AR Playhouse Simulator Deploy Guide

By Kieren McCord | kieren.mccord@asu.edu
Last Updated: 16 September 2022

Overview: This guide walks you through the steps to not only download and deploy the latest version of the AR Playhouse Construction Simulator , but also enables you to immediately receive and deploy any updates. While specific to this application, most steps can be adapted to apply to deploying any Unity->HoloLens application. The guide assumes you are working on a Windows computer. The process is very similar for a Mac, but some of the deployment steps require running a Windows emulator. There are four major steps in the proces, each associated with a specific software or hardware and are as follows:

1. **GitHub:** Getting access to the project repository and updates via GitHub Desktop
2. **Unity**: Opening the Project in Unity via Unity Hub and exporting the solution
3. **Visual Studio:** Opening the solution in Visual Studio 2019
4. **HoloLens 2:** Deploying the solution to the HoloLens 2

1. GitHub Desktop: Git-ing started

a. **Download** GitHub Desktop here
   i. If you don't have a GitHub account, **create** one
   ii. **Open** GitHub Desktop
   iii. **Select File->Clone Repository** (Figure 1)

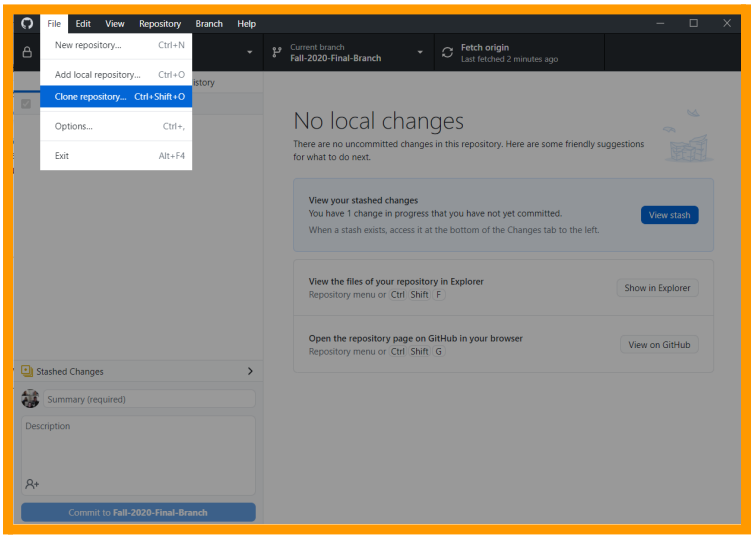

Figure 1. File->Clone repository within GitHub Desktop

   iv. **Click** URL and enter the following in the prompt box:
      1. https://github.com/ETBIMlab/AR-Playhouse-Design-Construct

2. (For a blank template where you can use your own model, navigate to: https://github.com/ETBIMlab/AR-Builder-Template)

v. **Select** the repository entitled AR-Playhouse-Design-Construct (see Figure 2)

vi. **Navigate** to the filepath on your computer where you want this stored (See Figure 2)

vii. **Click** 'Clone' (See Figure 2)

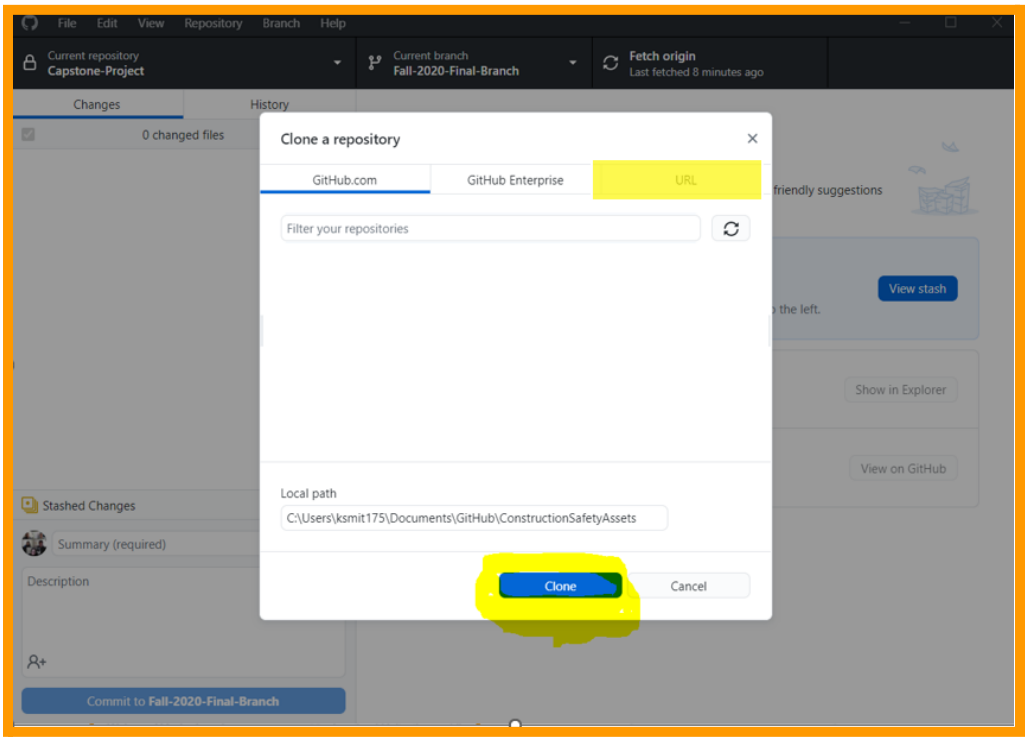

Figure 2. Entering the repository URL and beginning clone

viii. When the repository is successfully cloned, make sure your Current repository is AR-Playhouse-Design-Construct and **navigate** to Fall-2020-Final-Branch as your Current branch (see Figure 3).

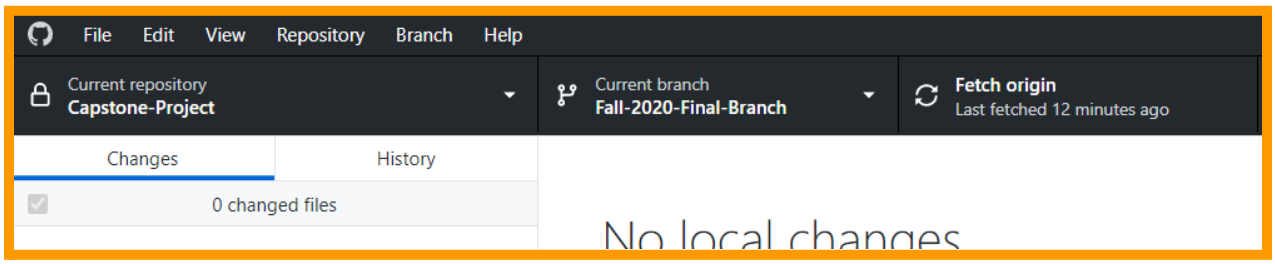

Figure 3. Navigating to the correct branch (your screen should match this figure)

ix. **Click 'Fetch origin'** (Figure 3) to ensure you have the latest changes. You will use this whenever you are deploying to ensure you have the latest version.

x. You are now ready to open the project in Unity.

2. Unity Hub: Opening the Project

a. **Download** Unity Hub or Unity from here. My preference is Unity Hub, which lets you toggle between versions.
   i. *(Unity Hub enables you to easily work on multiple unity projects. If you anticipate never using unity again, just downloading a single version is fine)*.
   ii. Whether you use Unity Hub or just one instance of Unity, make sure you include **version 2019.3.9f1** and that you include build support for **Universal Windows Platform.**

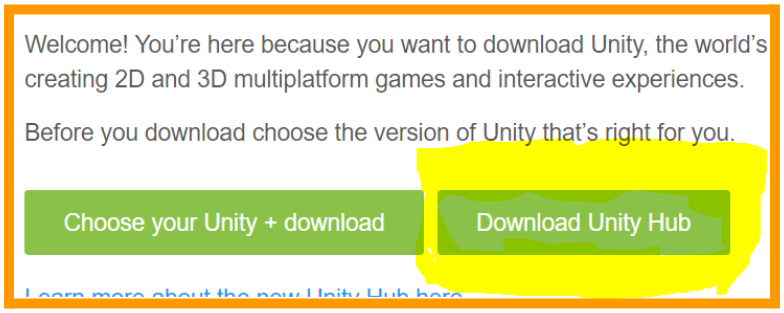

Figure 4. Screenshot of download button for Unity Hub

b. Open Unity Hub, make sure you are in the 'Projects' tab, and **click 'Add'** (See Figure 5).

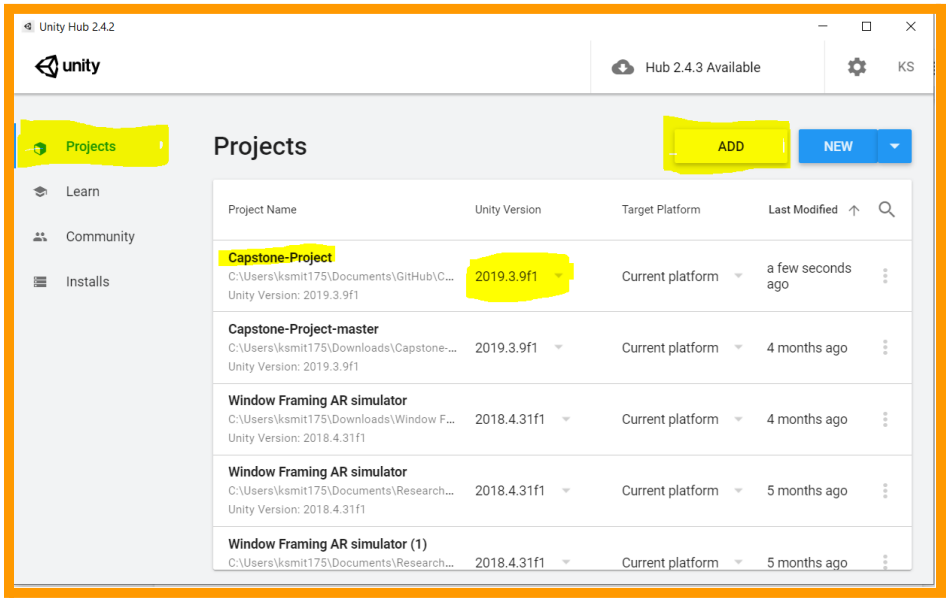

Figure 5. Navigating to 'Projects' folder and clicking 'ADD' to add the project, then selecting the correct project and version.

c. When you have the project added, make sure the 'Unity Version' is 2019.3.9f1 (See Figure 5) then **click on the project title** to open the project.
d. **Wait** for the project to load (this can take several minutes and it may seem like the program closed)
e. When opened, your project should look something like Figure 6. If you don't see the playhouse in the scene, navigate to Assets->Scenes->Summer2021 within the file navigation pane at the bottom of the screen.

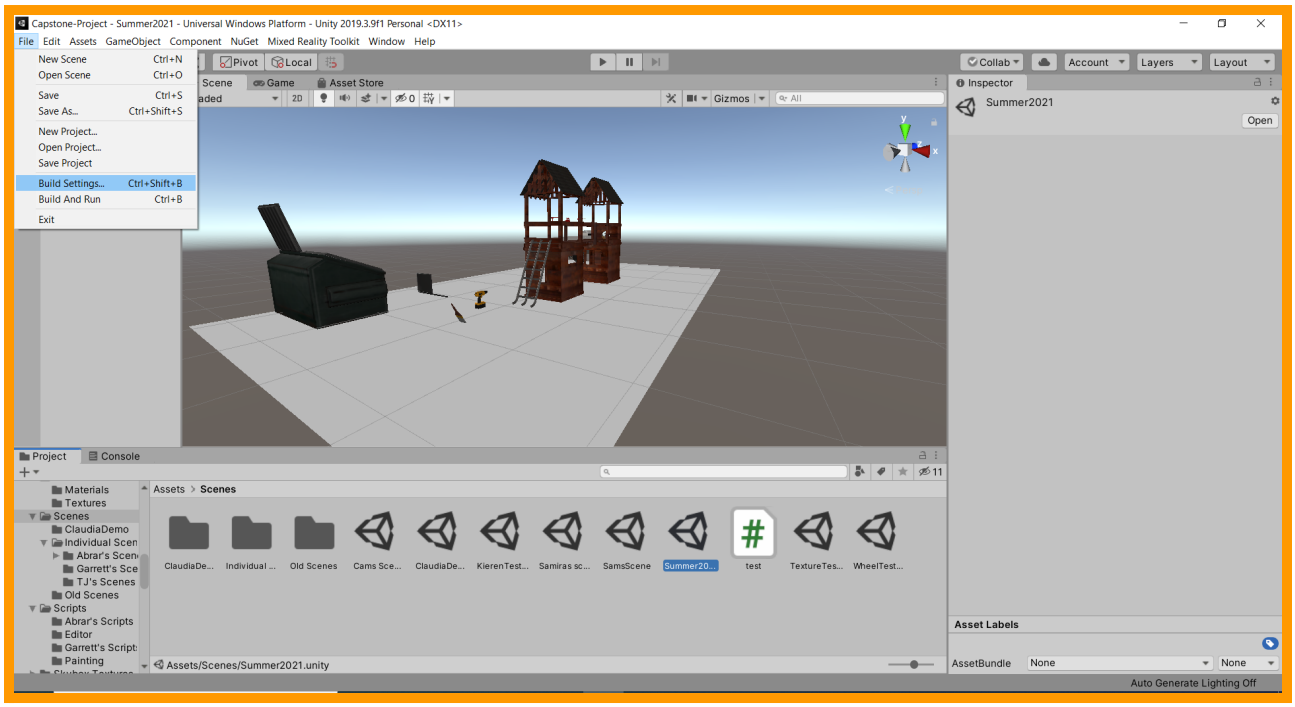

Figure 6. Navigating to the build settings within the project

f. **Navigate** within Unity to File-> Build settings (see Figure 6).
g. **Match** your build settings to Figure 7
    i. The current (June 22, 2021) scene we are working on is Summer2021. In the future, it could be updated, so feel free to reach out and double check which scene is the latest.
        1. For future updates, you may have to use the 'Add Open Scenes' button (Figure 7) if there is a new scene since you last deployed

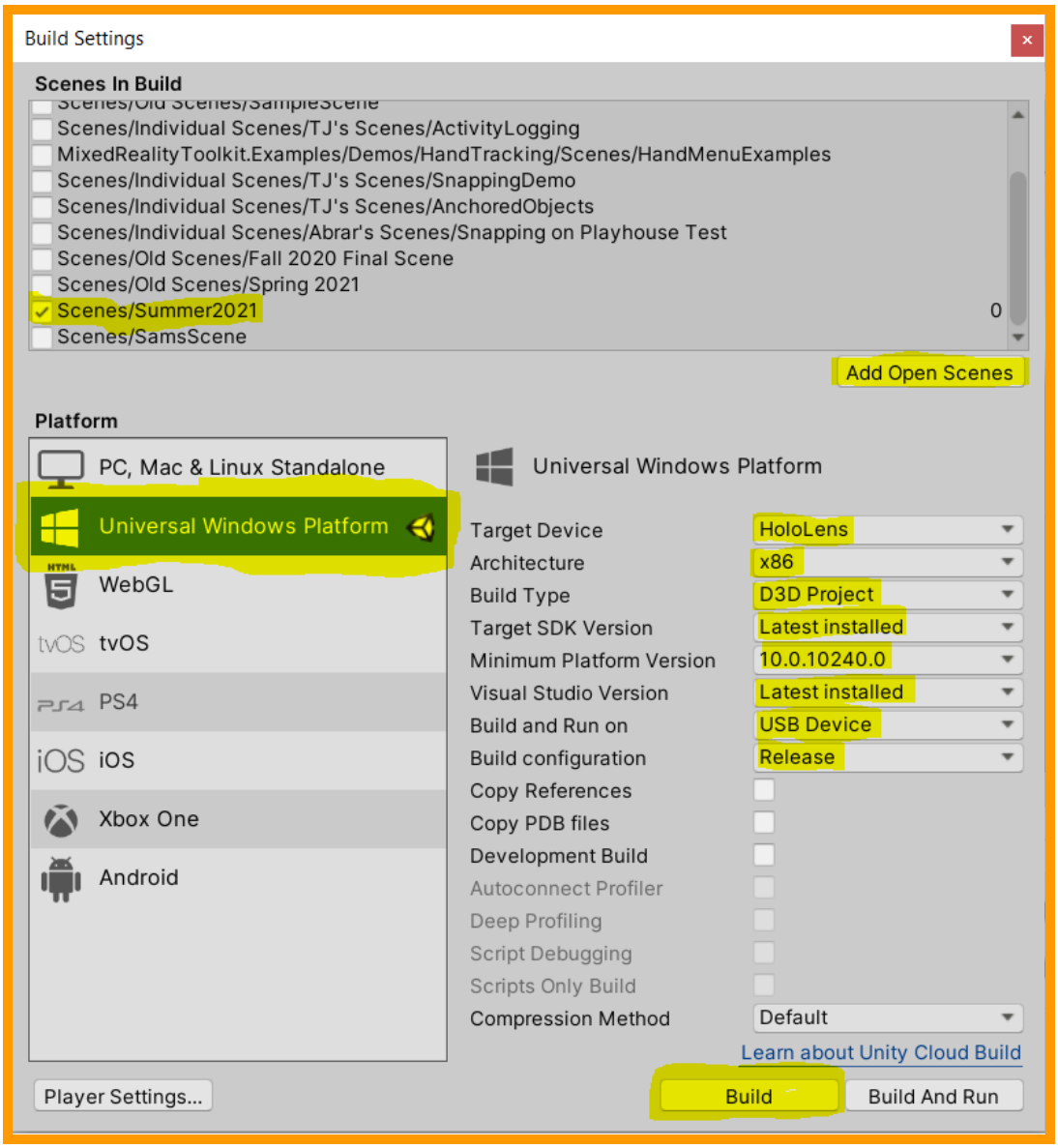

Figure 7. Build settings within Unity

h. **Press** 'Build' and navigate to the file location where you want the solution stored (this can be a new folder that you create anywhere).

3. Visual Studio: Open and Build Solution

a. When the solution is fully built, it should automatically open the folder where it is located. If not, navigate to the location you chose in step 2h (See Figure 8).

b. To open this file with Visual Studio 2019, first **install** Visual Studio 2019 (if you don't already have it) by downloading it here.
   i. The Community Free download works great for our needs on this project.

c. Next, go back to the folder with the solution, right click, and choose to **open with Visual Studio 2019** (Figure 8).

i. You can just double click the solution if all you have installed is Visual Studio 2019, but if you have previous versions of Visual Studio, it will default to those.

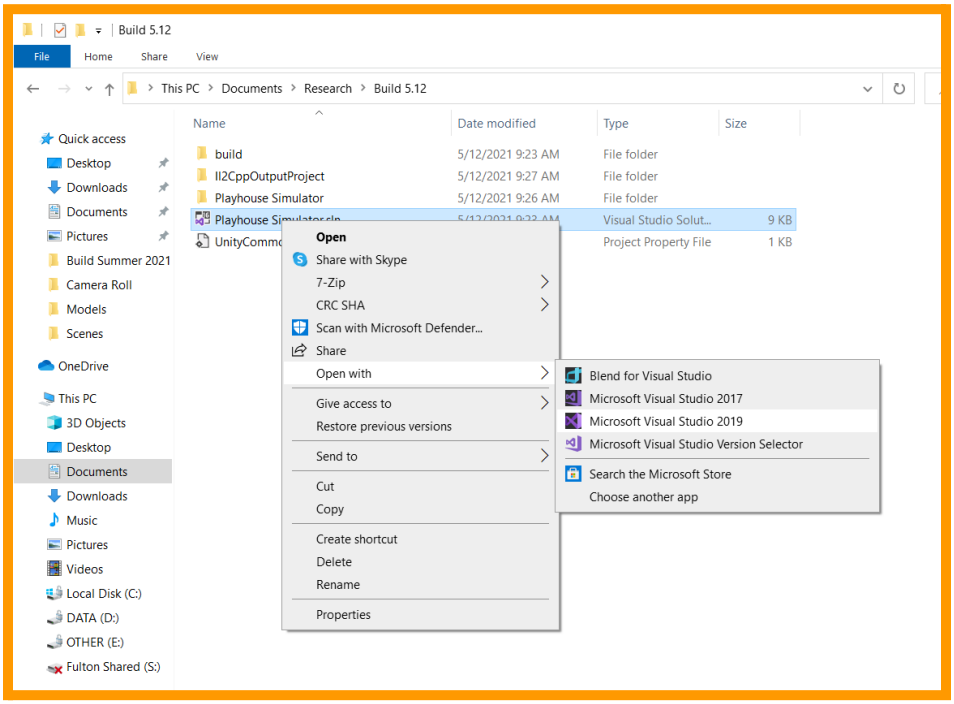

Figure 8. Opening Visual Studio solution file (.sln) within file explorer

d. **Match** your settings to the Figure 9 (indicated by the arrows)

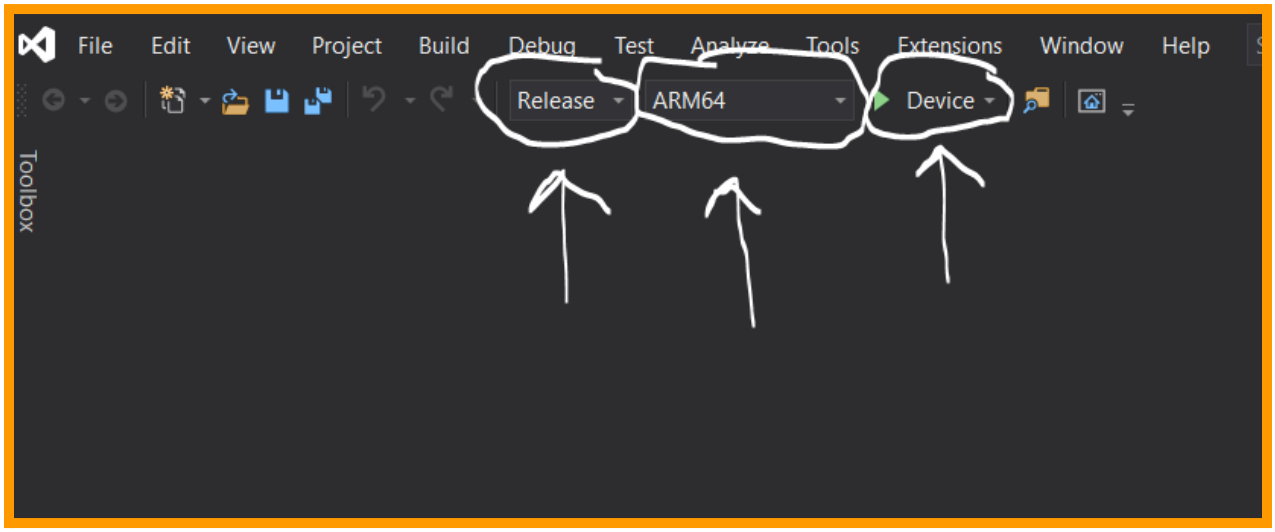

Figure 9. Build settings within Visual Studio

e. **Click Build->Build Solution** or Ctrl+Shift+B

f. After a few minutes, the Output window should tell you if your build was successful and your screen should look like Figure 10. This means the solution is built and is ready to be sent to the HoloLens

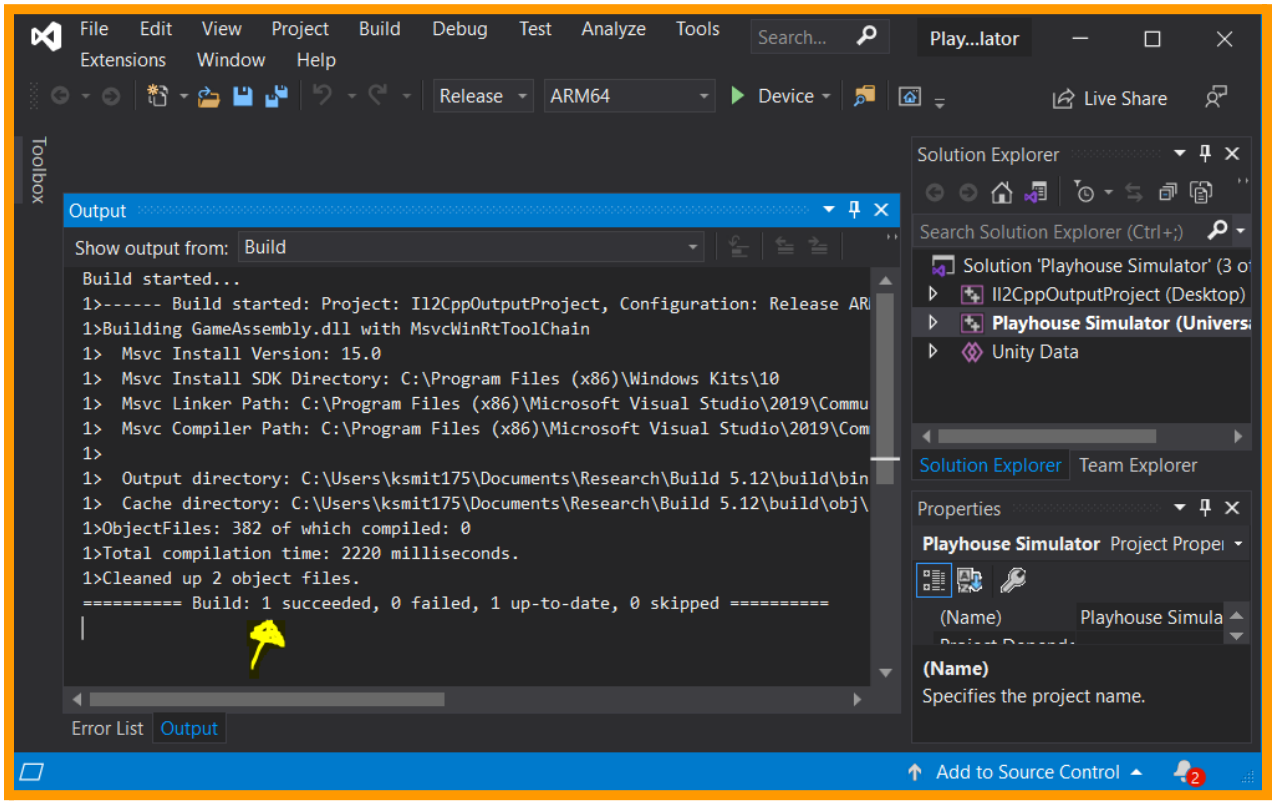

Figure 10. Successful Build Indication

g.

4. HoloLens 2: Deploy Application

a. This step involves both Visual Studio and the actual HoloLens
b. This guide assumes that you have already set up the HoloLens and know how to navigate within the device and only mentions relevant settings for development. If you have questions regarding using the device, see this documentation from Microsoft for assistance.
c. On the Hololens, go to **Settings-> Update & Security -> For developers**
    i. Make sure 'Use developer features' is **On** and 'Device Discovery' is **On** (see Figure 11)

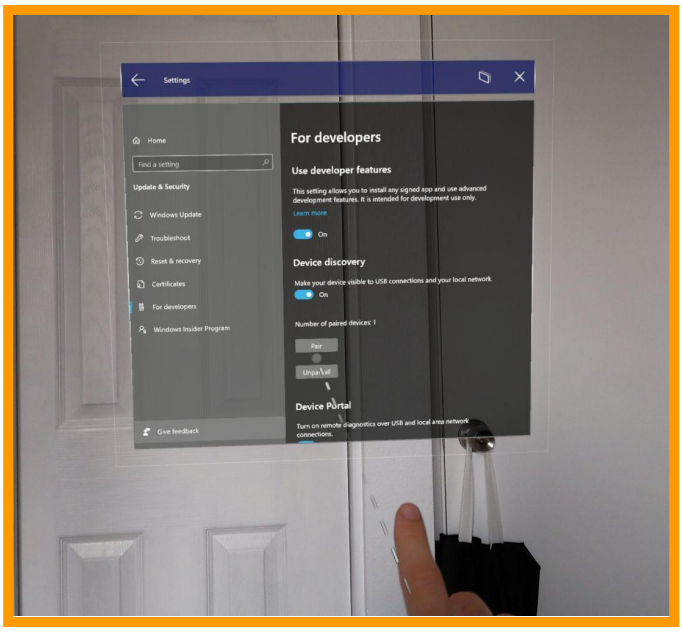

Figure 11. Demonstration of Developer features on and Device discovery on (blue buttons), and the 'Pair' button about to be selected by the cursor

ii. It is also helpful to turn the 'Device Portal' **On** (This enables screen casting later)

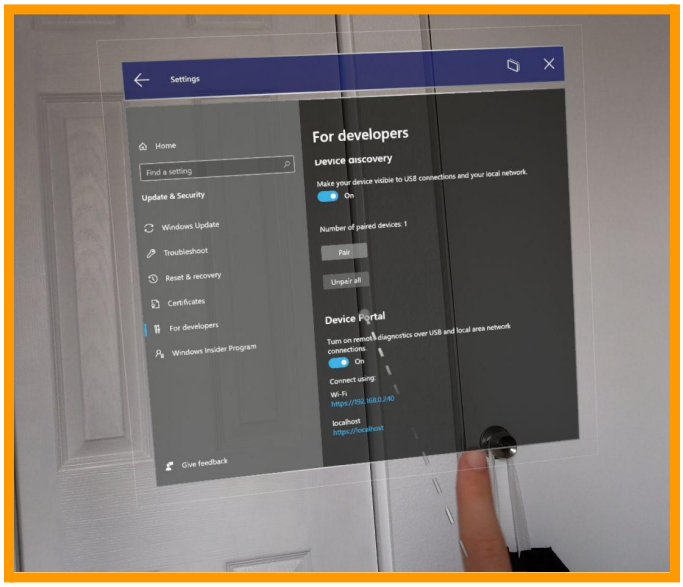

Figure 12. Demonstration of Device portal on (same screen as Figure 11, but scrolled down)

d. **Plug** the HoloLens into the computer you are using.
e. **Click 'Pair'** on that same screen within the HoloLens, and note the PIN that is provided.
f. In Visual Studio, select **Build->Deploy Solution**.
g. When prompted for a PIN, **enter the PIN** from 4e.

h. After a bit of time, you should see a message similar to figure 10, but it will say Deploy: 1 succeeded.
    i. Troubleshooting tips: If you see 'boostrap error' or 'device failed to connect' or similar errors, try the following (in any order)
        1. **Unplug and replug** in the HoloLens to the computer while Visual Studio solution is open. Try to deploy again.
        2. Leave the device plugged in and **close then open** the solution again in Visual Studio. Try to deploy again.
        3. **Unpair** all devices within the HoloLens(same place as where you pair-Figure 11) then try to deploy again. (It will prompt you for another PIN, click pair again-the PIN changes each time)
i. Within the headset, **navigate** to Playhouse Simulator through the main menu->apps->all apps, open, and you are set to go!
    i. (Use the voice command 'Set Space' after placing the cylinder on the floor to make the playhouse appear)
j. Let me know if you have any questions!