Contents

[HoloBasket - Developing your First Holographic App with Unity 2](#_Toc472726886)

[Holograms and Gaze 2](#_Toc472726887)

[Configuring the environment 2](#_Toc472726888)

[Preparing a Scene for Holographic Content 9](#_Toc472726889)

[Positioning the Holograms 9](#_Toc472726890)

[Gaze 11](#_Toc472726891)

[Managers 12](#_Toc472726892)

[Colliders 12](#_Toc472726893)

[Build and Deploy 13](#_Toc472726894)

[Spatial Mapping and Gesture 16](#_Toc472726895)

[Spatial Sound 21](#_Toc472726896)

# HoloBasket - Developing your First Holographic App with Unity

To develop our first HoloLens app we will use Unity 3D and Visual Studio.

Make sure you installed all necessary tools as described at the Windows Developer Center <https://developer.microsoft.com/en-us/windows/holographic/install_the_tools>

Our Sample is based on Unity 5.5.0f3 and the HoloToolkit.

The app we will develop consists in a basket which can be freely moved in our environment and to which we can throw balls at, trying to score. We will position holograms in our real world, add the possibility to interact with them through Gaze, Gestures and voice commands. We will also add spatial sound. and we will use spatial mapping to make our holograms be aware of the world around us and behave as they would really be part of it.

Alright, let’s start.

# Holograms and Gaze

## Configuring the environment

The first thing to do is to launch Unity. If you never used it before, you will need to log in and create a new project by clicking on “New” as shown in Fig. 1.

Fig. 1 New Project

In the next screen (Fig. 2), as “Project name” enter “HoloBasket”, as location assign C:\ (Unity will automatically create for us the HoloBasket folder) and leave the rest unchanged.

Fig. 2 Assigning a project name

By clicking on the “Create Project” button you will create the project.

At this point it will be necessary to configure the project with the assets you will need in the development phase. First of all, you will need to install HoloToolkit, so go to GitHub and donload the latest version <https://github.com/Microsoft/HoloToolkit-Unity> . One you downloaded it and you extracted from the zip archive, go to Assets and select and copy the content as shown in (Fig. 3).

Fig. 3 Copying selected contet of the Assets folder

You have now to past the contet in your C:\HoloBasket\Assets folder

Going back to Unity, you may notice that the editor is configuring the HoloToolkit (this operation may require some time to complete).

Fig. 4 Unity configures HoloToolkit

Once this process is complete, you will notice a new item, “HoloToolkit”, in the menu bar.

Fig. 5 The new menu bar item, HoloToolkit

Now, from the HoloToolkit menu, select:

Configure🡪Apply HoloLens Scene Settings

then

Configure 🡪 Apply HoloLens Capability Setting

Last but not lease

Configure 🡪 Apply HoloLens Project Settings

The editor will first ask whether you want to force text serialization of assets, to which you should answer “Yes” (Fig. 6). Then, it will ask whether you want to reload the project (Fig. 7). Also in this case you will need to answer “Yes”.

Fig. 6 force text serialization of assets

Fig. 7 Reload Now

With this procedure, you first configured the Main Camera so to make it render as background the color black (which in HoloLens represents the transparent color) and then we configured the project so that it yields a Windows Store App as output.

Save the scene, File 🡪 Save Scene by naming it “Basket”.

Now let’s go to File 🡪 Build settings

Click on Add Open Scenes

Tick the Unity C# Project box

Click on Player Settings (Fig. 8)

Fig. 8 Build Settings

This will open, in the Inspector panel, the Player Settings (Fig. 9)

Fig. 9 Player Settings in the Inspector panel

Here all the work has already be done HoloToolkit already, you can just have a look, for example, at the Capabilities inside the Publishing Settings and you will notice that needed capabilities are alredy setup

At this point you only need to check the Quality Settings. From the menu bar, select Edit🡪 Project Settings 🡪 Quality

The Quality Settings will open in the Inspector panel, but also here HoloToolkit has already configured everything, namely the quality level, on fastest.

All we need to do now is start developing our project.

## Preparing a Scene for Holographic Content

Create a new Scene: File -> New Scene

Remove the default Main Camera and Directional Light objects in the scene.

Add the HoloLensCamera.prefab (found under HoloToolkit/Input/Prefabs).

To do so, in the lower part of the Project toolbar, select Assets🡪HoloToolkit🡪Input🡪Prefabs, on the right hand side select HoloLensCamera and drag and drop it to the right in an empty area of the Hierarchy panel

Add the DefaultCursor, in the lower part of the Project toolbar, select Assets🡪HoloToolkit🡪Input🡪Prefabs🡪Cursor, on the right hand side select DefaultCursor and drag and drop it to the right in an empty area of the Hierarchy panel.

## Positioning the Holograms

Going back to the path where you donloaded this repo you will find a folder named “0BaseAssets”, inside this folder you will find the 3D objects, you will need to copy the file Hoop.3ds and paste it in the folder: C:\HoloBasket\Assets

Back in Unity, you can notice how in the Project panel, in the Assets folder, our basket is already there to be used (Fig. 10).

Fig. 10 The basket in the Assets folder

Right-click in an empty area of the Hierarchy panel, on the left-hand side, and select Create Empty (Fig. 11).

Right-click on the GameObject just created, select Rename and change the object name to HologramCollection.

Fig. 11. Hierarchy panel

In the Assets panel, select the object Hoop and drag and drop it into the Hierarchy panel, exactly over the GameObject HologramCollection, so that Hoop is added as child item of this object, as shown in Fig. 12.

If during dragging and dropping you accidentally placed Hoop in a different position on the Hierarchy panel, you will need to delete it (by selecting it and pressing Delete on the keyboard) and start over with drag and drop.

Fig. 12 Hoop as child item of HologramCollection

Once placed Hoop as child item of HologramCollection in the Hierarchy panel, select HologramCollection and go to the Inspector panel on the left.

Let’s move our object so that it appears in front of the user when he wears the HoloLens and launches the app. To do so, after selecting HologramCollection in the Hierarchy panel, change the Position values in the Inspector panel by setting X to 0, Y to -0.5 and Z to 2 (as shown in Fig. 13).

Fig. 13 Modified position for HologramCollection

This way, our hologram will be positioned 0.5 meters below and 2 meters in front of the user’s head (when the application is launched).

Set Rotation so that X and Z will be 0 and Y to 180

Now select Hoop in the Hierarchy panel and change the Scale values in the Inspector panel by setting X, Y and Z to 0.3, so that our model will have the right proportions once represented in HoloLens.

Fig. 14 Hoop Scale

## Gaze

At this point you can start adding some interactive functions to your hologram by showing a cursor which will represent the Gaze, the direction of your head when wearing HoloLens.

First you will need to add a cursor to your scene (Basket), so that the cursor will show the user where he is pointing his Gaze.

To do so, in the lower part of the Project toolbar, select Assets🡪HoloToolkit🡪Input🡪Prefabs🡪Cursor, on the right hand side select Cursor and drag and drop it to the right in an empty area of the Hierarchy panel (Fig. 15).

Fig. 15 Hierarchy panel with Cursor

If during dragging and dropping you accidentally placed Cursor in a different position, you will need to delete it (by selecting it and pressing Delete on the keyboard) and start over with drag and drop.

## Managers

Right-click in an empty area of the Hierarchy panel on the left, and select Create Empty (Fig. 11).

Right-click on the GameObject just created, select Rename and change the object name to Managers.

In the Assets panel below, select Assets🡪HoloToolkit🡪Input🡪Prefabs, then on the right select InputManager and drag and drop it on the right in the Hierarchy panel so that the script is added as child item of Managers (Fig. 16).

Create an empty object in your scene and make sure its transform is zeroed on the origin. Rename it 'Managers'.

Add the InputManager.prefab (found under HoloToolkit/Input/Prefabs) as a child to your new 'Managers' Object.

Add an Event System to your scene by right click on 'Managers' object in your scene Hierarchy: UI -> Event System.

Fig. 16 Managers and GazeManager

If during dragging and dropping you accidentally placed GazeManager (inside the Hierarchy panel) in a different position, you will need to delete it (by selecting it and pressing Delete on the keyboard) and start over with drag and drop.

## Colliders

As last step, let’s add some interactivity to the object. To do so, go to the Project toolbar, select Assets🡪Hoop and in the Inspector panel tick the box Generate Colliders.

Fig. 17 Generate Colliders

## Build and Deploy

Thanks to HoloToolkit, it is now extremely easy to create the package necessary to deploy and launch the application on HoloLens (connected to the PC on which you are working through a USB connection).

In the HoloToolKit menu, select Build Window, fill in the username and password fields with the appropriate values and click on Build SLN, Build APPX, then Install, as shown in Fig. 18.

Fig. 18 Build and deploy on HoloLens

And all done. Just a few seconds for the APPX package to be created and deployed on your device and the app that you just developed will appear on your HoloLens.

From the start menu, Gaze on the HoloBasket icon (Fig. 19), then Air Tap. The application will launch.

Fig. 19 Gaze on HoloBasket

Once the application is launched, you will see the basket. If you move your Gaze from the basket to a non-virtual object you will see the cursor change shape and color, therefore everything done so far works perfectly.

Obviously, as the spatial mapping as well as all the functions to move and position the basket have not yet been added, it will not be possible to move it yet. We will get there.

Fig. 20 The basket as it appears on HoloLens

# Spatial Mapping and Gesture

Sure, it is a nice basket, but it would be even nicer if I could move it around in the room and place it wherever I want: on the table or the floor. Let’s see how to do that.

First of all you will need to add a sort of collider to the basket so to provide a surface Gaze and Air Tap can interact with.

In the Hierarchy panel on the left hand side, expand HologramCollection and select Hoop (Fig. 21).

On the right hand side, click on the Add Component button (Fig. 21) and in the search box type the text “box”, then select Box Collider to add it to the Inspector as shown in the Inspector panel.

Fig. 21 Box Collider

Modify the Box Collider properties as represented in Fig.22, by setting Center X to -0.7, Center Y to 0.5, Center Z to -1, Size X to 1.5, Size Y to 1 and Size Z to 2.

Fig. 22 Box Collider Properties

In the Assets panel, expand SpatialMapping 🡪 Prefabs, on the right hand side select the object SpatialMapping and drag and drop it in an empty area of the Hierarchy panel, as shown in Fig.23.

Fig. 23 Hierarchy with SpatialMapping

If during dragging and dropping you accidentally placed SpatialMapping (inside the Hierarchy panel) in a different position, you will need to delete it (by selecting it and pressing Delete on the keyboard) and start over with drag and drop.

In Hierarchy, select Hoop and on the right, in the Inspector panel, click on the Add Component button. In the search box, type “han” and select the Hand Draggable script to add it to the Inspector toolbar as shown in Fig. 24.

Fig. 24 Adding the TapToPlace script

To check whether the basket can now be moved around, it is necessary to repeat the app’s deployment on HoloLens, as explained in the section Build and Deploy.

Hooray, if you now move the Gaze to the base of the basket and Air Tap on it, you will be able to move the basket to any other place in the room.

Let’s now turn our attention to the ball so that we can shoot some hoops.

# Gesture

Go to C:\0BaseAssets\ folder and copy Shooter.cs. Paste it in the folder C:\HoloBasket\Assets

In the folder Tools\3D Models of the CD copy the file Basketball.3DS and paste it in the folder C:\HoloBasket\Assets

In the Assets panel right-click on an empty area, select Create 🡪 Physic Material and name it “Bouncy”, as shown in Fig. 25.

Fig. 25 Create a new Physics material

In the Inspector panel set properties for Bouncy as shown in Fig. 26

Fig. 26 Bouncy - Settings

Go to C:\0BaseAssets\ folder and copy Basketball.3DS. Paste it in the folder C:\HoloBasket\Assets

Back in Unity, In the Assets panel, select the “Basketball” object and drag and drop it in an empty area of the Hierarchy panel. Select the object just created and drag and drop it in the Assets panel again so to create a prefab object. Then select the Basketball object in the Hierarchy panel and delete it.

At the end of the procedure, both Hierarchy panel and Assets panel will have to look as in Fig. 27

Fig. 27 Creating the prefab object Basketball

Select, in the Assets panel, the prefab object Basketball and set Scale X, Y and Z to 0.0018, then untick the box Receive Shadows as shown in Fig. 27.

In the Inspector panel, click on the Add Component button, in the search box type “ri” and select Rigidbody so that it gets added to the Inspector toolbar as shown in Fig. 28.

In the Inspector panel, click on the Add Component button, in the search box type “sph” and select Sphere Collider so that it gets added to the Inspector toolbar as shown in Fig. 28.

Fig. 28 Inspector panel for the prefab object Basketball

Once added the Sphere Collider it will be necessary to choose Bouncy as Material.

This way the ball, besides having gravity, will also be able to bounce.

In the Hierarchy panel select HoloLensCamera. On the right, in the Inspector panel, click on Add Component (Fig. 29) and in the search box type “sho”, select the script Shooter to add it to the Inspector as shown in Fig. 29. In addition to this, it is necessary to connect our ball to the Shooter script so that at every Air Tap event a ball is shot.

In the Assets panel select the prefab object Basket and drag and drop it in the Prefab property of the Shooter script, as shown in Fig. 29.

Fig. 29 Adding Shooter to Main Camera

To check whether our basket can now move, it is necessary to redo the app deployment on HoloLens, as explained in the section Build and Deploy.

# Spatial Sound

In the menu bar, select Edit🡪Project settings 🡪 Audio

In the Inspector panel, on the right, set the property Spatializer Plugin to MS HRTF Spatializer

In the CD, in the Tools folder, copy BallBounce.wav and paste it in the folder C:\HoloBasket\Assets

In the Assets panel select the object BallBounce and drag and drop it over the prefab object BasketBall, as shown in Fig. 30.

Now, if in the Assets panel you select the prefab object BasketBall, you will find, in the Inspector panel on the left, the audio source BallBounce.

Here you will need to tick the box Spatialize and set the value of Spatial Blend to 1 (3D) as shown in Fig. 30.

Fig. 30 Adding Spatial Sound to the basketball

Our work is now done.

Re-execute the deployment of the app on HoloLens, as explained in the section Build and Deploy, and have fun with HoloBasket.