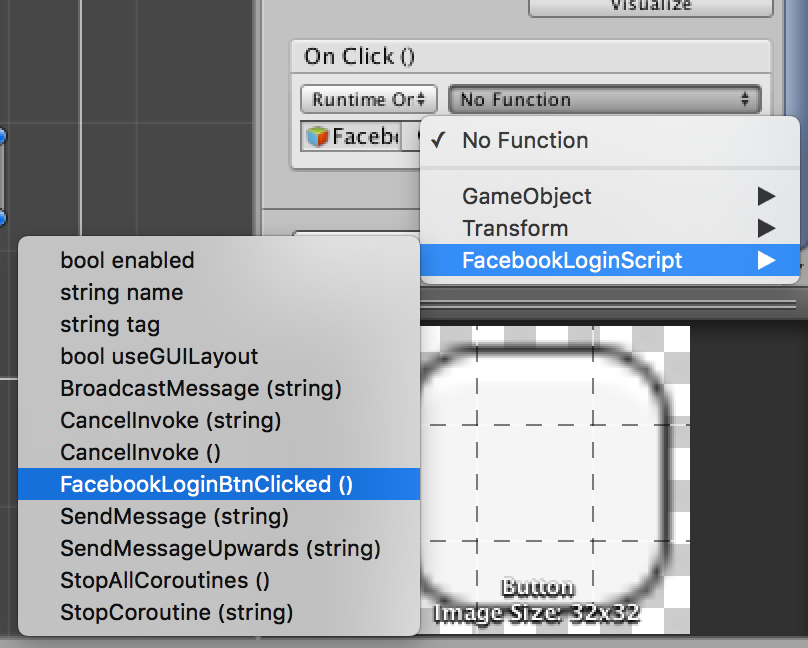
None (Object)에 FacebookLoginManager 드래그해서 놓기

FacebookLoginScipt.cs 에 아래와 같이 FacebookLoginBtnClick() 메소드 추가



**No Function – FacebookLoginScript – FacebookLoginBtnClicked()** 클릭



# Facebook SDK 를 이용한 로그인 구현

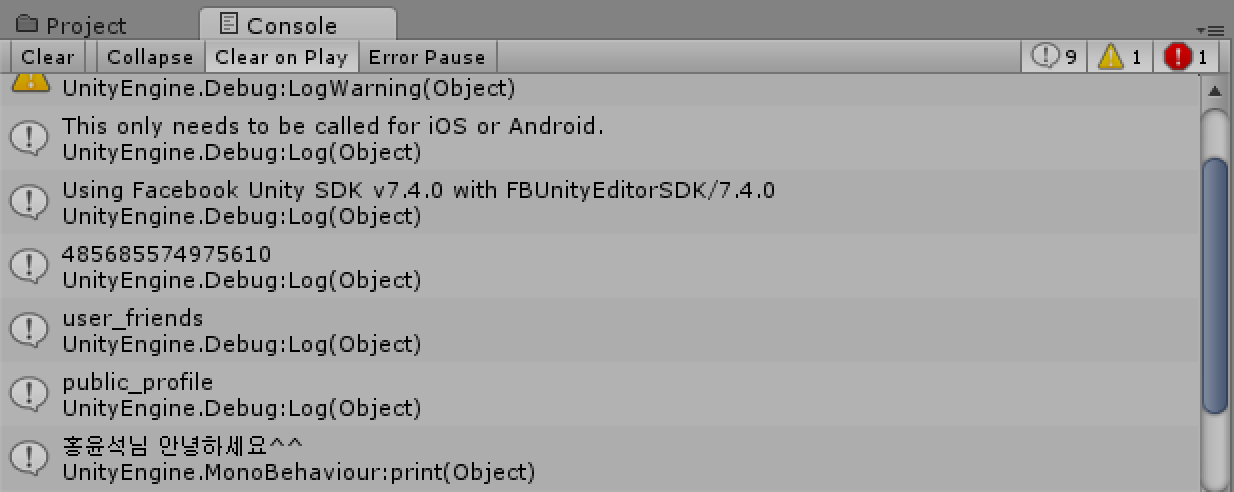
1. Facebook에 앱 추가하기
2. Azure Mobile App 에 페이스북 앱 추가하기

|  |
| --- |
| 1. Facebook 개발자 웹 사이트로 이동한 다음 Facebook 계정 자격 증명으로 로그인합니다. 2. (선택 사항) 아직 등록하지 않은 경우 **My Apps**, **Register as a Developer**를 차례로 클릭하고 정책에 동의한 후 등록 단계를 따릅니다. 3. **My Apps** > **Add a New App** > **Website** > **Skip and Create App ID**를 클릭합니다. 4. **Display Name**에 앱의 고유한 이름을 입력하고 앱에 대한 **Category**를 선택한 다음 **Create App ID**를 클릭하고 보안 검사를 완료합니다. 새 Facebook 앱에 대한 개발자 대시보드로 이동됩니다. 5. **App Secret** 필드에서 **Show**를 클릭하고 요청될 경우 암호를 제공한 다음 **App ID** 및 **App Secret** 값을 적어 둡니다. 나중에 이 값을 사용하여 Azure에서 응용 프로그램을 구성합니다. - AZURE.NOTE - **참고:** **보안 정보** 앱 암호는 중요한 보안 자격 증명입니다. 다른 사람과 이 암호를 공유하거나 클라이언트 응용 프로그램 내에 배포하지 마세요.   왼쪽 탐색 모음에서 **Settings**를 클릭하고, **App Domains**에 모바일 서비스의 도메인을 입력하고, 선택적 **Contact Email**을 입력하고, **Add Platform**을 클릭하고, **Website**를 선택합니다.   1. 왼쪽 탐색 모음에서 **Settings**를 클릭하고, **App Domains**에 모바일 서비스의 도메인을 입력하고, 선택적 **Contact Email**을 입력하고, **Add Platform**을 클릭하고, **Website**를 선택합니다.      1. **Site URL**에 모바일 서비스의 URL을 입력하고 **Save Changes**를 클릭합니다. 2. **Show**를 클릭하고 요청될 경우 암호를 제공한 후 **App ID** 및 **App Secret** 값을 기록해 둡니다.      1. **Advanced** 탭을 클릭하고, **Valid OAuth redirect URIs**에 아래URL 형식 입력한 다음 **Save Changes**를 클릭합니다.   **https://<mobile\_service>.azure-mobile.net/login/facebook**  아래 사이트에서 위 내용에 대해 자세하게 볼 수 있습니다.  *https://azure.microsoft.com/ko-kr/documentation/articles/mobile-services-how-to-register-facebook-authentication/* |

1. SDK 다운로드 하기 ([*https://developers.facebook.com/docs/unity*](https://developers.facebook.com/docs/unity))
2. Unity 프로젝트에 SDK 추가하기 (위 사이트의 Getting Started 참고)
3. loginGame 에서 버튼 클릭했을 때, Facebook Permission 받아오도록 구현 (위 사이트의 Example 참고해서 FacebookLoginScript 구현)

|  |
| --- |
|  |

페이스북 SDK 를 사용하여 로그인 성공했을 때



# CloudBread 서버 호출하기

1. Http Request 를 쉽게 할 수 있도록 다음과 같은 파일 추가하기

|  |
| --- |
| [*https://github.com/CloudBreadProject/3rd-Offline-Camp*](https://github.com/CloudBreadProject/3rd-Offline-Camp)  위 주소에서 CloudBread 폴더 다운받고, 프로젝트에 추가하기 |

1. Facebook에서 발급한 UserToken 을 사용하여, 애저 모바일 앱 토큰 발급받기(<https://www.github.com/cloudbreadproject/CloudBread-Client-Unity> 참조)

애저에서 발급받은 Token과 UserID

|  |
| --- |
| {  "authenticationToken" : "<발급받은 Token>",  "user" : {  "userId" : "sid:<발급받은 sid>"  }  } |

* authenticationToken 을 호출 할 때마다 헤더에 X-ZUMO-AUTH 값으로 넣어야만 서버로부터 응답을 받을 수 있다.
* userId의 sid 를 게임에서 유일한 값 (memberID)로 사용

# CloudBread 에 회원가입하기

1. Facebook에서 UserID와 이름을 가져와서 CloudBread 서버에 등록하기 CBInsRegMember API 호출 (PostMan 참고)
2. 유효한 사용자이면, 게임 화면으로 넘어가기

CBInsRegMember API호출 시, 새로 가입 된 사용자 응답

|  |
| --- |
| {  "result": "2"  } |

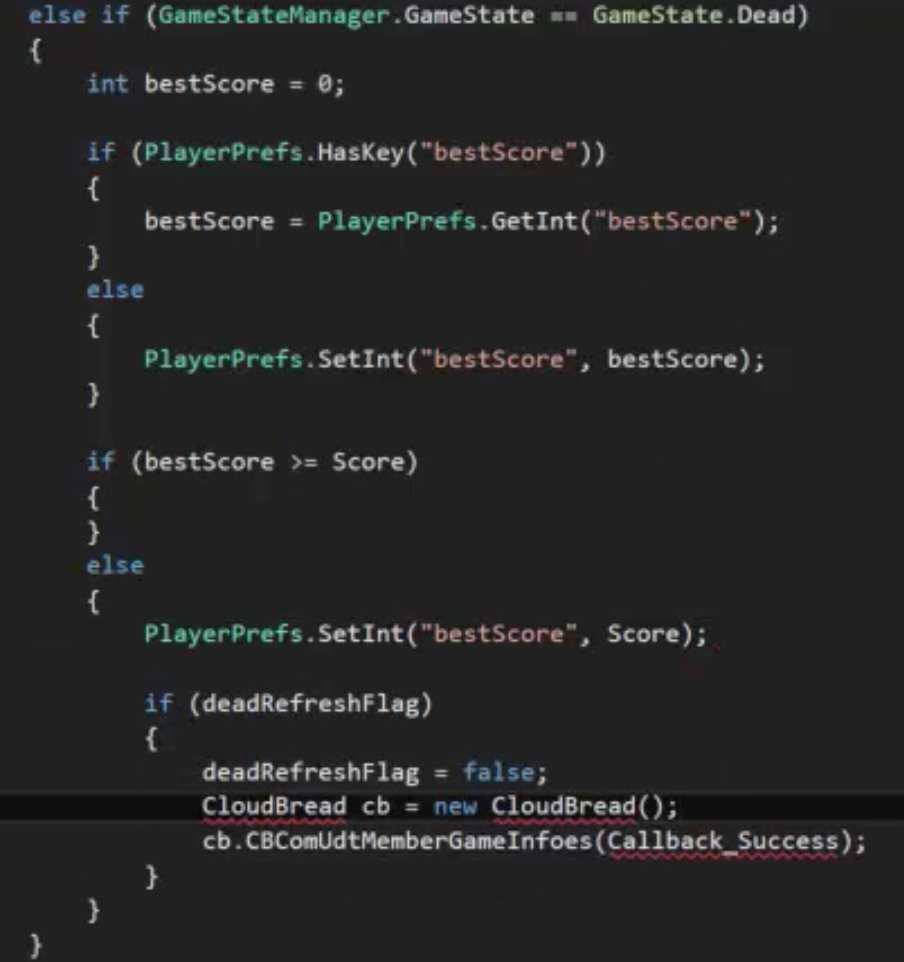
CBInsRegMember API호출 시, 중복된 사용자(이미 가입된 사용자)

|  |
| --- |
|  |

|  |
| --- |
| CloudBread.cs 파일에 , CloudBread 서버에 회원 등록을 하기 위해  CBInsRegMember API 를 호출하도록 구현    FacebookLoginScript에 다음과 같이 함수 수정 및 추가 |

# CloudBread에 게임 기록 저장하기

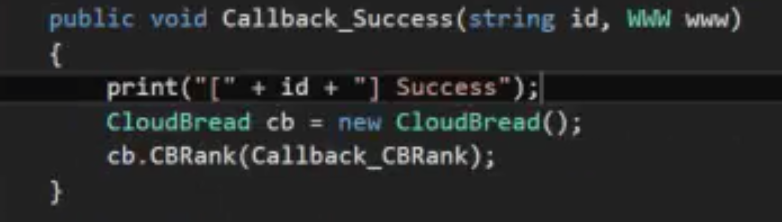
1. ScoreManagerScript.cs 파일 열기
2. 게임 끝났을 때, CBComUdtMmberGameInfoes 호출을 위해,  
   GameState가 Dead 일 때 추가하기



1. CBComUdtMmberGameInfoes API 호출을 위해  
   MemberGameInfo 클래스 생성 (PostMan 참조)
2. CloudBread 클래스에 CBComUDTMemberGameInofes 호출 구현



1. ScoreManagerScript 에서 CBComDUTMemberGameInfoes 호출했을



이 자료는 클라우드브래드 캠프를 위해 제작된 자료입니다.

이 자료는 웹에도 공개할 예정입니다.

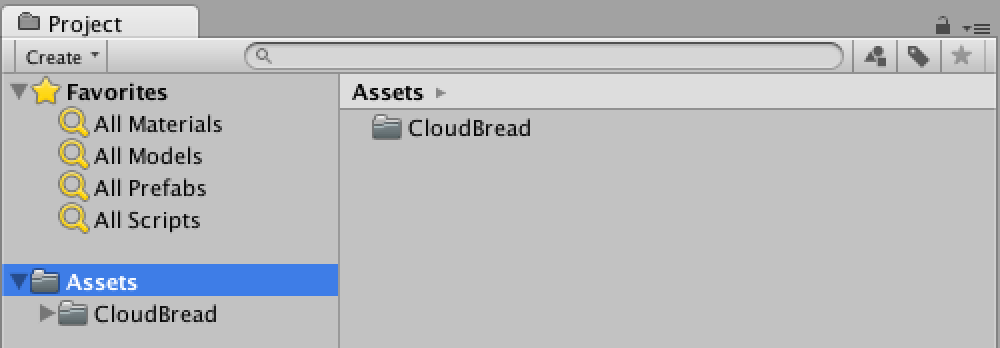
CloudBread-Unity-SDK 사용방법

# 아래 사이트에서 프로젝트 소스코드 다운받기

<https://github.com/CloudBreadProject/CloudBread-Unity-SDK>

# 프로젝트에 추가하기

**CloudBread-Unity-SDK-master – Asset – CloudBread** 폴더 추가하기

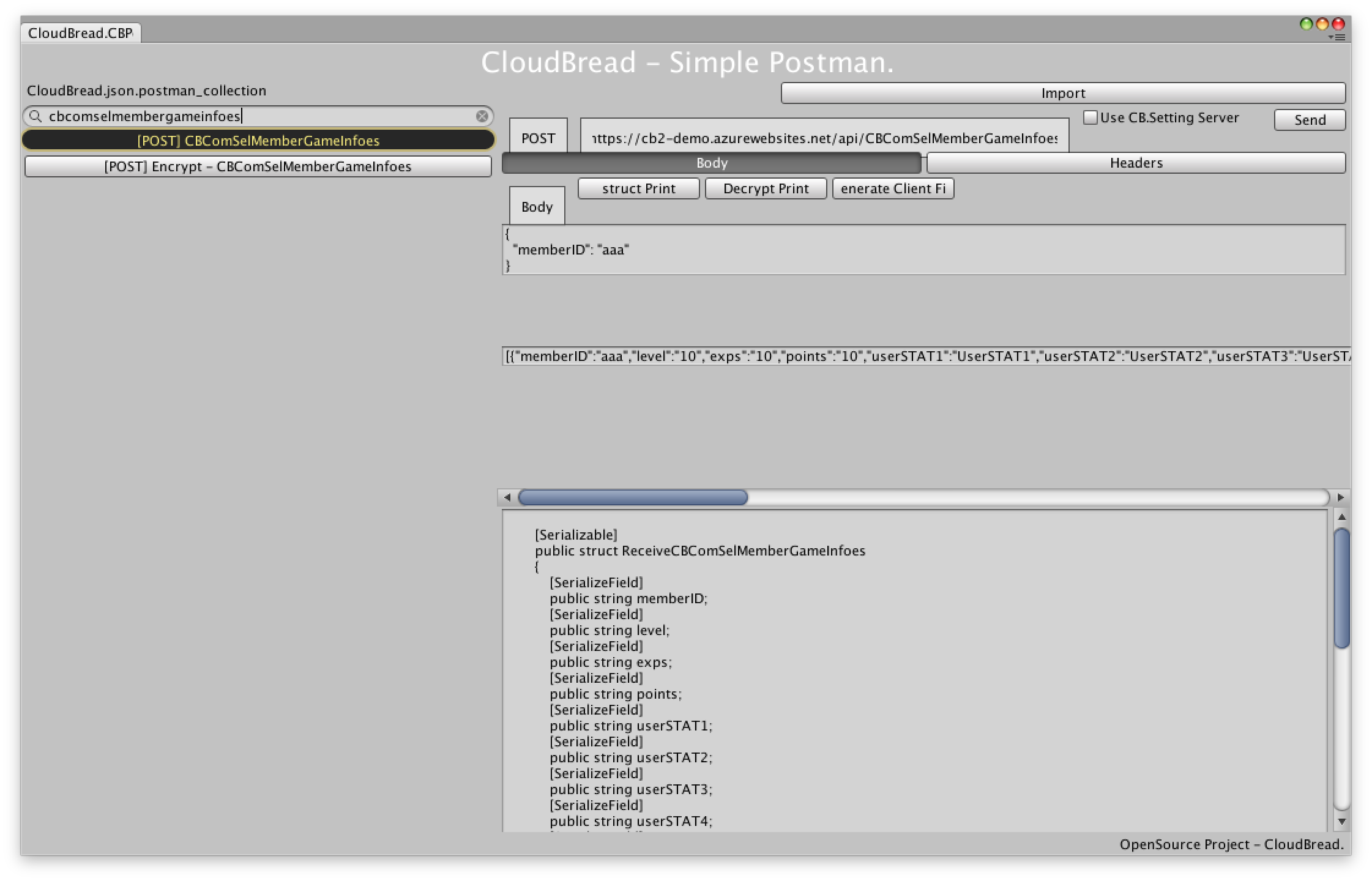




CloudBread 메뉴가 생기는 것을 확인 할 수 있음

# PostMan 파일 추가하기

* **CloudBread – CB-PostMan** 클릭
* 다운받은 PostMan 파일 (.json.postman\_collection )을 Import 시키기

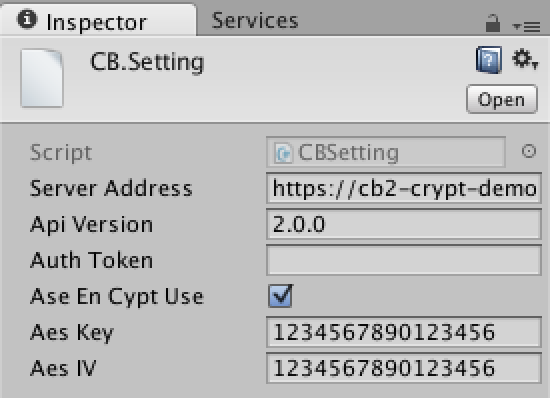


# 호출 하고 싶은 API 클래스 추가하기

위 화면에서 원하는 API 를 검색하고, **enerate Client Fi** 버튼 클릭

# 자신의 CloudBread 서버 주소 변경하기

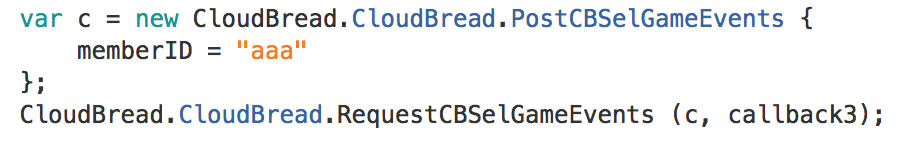
**CloudBread – Resources – CB.Settings** 클릭



여기서 서버 주소와 암호화 키와 사용 여부를 설정 할 수 있음

# 상황에 맞게 필요한 API 호출하기

다음과 같이 Post 할 데이터를 채우고, API를 Request 하면 됨



API Request에 대한 응답을 다음과 같이 콜백으로 받을 수 있음

