Publishing a Unity game to the Mac OS App Store:

Step 1. In Unity, choose “File” > “Build Settings” > Platform “PC, Mac & Linux Standalone” > Choose target platform “Mac OS”.
Click “Player Settings”

In player settings, make sure these five settings are checked:

Default is Full Screen = true
Default is Native Resolution = true
Capture Single Screen = false
Display Resolution Dialog = false
Mac App Store Validation = true

Step 2. Go to the application build that you have just created using Unity. Right click on the build to ‘Show Package Content’. Inside Content folder, edit the Info.plist file. You can use text editors like Sublime Text or Atom to edit the Info.plist.
All the code in the Info.plist is in the form of key and value pairs.
Make sure that:
Add a new <key>LSApplicationCategoryType</key> with values <string>public.app-category.games</string>

Step 3. In your Build folder, create a file GAMENAME.entitlements, with sand-box key like this:

```
<?xml version=”1.0" encoding=”UTF-8"?>
<!DOCTYPE plist PUBLIC “-//Apple//DTD PLIST 1.0//EN” “http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version=”1.0">
<dict> <key>com.apple.security.app-sandbox</key> <true/> </dict> </plist>
```

Step 4. Assume you have already paid and got an Apple developer account. Visit https://developer.apple.com/account/ , then go to certificates, choose the dropdown with “Mac OS X” and click the “+” button for creating new certificate. You will need to go through the process twice to get type of “Mac App Distribution” and “Mac Installer Distribution”. Save them to key chain, which will be use later.
Next, go to “Identifiers” and choose “App IDs” tab, create an application id which matches the bundle id of the application in unity. For example: com.companyname.gamename
idmsa.apple.com
Sign In - Apple
Sign in with your Apple ID

Step 5. Login to App Store Connect and create a new application by using MyApps > Click on "+" > New Mac App

Step 6. Fill in all the required information such as name of the app, screenshots of the application, etc etc

Step 7. Download and install the latest Application Loader to submit the app. There are couple to steps to go through first:

Step 8: Sign the app with the created document in step three.

```
codesign -f -s '3rd Party Mac Developer Application: DEVELOPER NAME' --entitlements "GAMENAME.entitlements" "/AppPath/GAMENAME.app"
```

You can find the 3rd Party Mac Developer Application DEVELOPER NAME by going to keychain access and searching for 3rd Party Mac Developer Application. In step 4: You get 2 different certificates which you add to the Keychain access: Mac App Distribution and Mac Installer Distribution". Mac App Distribution is used here. Click on it to find the DEVELOPER NAME value.

However, you’re likely to get the error:

```
code object is not signed at all
```

This is because the subcomponent is not signed. You could sign the subcomponents one by one, including libmono.0.dylib and libMonoPosixHelper.dylib etc, but easier way just use the deep sign command, like this:

```
codesign -f -s '3rd Party Mac Developer Application: DEVELOPER NAME' --entitlements "GAME.entitlements" “GAMENAME.app" --deep
```

Step 9: Next build the .pkg file:

```
productbuild --component GAMENAME.app /Applications --sign "3rd Party Mac Developer Installer: DEVELOPER NAME" GAMENAME.pkg
```

Step 10: Remove any existing game.app in your machine then you can verify the install by:

```
sudo installer -store -pkg GAMENAME.pkg -target /
```

Step 11: Finally, open the Application Loader and ‘Deliver Your App’ by choosing the GAMENAME.pkg. If things go well, then you should be able to upload it. Otherwise fix the specific errors accordingly. It will takes a while to go through.
Also note, that if you want to upload it again to the App Store, and get an error stating that you need to upgrade the build version, just remember that in the info.plist, "Bundle Version String Short" is the version number and the "Bundle Version" is the build number