**카메라 기능 구현**

NativeCamera라는 카메라 Opensource를 이용

1. CameraUnityPackage 패키지를 Import Package로 Import

패키지 설명:

- Plugins

**mylibrary-debug.aar(android)**: 안드로이드 모듈 플러그인입니다. 서버에 이용됩니다

**CopyThistoManifest(android)**: AndroidManifest에 들어갈 내용입니다. 어플리케이션 권한 관련 내용이 들어가있습니다

**NativeCamera**: 오픈소스 카메라 모듈입니다. 사진 촬영 및 저장에 이용됩니다.

- **GamerScene**: 카메라가 구동되는 Scene입니다. 이 씬을 삭제하고 다른 씬과 합쳐도 무방합니다. 버튼은 두 개가 있는데, Capture 버튼을 누르면 카메라가 호출되고 사진을 찍으면 종료되고 사진이 저장됩니다. (자잘한 버그로 사진 촬영을 몇 번 반복해야 종료될 수도 있습니다) 그 후 Server 버튼을 누르면 TCP/IP 서버로 저장된 이미지가 전송됩니다. Canvas에 CameraServer이 적용되어있으며, 만약 커스텀을 하고싶다면 해당되는 class를 불러오면 됩니다. (Capture: CameraServer.TakePicture / Server: CameraServer.Server)

- Scripts-**CameraServer.cs**: 카메라 및 서버를 총괄하는 클래스입니다. **IP 및 포트를 파이썬 서버와 동일하게 입력해주세요.** 중간에 수정가능이라고 표시한 부분은 촬영한 사진이 미리보기로 일정 시간동안 출력되게 만들어놓은 것인데, 현재 버튼에 가려져서 잘 보이지는 않습니다 ㅠㅠ. 게임에 맞춰서 자유롭게 수정하면 좋을 것 같습니다. 본 스크립트를 Canvas 혹은 다른 요소에 드래그하여 Component로 넣어주세요.

**정상 구동을 위해 추가적으로 반드시 해야 될 것**

1. Build Settings에서 Android로 전환

2. Unity-Build Settings-Player Settings-Publishing Settings-Build-Custom Main Manifest 활성화 후 생성된 Manifest 파일을 “CopyThistoManifest.txt” 파일의 내용을 복사해서 붙여넣기하여 내용을 교체해주세요.

3. Unity-Build Settings-Player Settings-Other Settings-Identification-Minimum API Level를 “Android 7.0 ‘Nougat’ (API Level 24)”로 바꿔주세요. (AAR 파일의 빌드와 같게끔 만들어줌)

4. Unity-Build Settings-Player Settings-Other Settings-Identification-Override Default Package Name 활성화, Package Name을 무조건 “**com.eleven.Unity1**”으로 설정하셔야 됩니다.