3D SYSTEMS

OpenHaptics®

Unity Plugin

User Guide

Original Instructions

TABLE OF CONTENTS

Introduction

The purpose of the OpenHaptics Unity plugin is to enable users to seamlessly add Haptics (the ability of touch) to their virtual world. This plugin integrates the Unity 3D components with the haptic world there by creating a wholesome experience for the user. This plugin works in harmony with the Touch and Touch X devices and is meant to fill a void in the 3D and VR space by allowing for the perception and manipulation of objects using the senses of touch and proprioception.

It is imperative that users and/or programmers have at least an intermediate knowledge of Unity and Openhaptics to derive the maximum benefit from this plugin.

Copyright

1 INSTALLING THE PLUGIN

Prerequisite

In order to Install the new OpenHaptics Unity Plugin, the following steps must be taken:

1. Download the latest Touch device drivers and configure the Haptic Devices to be used with the plugin (Use the Touch Smart setup or Touch Setup to configure device profiles).
   a. Access the download link for latest drivers: 2018.07.31: http://support1.geomagic.com/Support/5605/5668/Article/View/3242
   b. Click on “Downloads”, then click on “Haptic Device Drivers”
   c. Scroll down to Product : OpenHaptics
   d. Ensure that the OS is listed as: WIN 10, 7 or 8.1 and Platform : 64 bit (only)
   e. Click on “Download” version : 2018.10.22

Version	Driver	Download
v2019	**Touch Device Driver v2018.10.22:** *for Ethernet or USB Touch devices (Interface:)* OS: 7/8/10 (64-bit)	Download

2. After installing the Touch Device Drivers, the user must install Openhaptics 3.5 developer version software from the link: https://3dssupport.microsoftcrmportals.com/knowledgebase/article/KA-01460/en-us
3. The user must have at the minimum the Unity Beta software (free version).
   a. Download the Unity plugin from the following location: https://unity3d.com/get-unity/download
   b. Unity2017 64 bit required. (Tested with Unity2017.3.1f1 – 64 bit)
   c. Operating System: Windows (PC Only) – 64 bit

Once the above pre-requisites are met, the user is ready to install and access the OpenHaptics Unity Plugin.

Note: This plugin does not support Intel graphics cards. The Unity Plugin does not work with version 3.4.0, and only works with version 3.5.0.

Configuring the Devices (Creating Profiles)

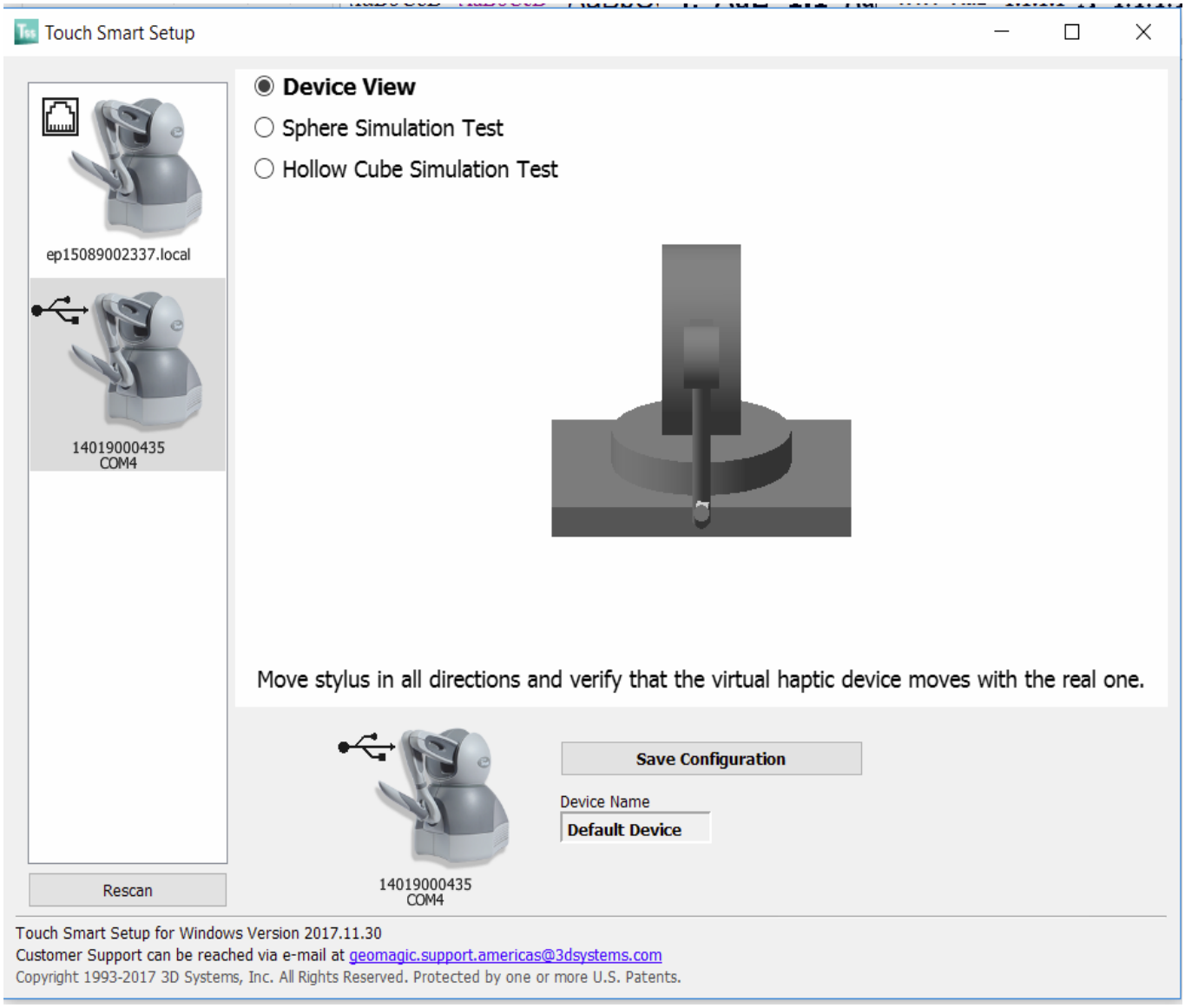
After downloading and installing the latest Touch Device Drivers, the next step is to set up the Haptic devices. Connect devices as needed by performing the following steps:

1. Open the Touch Smart Setup application from the desktop, or from the following location: C:\Program Files\3D Systems\Touch Device Drivers
2. Select either Single or Dual device mode as needed
3. Follow the instructions in the application as prompted and set up the device(s) accordingly. For example: if using a LAN device the device will need to be paired prior to operation.
4. Create Configuration (Config) names by setting up and calibrating the devices.

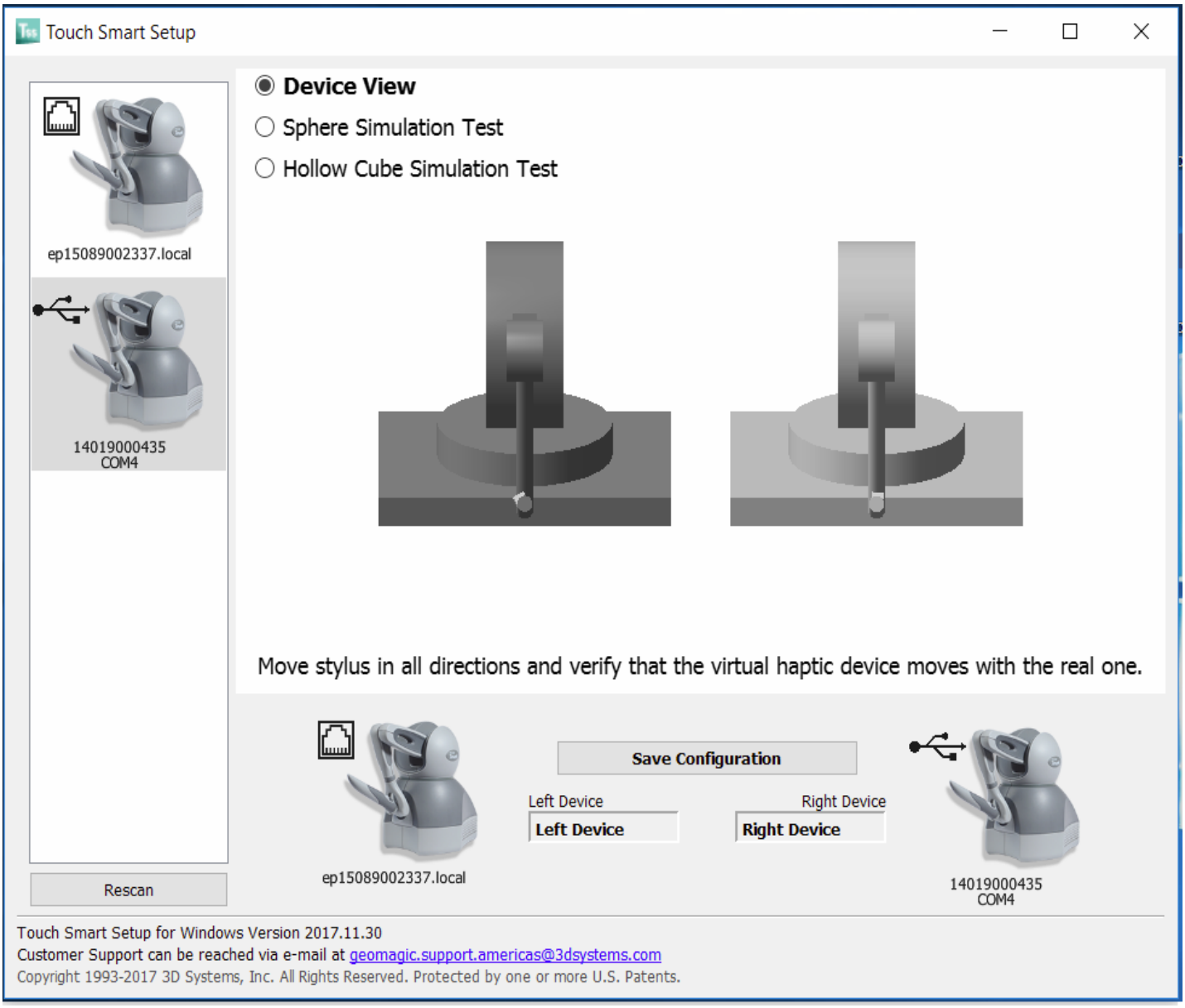
The following profile or configuration names are created as a result:

- Single device: Default Device
- Two devices: Left Device and Right Device

Single Device



Two Devices



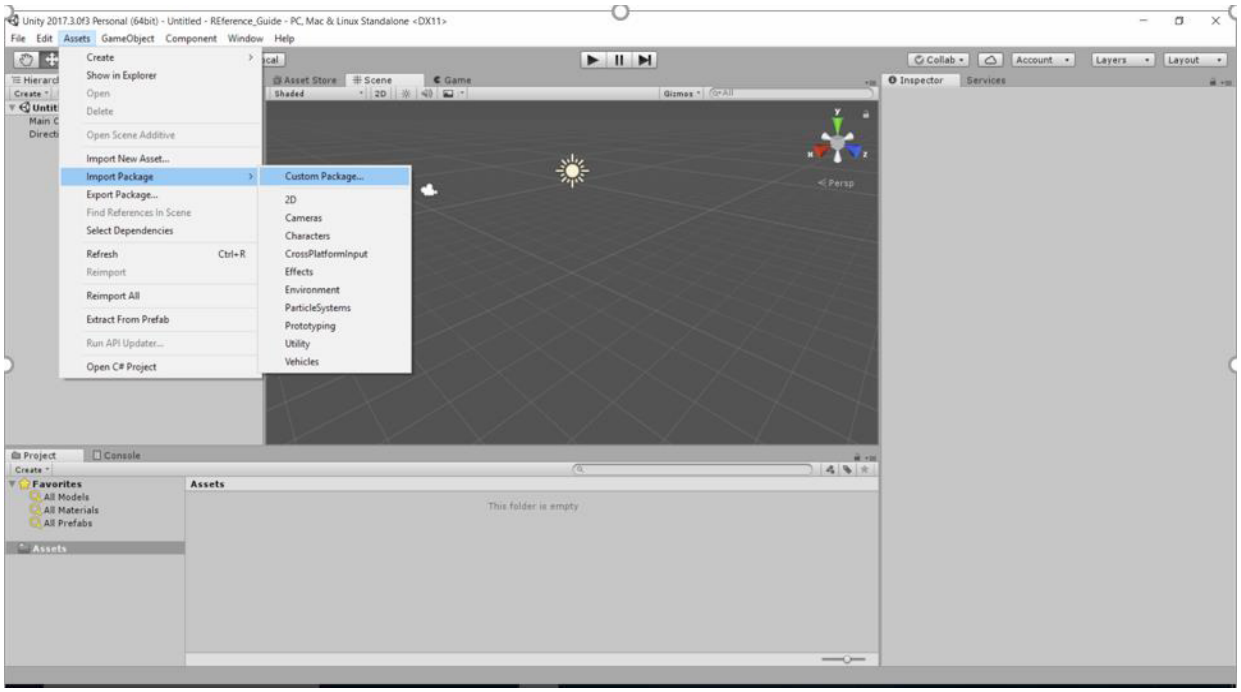
Note: Use the Touch Setup application and create custom profile names. The Touch Setup application can be accessed from the following locationn: C:\Program Files\3D Systems\Touch Device Drivers Select Touch Setup.

Tip: After creating a custom profile using the Touch Setup application, Users must **Calibrate** the devices using the Touch Diagnostic application found in the same location as the Touch Setup application.

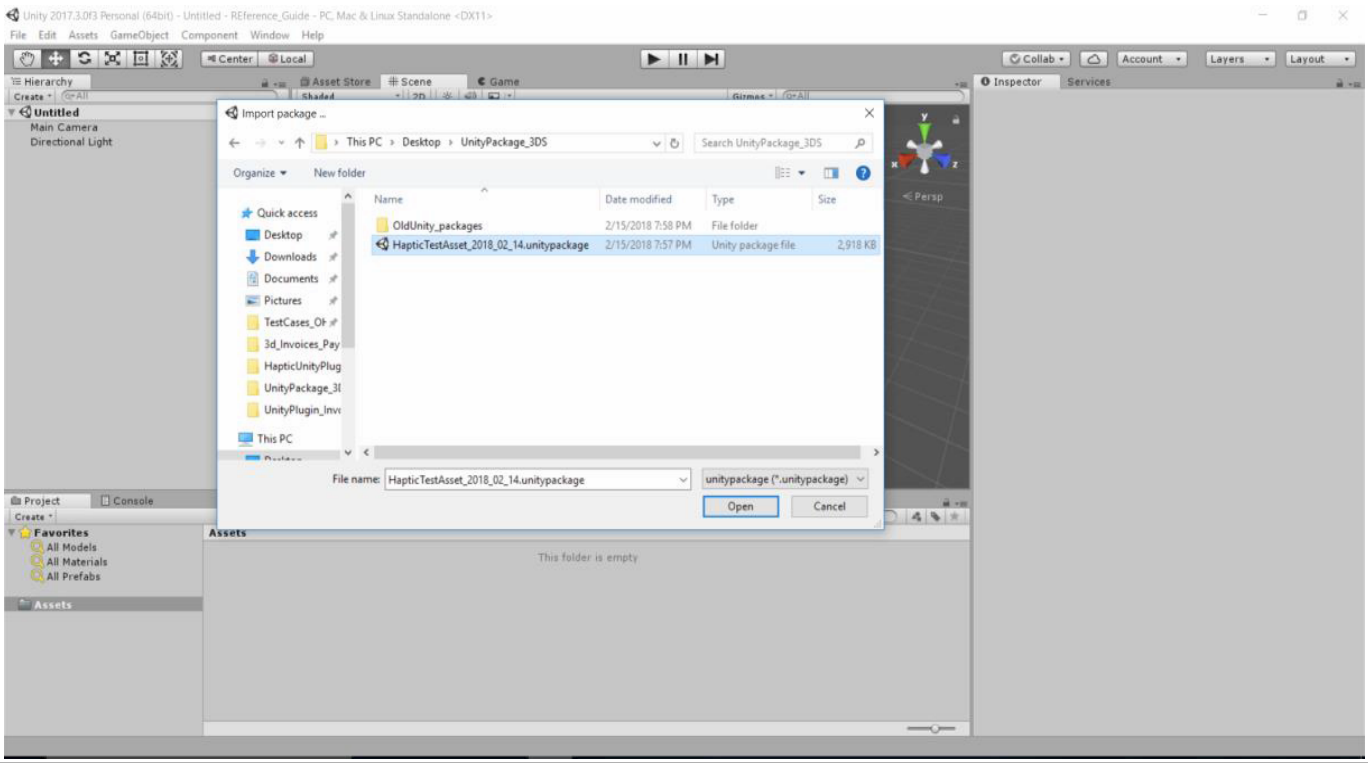
Plugin Installation

Perform the following steps to install the Unity plugin:

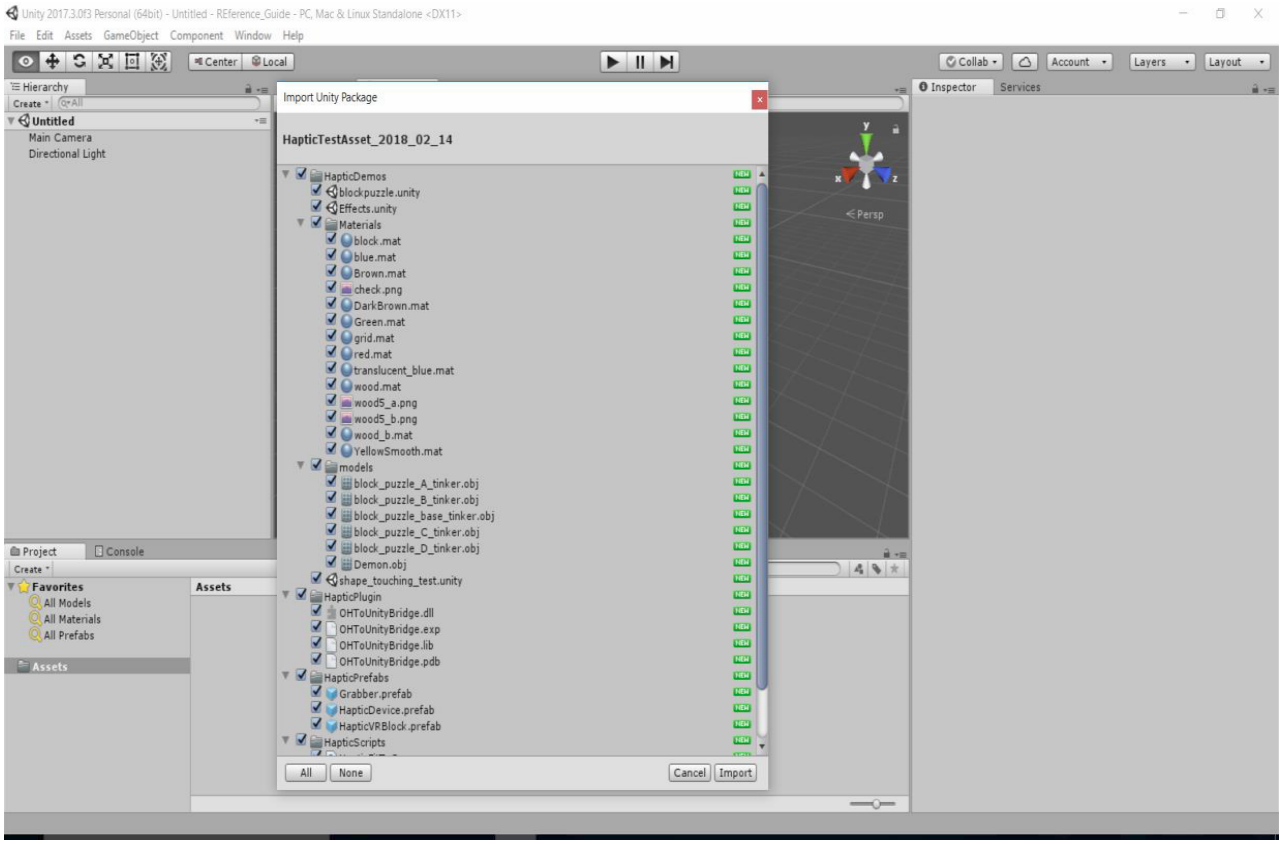
1. Download the OpenHaptics Unity package from the following link: https://3dsystems.sharefile.com/d-s7e02b9133f045a4a
2. Unzip the downloaded archive named **OpenHaptics Unity Plugin Beta**
3. Import the plugin into Unity:
    a. Open Unity and create a **New Scene.**
    b. Import the 3DS OpenHaptics Unity plugin by selecting and importing Unity package from the saved location.
    c. Select **Assets tab->Import Package-> Custom Package-> HapticTestAsset_2018_09_24** (refer to the image below).



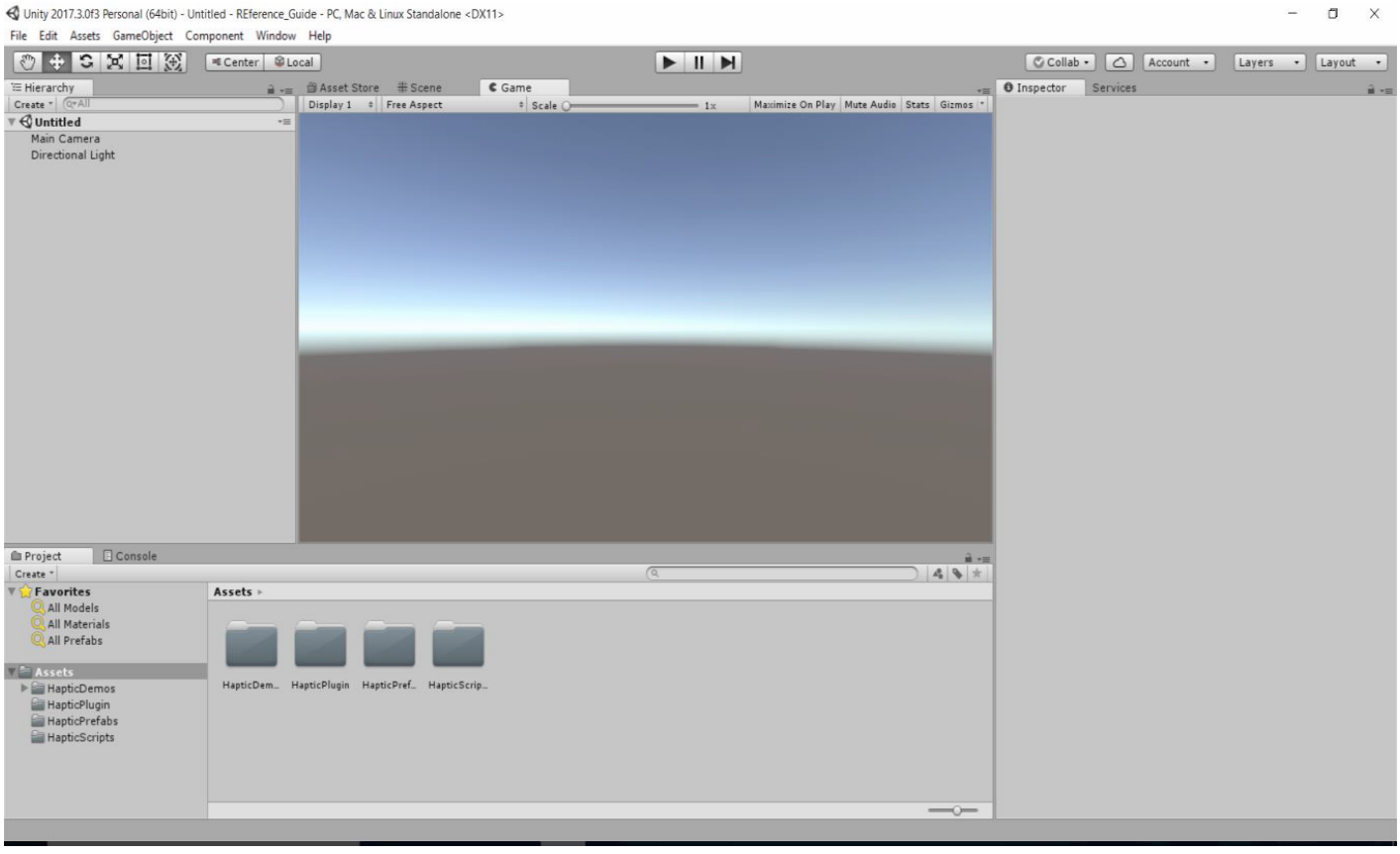
    d. Select the asset package from the downloaded location.



e. Click the option for **Open**. All components of the package are listed for Import.



f. Click **All**, then click **Import**. All the components of the plugin are imported and the components are displayed.



2 UNDERSTANDING COMPONENTS OF THE PLUGIN

There are four major components of the OpenHaptics Unity plugin:

- Haptic Demos
- Haptic Plugin
- Haptic Prefabs
- Haptic Scripts

These are outlined below.

Haptic Demos

The **Haptic Demo** folder contains all the sample demos that were developed using the plugin. The folder also hosts four sub folders, namely:

- **Demo Scripts** – Contains scripts used in the sample demos
- **Materials** – Sample materials used in the demos which can also be used by the user to create new or customize existing demos.
- **Models** – Contains sample 3D models used in the demos
- **UI** – Has sample UI objects used in the sample demos.

These folders serve as repositories to host the components used to create the pre-fabricated demos. This allows for a seamless user experience when interacting with the plugin and also enables user to add components or scripts as they develop their custom demos.

Haptic Plugin

This folder encompasses the .dll file which acts as the bridge between Unity and OpenHaptics (OH) exposing the various functions of OpenHaptics.

Haptic Prefabs

This particular folder is extremely important as it contains the prefabricated haptic device with the **Grabber**. This prefab, as the name suggests, is a pre-programmed setting linking the haptic device to the demos. The user can directly incorporate the haptic device into their demos by adding this prefab.

There is n additional sub-folder titled **More Prefabs.** This folder gives the user option to only add the pre-fabricated haptic device and create their own custom cursor (explained in later sections) to touch and feel objects added in the scene. There is also an option to add a haptic device pre-fabricated with VR option (HapticVRBlock).

Haptic Scripts

This section contains the various haptic scripts used in the demos, as well as others that can be added by the user to create their own custom haptic experience.

Teaching by Example: Adding Haptics to Unity

3 CHAPTER ONE - HELLO HAPTICS DEMO

This demo explains how to add a haptic device to a Unity scene and outlines the variety of notable settings available when adding a haptic device. The following chapters will also include tips and “hacks” to help the user achieve maximum utilization of the utility.

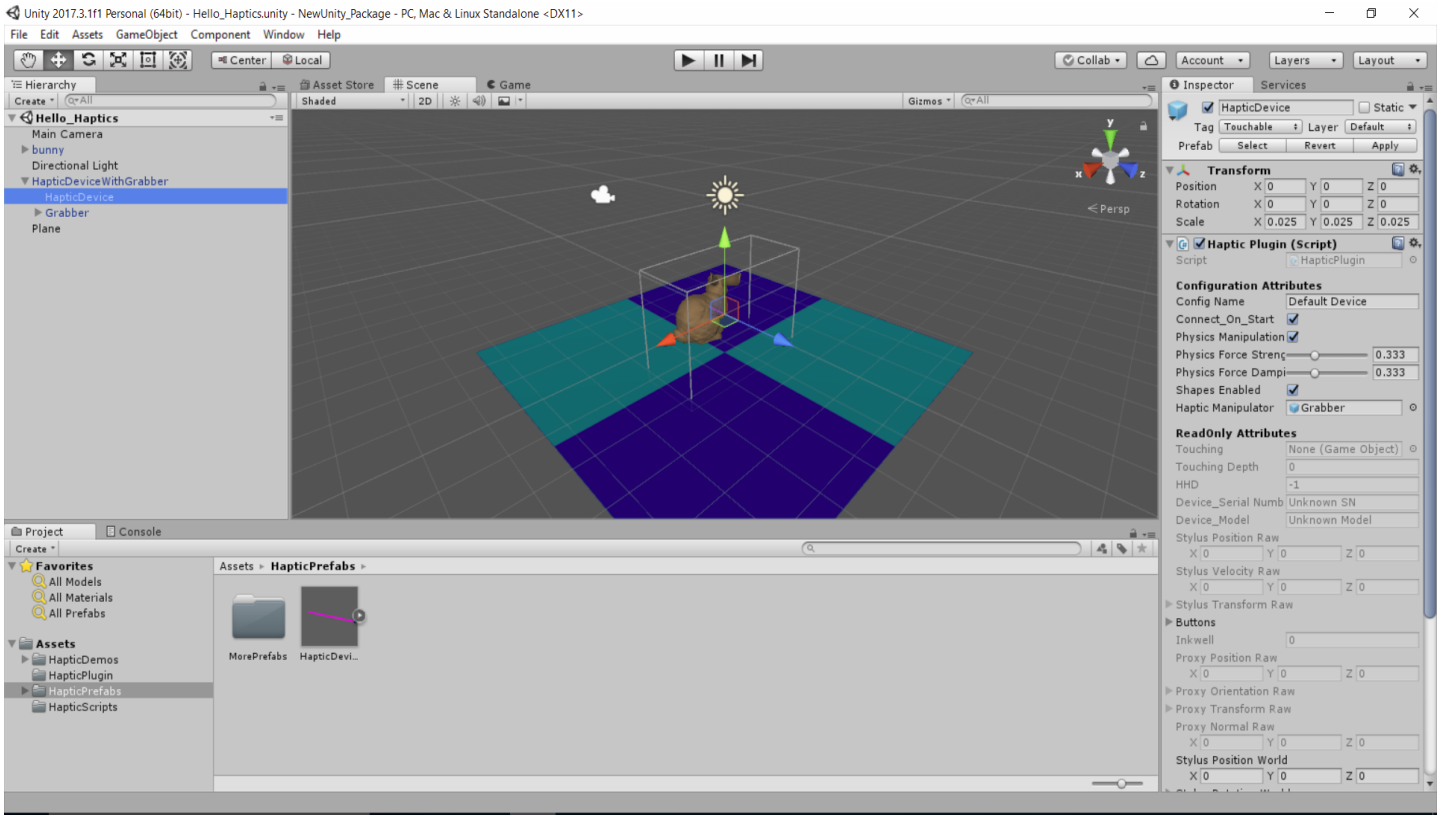
This demo can be found under **Assets -> Haptic Demos Folder -> Choose “Hello_Haptics”**

Double-click on the demo to open it and view the various components and objects it contains. The demo consists of the following elements:

- Main Camera
- Directional Light
- Haptic Device with Grabber
- Bunny (3D object)
- Plane (3D object) The Directional Light and Main camera are added by unity by default to every new scene.

HapticDeviceWithGrabber: This prefab must be added to every scene for the user to add haptics to their Unity demos. This object can be found in the **Haptic Prefabs** folder. This prefab contains two parts:

1. **Haptic Device**: This contains the components needed to integrate the haptic hardware in the demo. The properties of this prefab can be seen in the **Inspector** window.



Important Properties:

Config Name: This field by default is set to **Default Device**. The user can change this after setting up an appropriate profile using the Touch Smart Setup or the Touch Setup application. Note that this field is case sensitive. The **Config** names must match the ones created using the driver apps. Refer to the Prerequisite section for information on setting up config names.

Haptic Manipulator: The **Haptic Manipulator** can be compared to a mouse cursor. It helps user to navigate and touch 3D objects. Users will only be able to touch 3D objects if the appropriate manipulator is specified.

Users can create custom Haptic Manipulators (henceforth referred as Haptic Cursors) and link them to the **Haptic Manipulator** field.

Tip: It is always recommended to add the **HapticDevicewithgrabber** or haptic device – found under **Haptic Prefab -** to the scene before adding the 3D objects. This gives the user a clear understanding of the available haptic workspace and proportionally scales objects within available haptic workspace.

3D Objects Properties:

Bunny Object: This object can be found in the **Haptic Demos** folder under **Models**. The properties of the Bunny object can be viewed in the **Inspector** window after selecting the Bunny object. Important properties to be noted are:

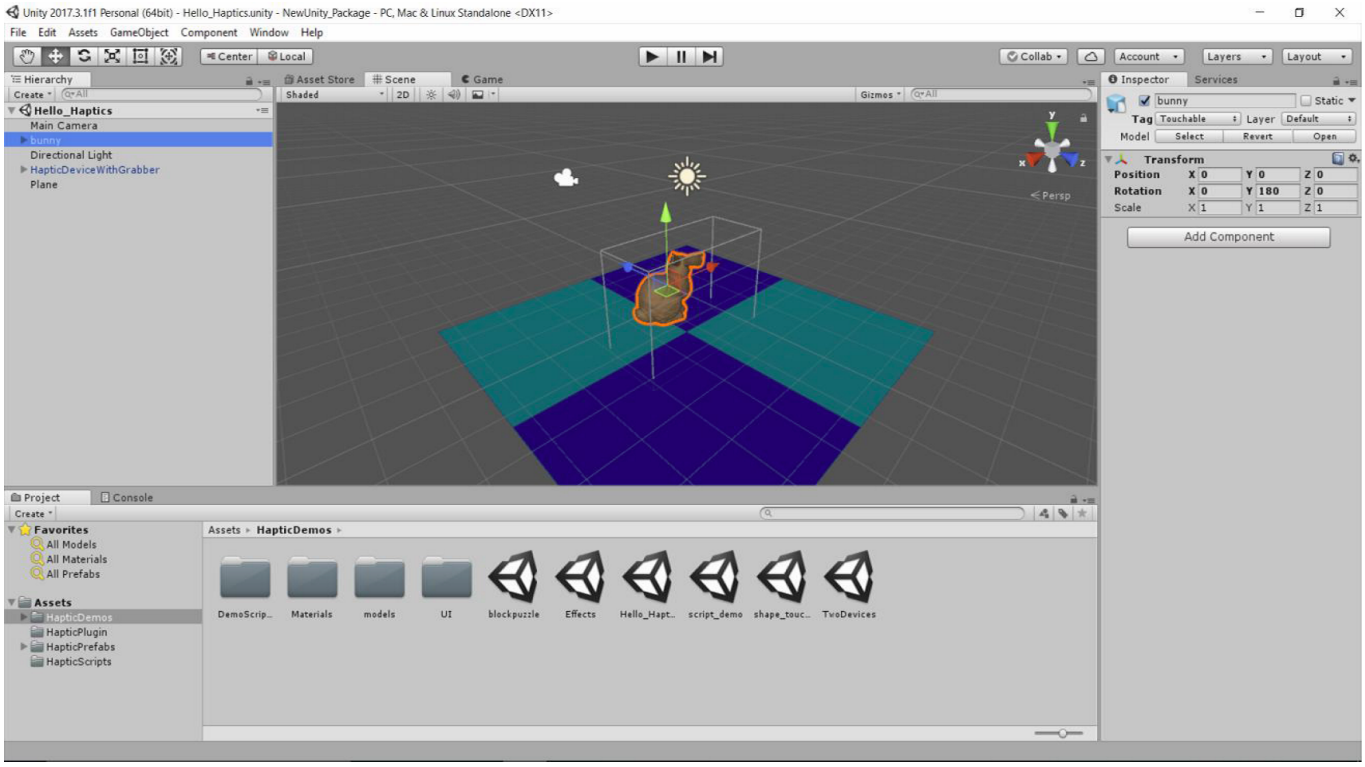
- **Tag: Touchable:** In order to touch any 3D object, the object must have the **Tag: Touchable** setting. This is true for Unity provided 3D objects which can be found under **gameobject->3D object**. Other custom 3D objects can be imported and placed under the **Haptic Demo -> Models** folder.

The **Plane** object can be found under **Gameobject->3DObject**.

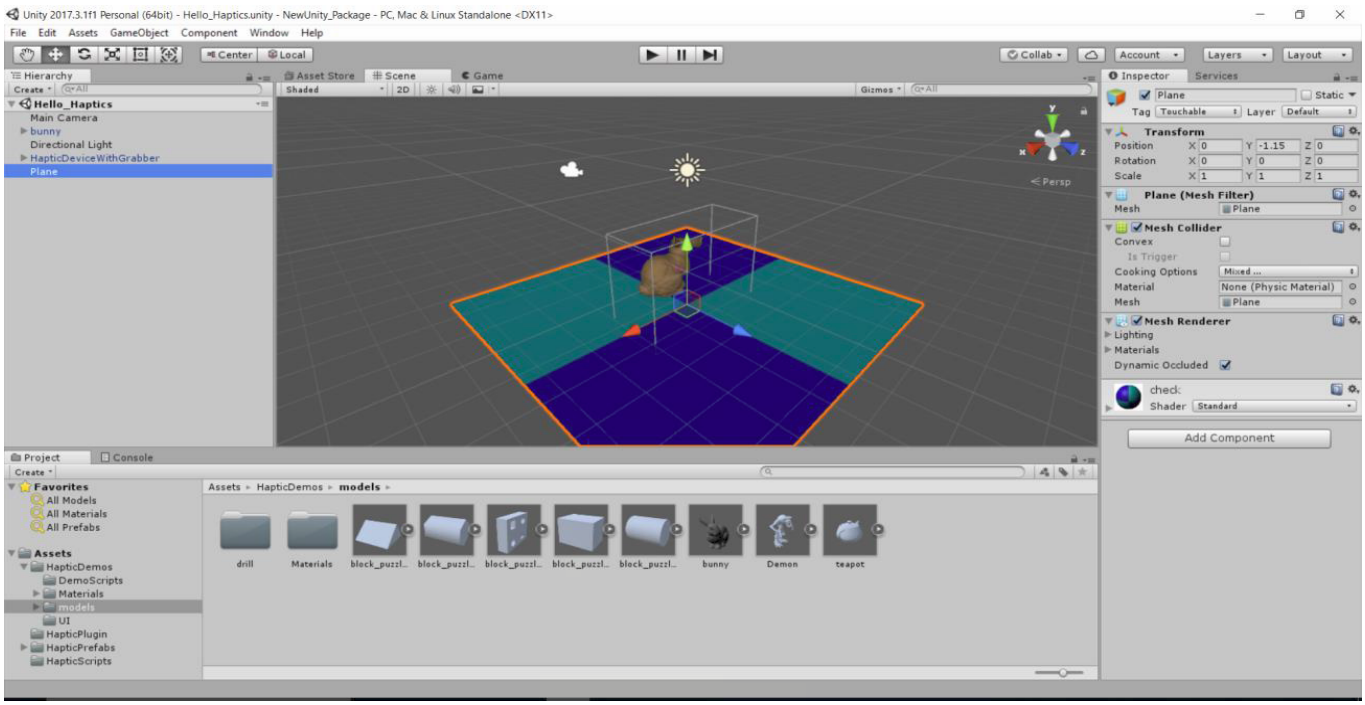
Note: Unity on its own adds mesh colliders to the available 3D objects.

The **Inspector** windows for the Bunny object and Plane object are shown below for reference and comparison.

Inspector Window: Bunny 3D Object



Inspector Window: Plane 3D Object (Unity Provided)



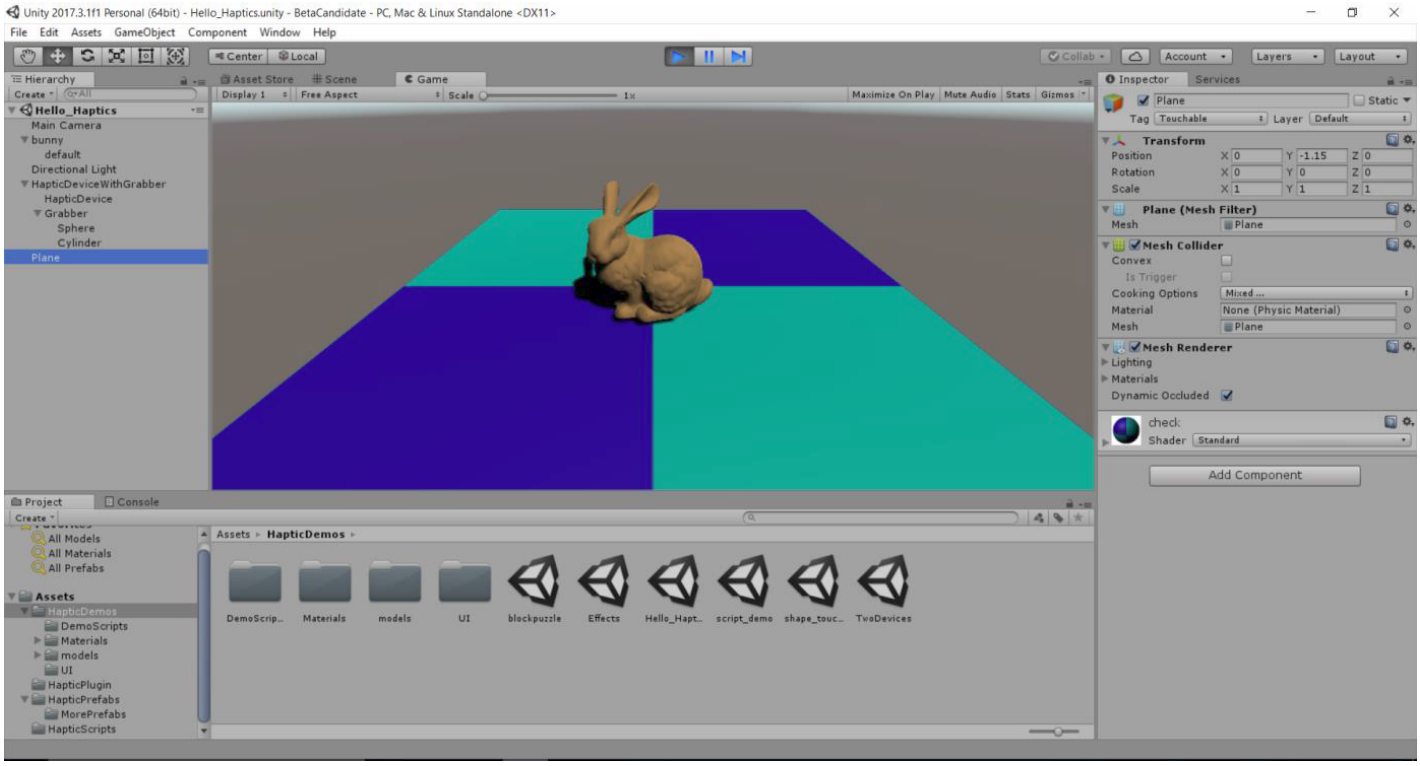
Play the Demo:

Click on the **Play** button and user must be able to touch and feel the custom 3D object (Bunny) and also the Plane 3D Object. In addition, user can see that the **Haptic Cursor**, as in this demo, the **Grabber** moves in line with the device movement.

Tip: It is best practice not to add objects or link the cursor to the haptic device when the demo is in the play mode. Changes made during the play mode do not reflect after stopping the demo. Hence users are advised to make all their changes before playing the demo.

Tip: Users can also scale the 3D objects including the cursor as per their requirement by changing values in the **Inspector** window.

Runtime View of the Demo:



In the following chapters we will be discussing how to use various haptic properties, effects, physics and other advanced techniques.

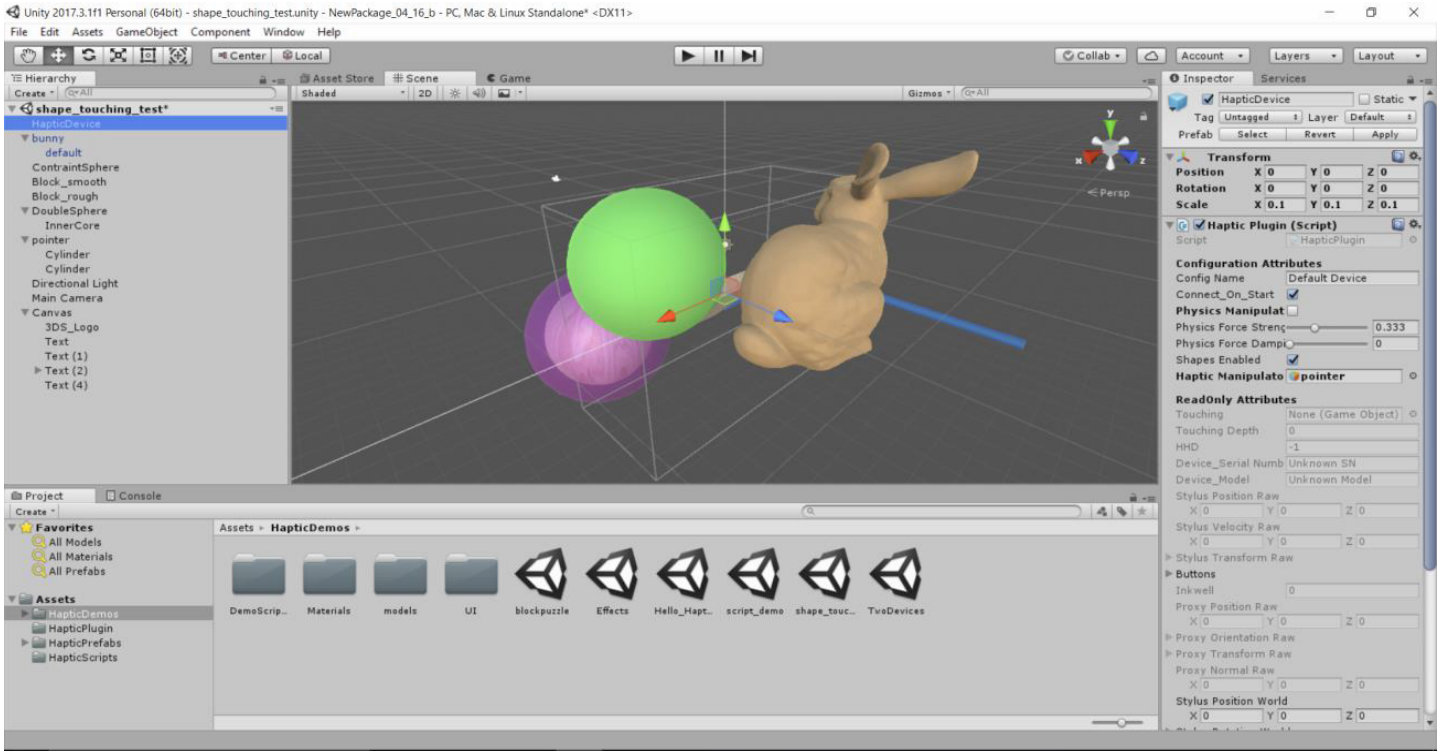
4 CHAPTER TWO - SHAPE TOUCHING TEST DEMO

This demo elaborates on the use of different modes of haptic rendering and adding Haptic Materials to a scene in Unity.

Note that this demo only uses the **HapticDevice** prefab found in the **Haptic Prefabs** folder and creates and links a custom cursor.

This demo can be found under **Assets -> Haptic Demos Folder -> Choose "Shape_Touching_Test".**

Once the demo is opened, users can see that there are various shapes like bunny and spheres displayed. The left pane (the **Hierarchy** pane) will clearly show all the objects added to the demo. The **Hierarchy** window looks like the screenshot below.



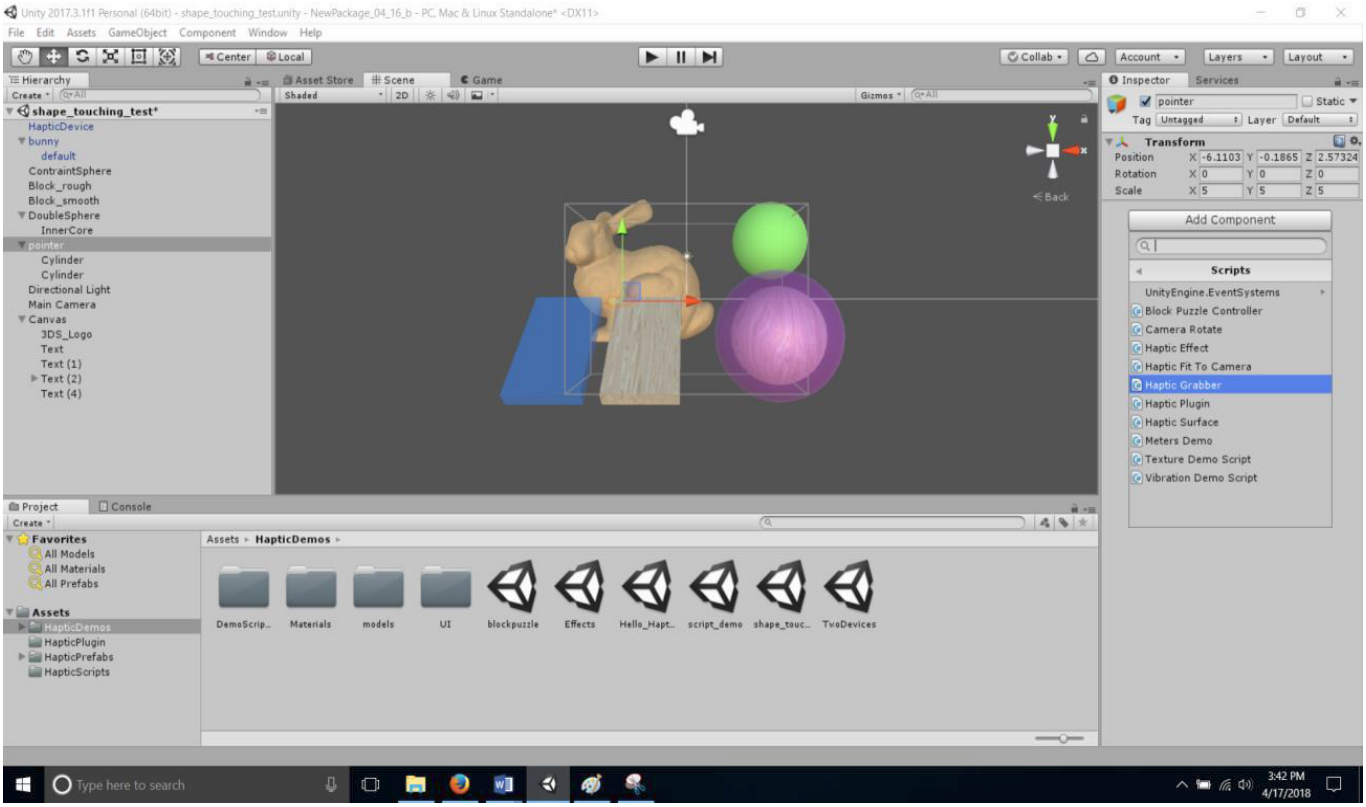
In this demo, only the **HapticDevice** prefab (can be found under **Haptic Prefabs->More Prefabs**) is added and not the **HapticDevicewithgrabberprefab**. This specific prefab (**HapticDevice** only) lets user to customize their haptic cursor. In this case, **Pointer** is a custom haptic cursor that plays the same role as the **Grabber**. Please note that the custom cursor must be a 3D object.

Creating a Custom 3D Cursor:

<<Need clarification on if grabber script needs to be added or not>>

It is extremely easy to create custom cursor in unity.

1. Add any 3D object to the scene. You can rename the 3d object to reflect your custom cursor name.
2. Link the cursor to the haptic device: Click on the **Haptic Device** from the hierarchy window. In the **Inspector** pane, link the custom cursor by drag dropping into the **Haptic Manipulator** field. Refer to the previous screenshot.
3. Add the **Grabber Script**: Add the grabber script to the custom cursor. Click on **Add component -> Scripts -> Select Haptic Grabber**. Select **Physical toggle: On Grab or On touch**.



Tips:

- Scale the haptic workspace appropriately to see the cursor move.
- Do not make custom cursor touchable. It will cause more friction while interacting with objects.

Important Properties:

Haptic Manipulator: The **Haptic Manipulator** can be compared to a mouse cursor. It helps the user to navigate and touch 3D objects. Users will only be able to touch 3D objects if the appropriate manipulator is specified.

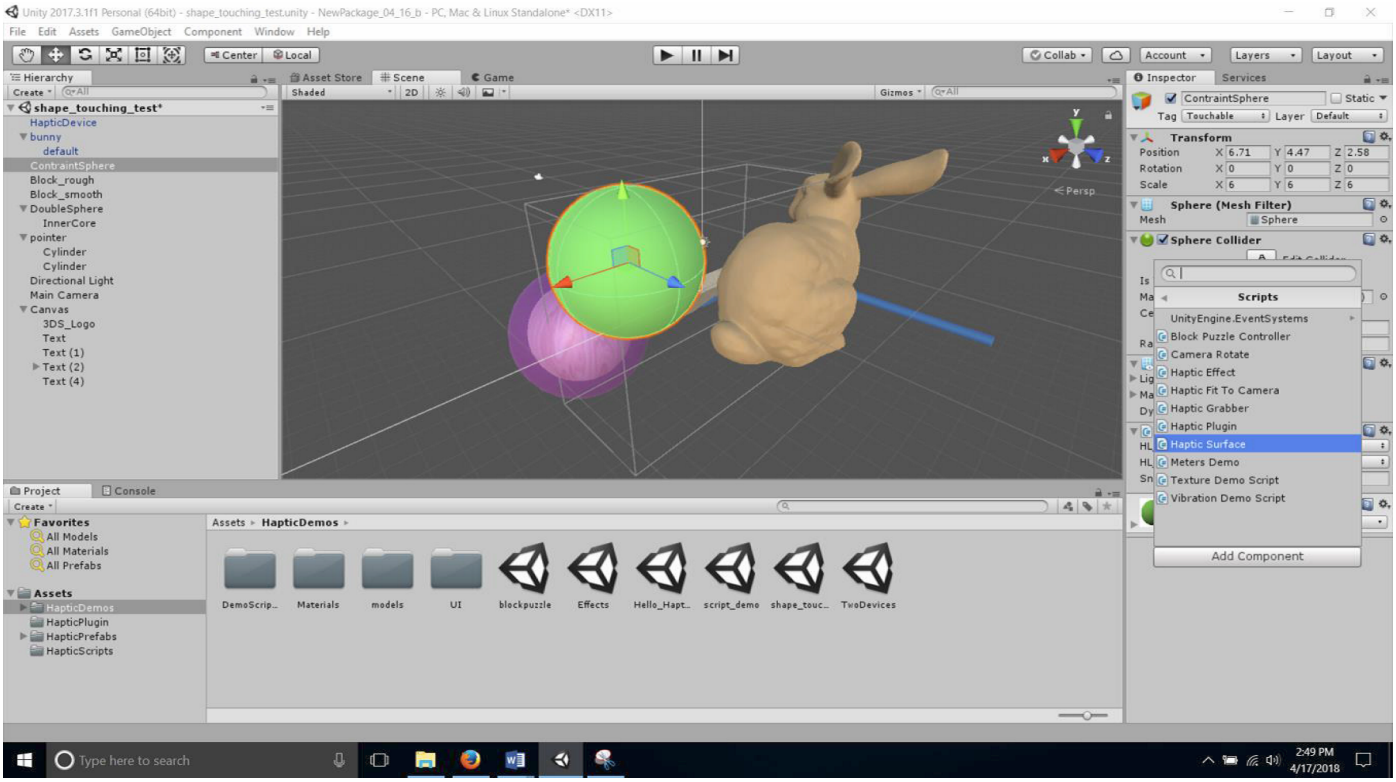
Whenever a custom cursor is created, it is imperative to link the cursor to the haptic device. This must be done through the **Haptic Manipulator** field. Once the cursor is created, drag and drop the cursor from the **Hierarchy Window** on the left side to the **Haptic Manipulator** field under **Inspector** for the **Haptic Device**. Refer to the above screenshot.

Adding Haptic Materials to 3D Objects:

Haptic materials can be added to any 3D object by incorporating the **Haptic Surface Script**.

To add the script, simply select the object and in the Inspector window, click on **Add Component > Scripts > Haptic Surface**.

See the following image for reference.



The available haptic rendering modes can be broadly broken down into **HL_Contact** and **HL_Constraint**.

There are 5 inherent properties under **HL_Contact:**

- **Stiffness** - The stiffness Haptic material surface directly translates to how hard a surface feels and haptic feedback can be expressed using the formula, F=Kx where F is the force applied is directly proportional to the distance (x) and K is the spring constant. Note: If stiffness is set at “Zero” then no other haptic material effect can be felt.
- **Damping** - Damping reduces the spring effect of the surface. Param must be between 0 and 1 where 0 represents no damping, i.e. a highly springy surface and 1 represents the maximum level of damping possible. Damping is added to reduce rebound effect.
- **Static friction** - The surface of the object (eg: plane) how hard it is to slide along the surface starting from a complete stop. A param value of 0 is a completely frictionless surface and a value of 1 is the maximum amount of static friction the haptic device is capable of rendering.
- **Dynamic Friction** – This property defines how hard it is to slide along the surface once already moving. A param value of 0 is a completely frictionless surface and a value of 1 is the maximum amount of dynamic friction the haptic device is capable of rendering.
- **Pop through** - Popthrough controls the amount of force the user must apply to a shape before the device pops through the shape to the other side. The larger the param value, the higher the force required. A param value of o disables pop through.

The only property under **haptic_Constriant** is **Snap Distance** which is the distance between the proxy position and the surface that must be exceeded to pull off a constraint. Parameter should be a floating point value representing the distance in millimeters in workspace coordinates. The default value is **FLT_MAX** to always be active.

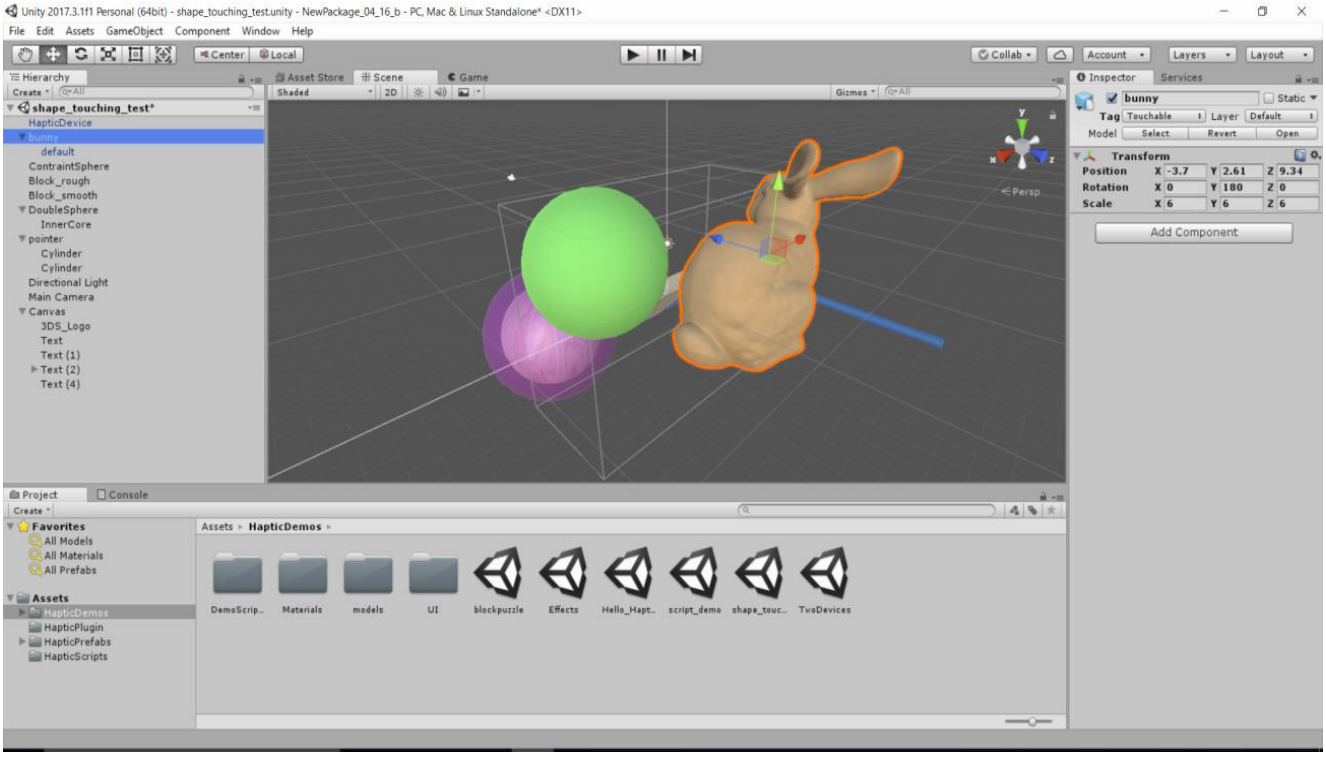
Tip: Users are recommended to make themselves aware of these properties by reading the **Openhaptics Programming Guide** for more elaborate and detailed understanding of this material.

We will now look at how these parameters have been implemented in the demo.

In the demo, users can see that there are six components or 3D objects. Each 3D object in the scene demonstrates the haptic rendering modes and material properties.

1. Bunny 3D object

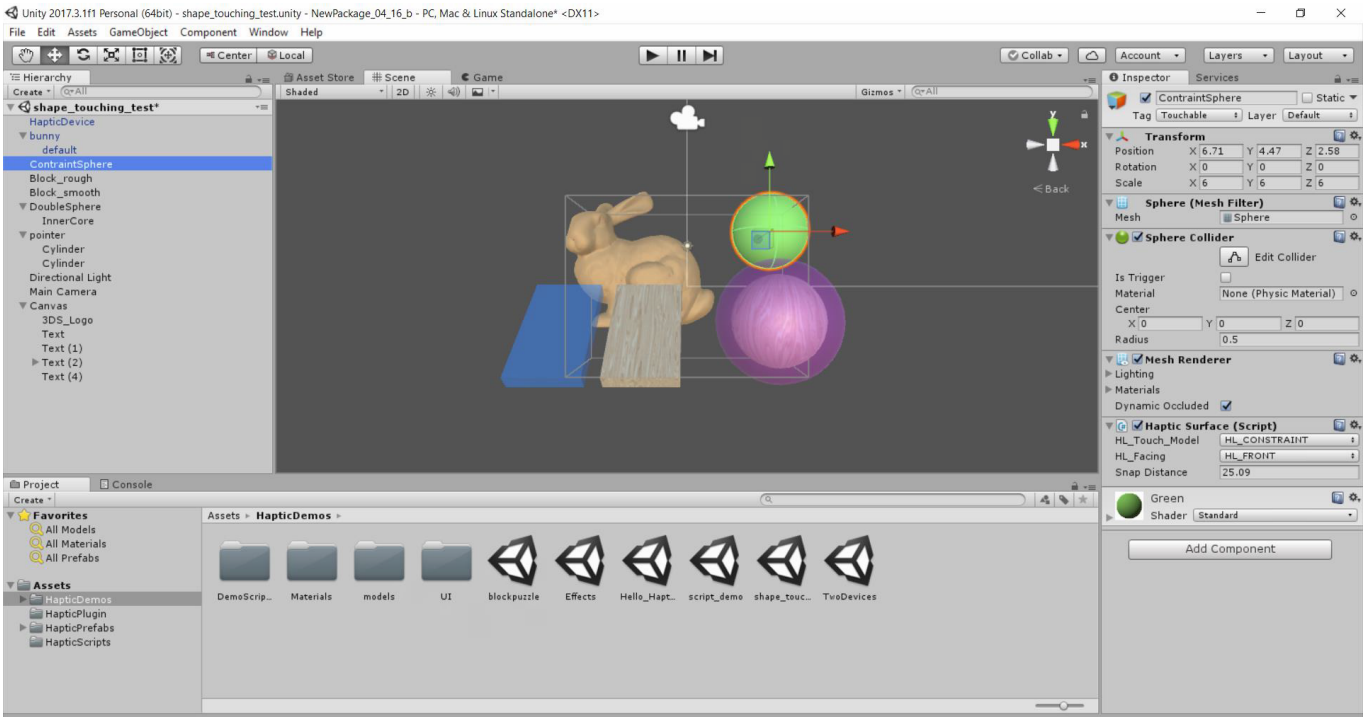
This object does not have any haptic materials added.



2. ContraintSphere Object

This object has a haptic material property of "**HL_Constraint**" with a **Snap Distance** of 25.09 set with **HL_Facing: HL_Front.**

These parameters let the user feel the snap distance at work so that if user moves the haptic cursor closer to the green sphere object they can feel force similar to that of a magnet as if the cursor is pulled in or attracted to the Surface of the sphere.



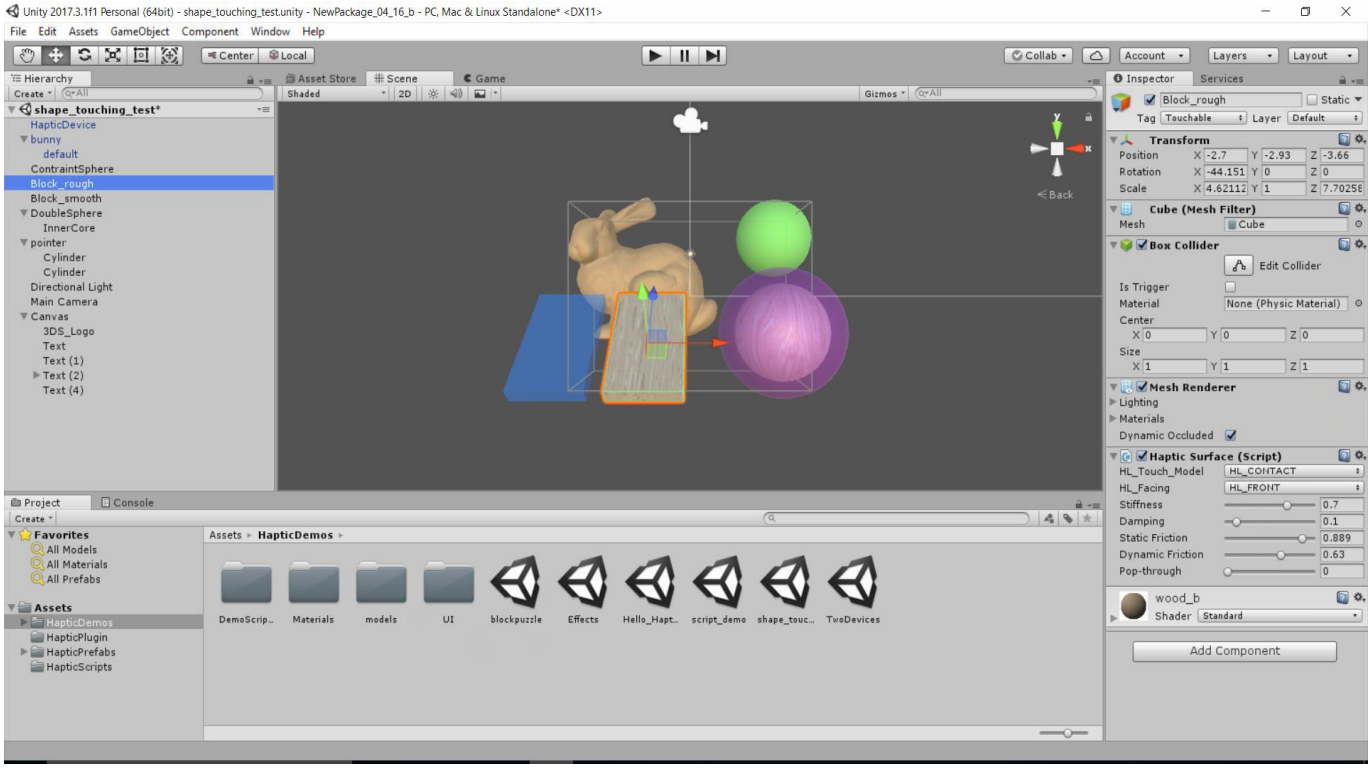
Note: The HL_Touch_Model parameter is where the users can select between "HL_Contact" and "HL_Constraint"

There is another parameter called **HL_Facing**. This parameter determines the "Side" of the surface to which the haptic surface property is to be applied. In this case, the parameter points to **HL_Front**, which is the front surface of the sphere.

HL_Facing has three options namely: **HL_Front**, **HL_Back**, **HL_Front_And_Back**. These parameters can be adjusted as desired by user to obtain desired surface properties.

3. Block_Rough

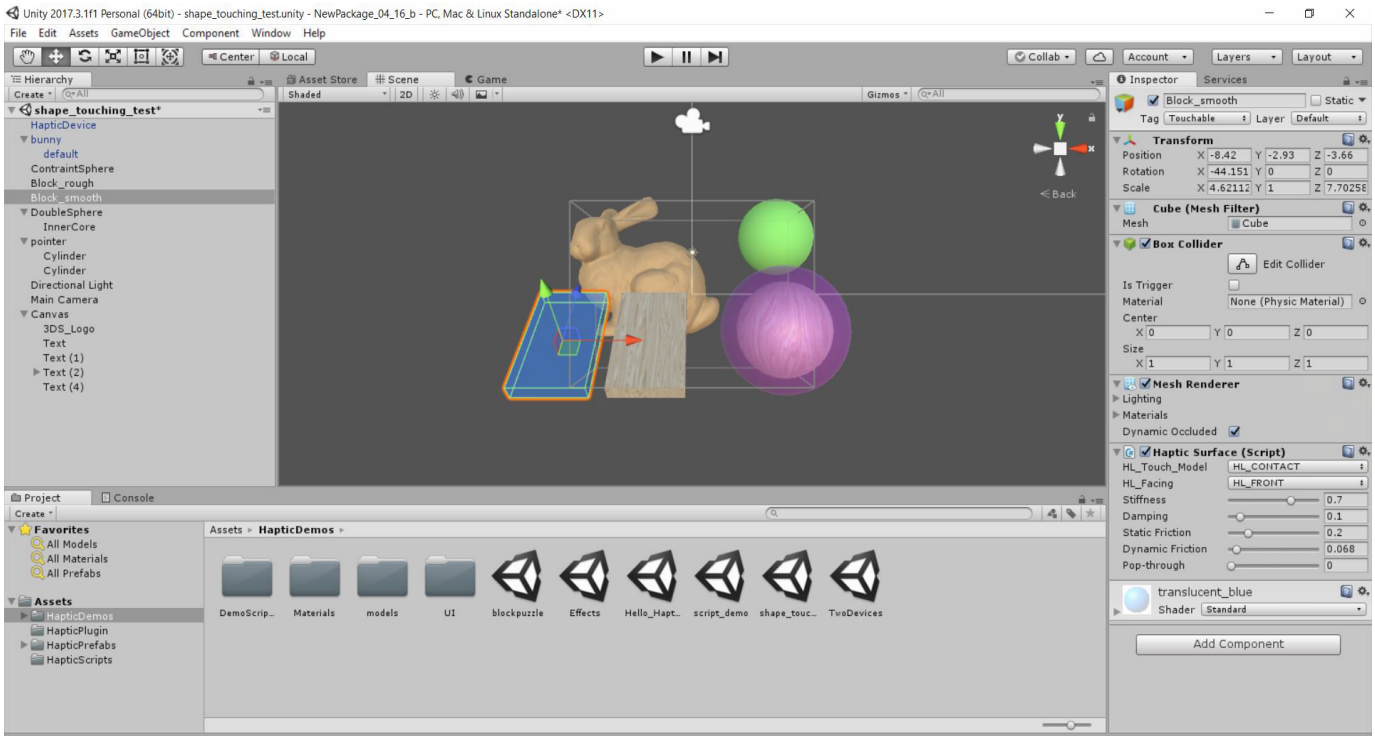
This 3D Object is one of the example of **HL_Contact** with **HL_Facing: HL_Front** and various values assigned to sub properties.



These values apply high friction to the wooden surface. These values can be altered as desired.

4. Block_Smooth

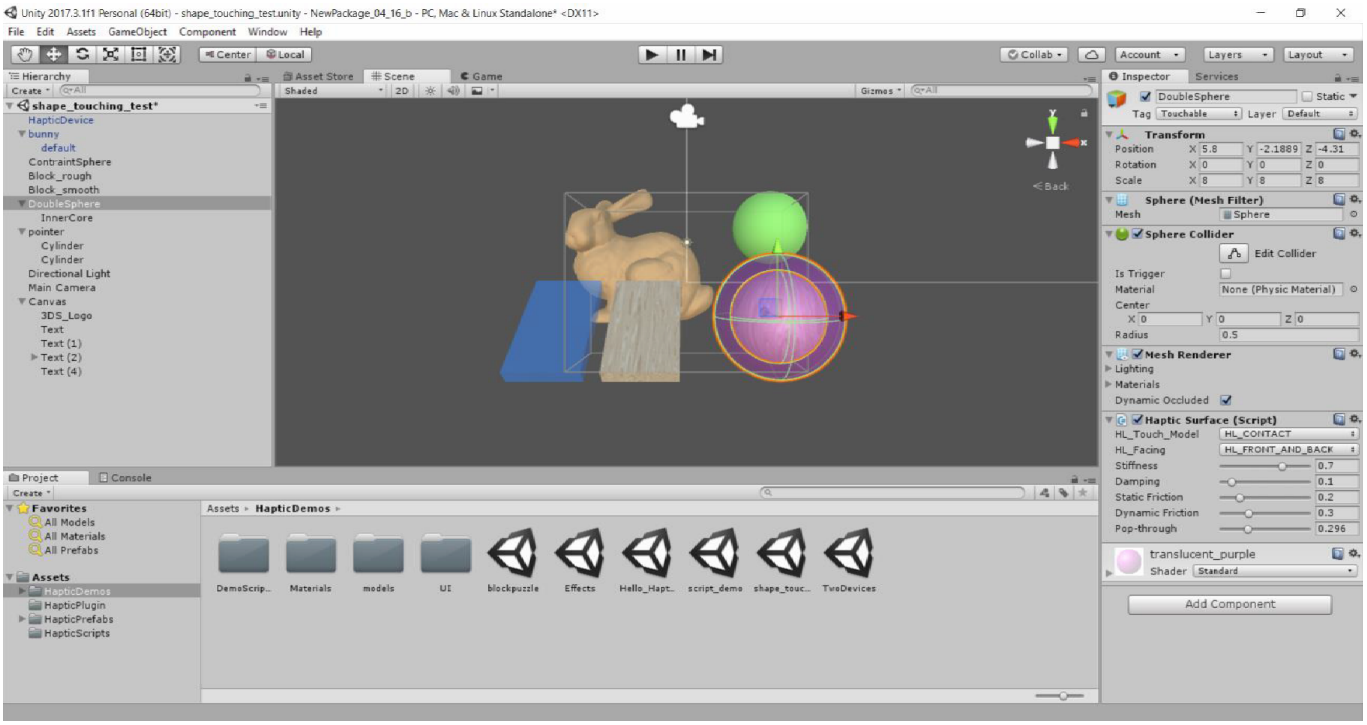
This 3D Object is one of the example of **HL_Contact** with **HL_Facing: HL_Front** and various values assigned to sub-properties.



These values pply low-friction to the translucent blue surface. These values can be altered as desired.

5. Double Sphere

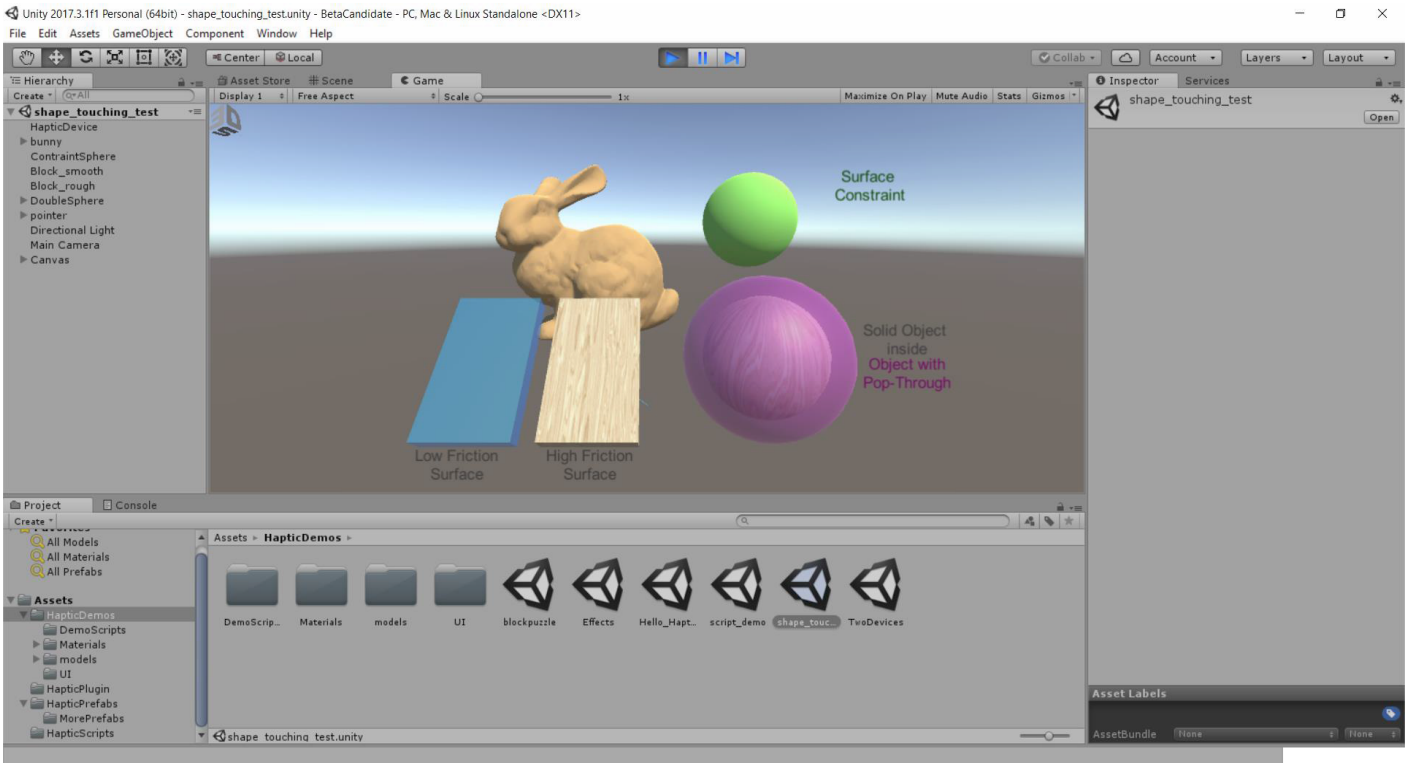
This 3D object is one of the example of **HL_Contact** with **HL_Facing: HL_Front _and_Back** and various values assigned to sub properties. This 3D object also has another sphere within it to demonstrate the pop through surface property and the **HL_Front_And_Back** property.



These properties let the user understand the pop through effect with which the users can pop through the purple sphere and feel the inner wooden sphere. The **HL_Front_And_Back** property is also demonstrated wherein users could feel the inner side of the purple sphere and the front side or outside of the purple sphere.

Runtime View of the Demo

Click the **Play** button to activate a runtime view of the the demo.



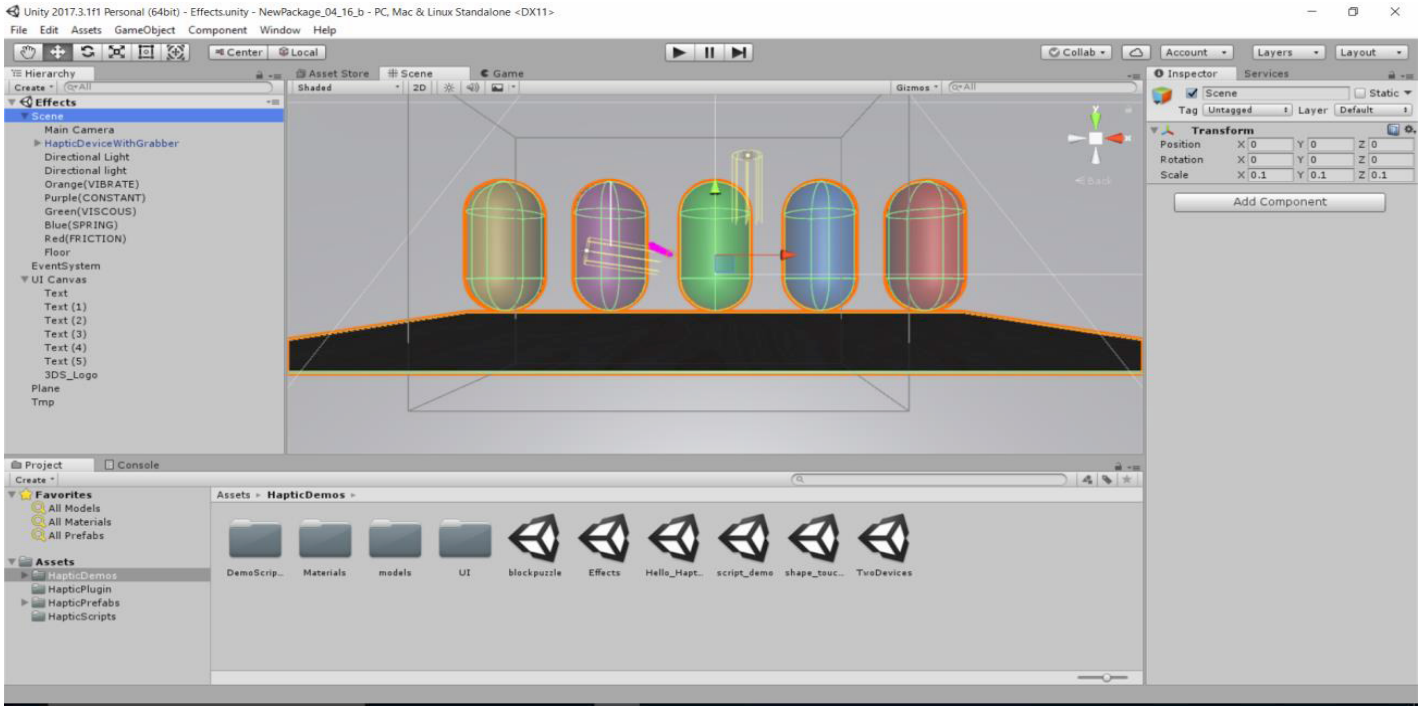
5 CHAPTER THREE - EFFECTS DEMO

This demo elaborates on how to add **Haptic Effects** to a scene in Unity.

It is to be noted that this demo uses the **Haptic Device with Grabber** prefab found in the **Haptic Prefabs** folder.

This demo may be found under **Assets ->Haptic Demos Folder -> Choose "Effects"**

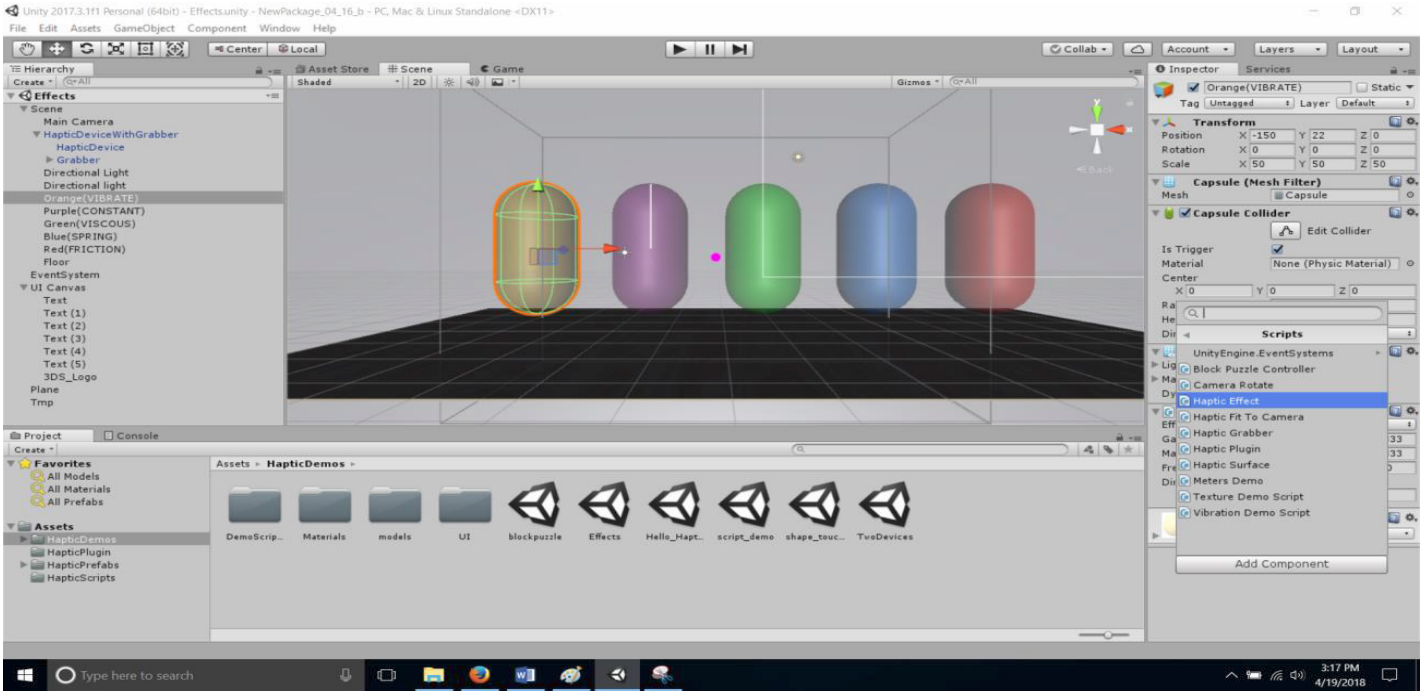
Once the demo is opened, he user is presented with six cylinders or capsules that demonstrate each haptic effect with appropriate parameters. See the following image of the demo.



Adding Haptic Effects to 3D Objects:

Haptic effects can be added to any 3D object by incorporating the **Haptic Effect** script. To add the script, simply select the object and in the Inspector window, then select **Add component -> Scripts -> Haptic Effect.**

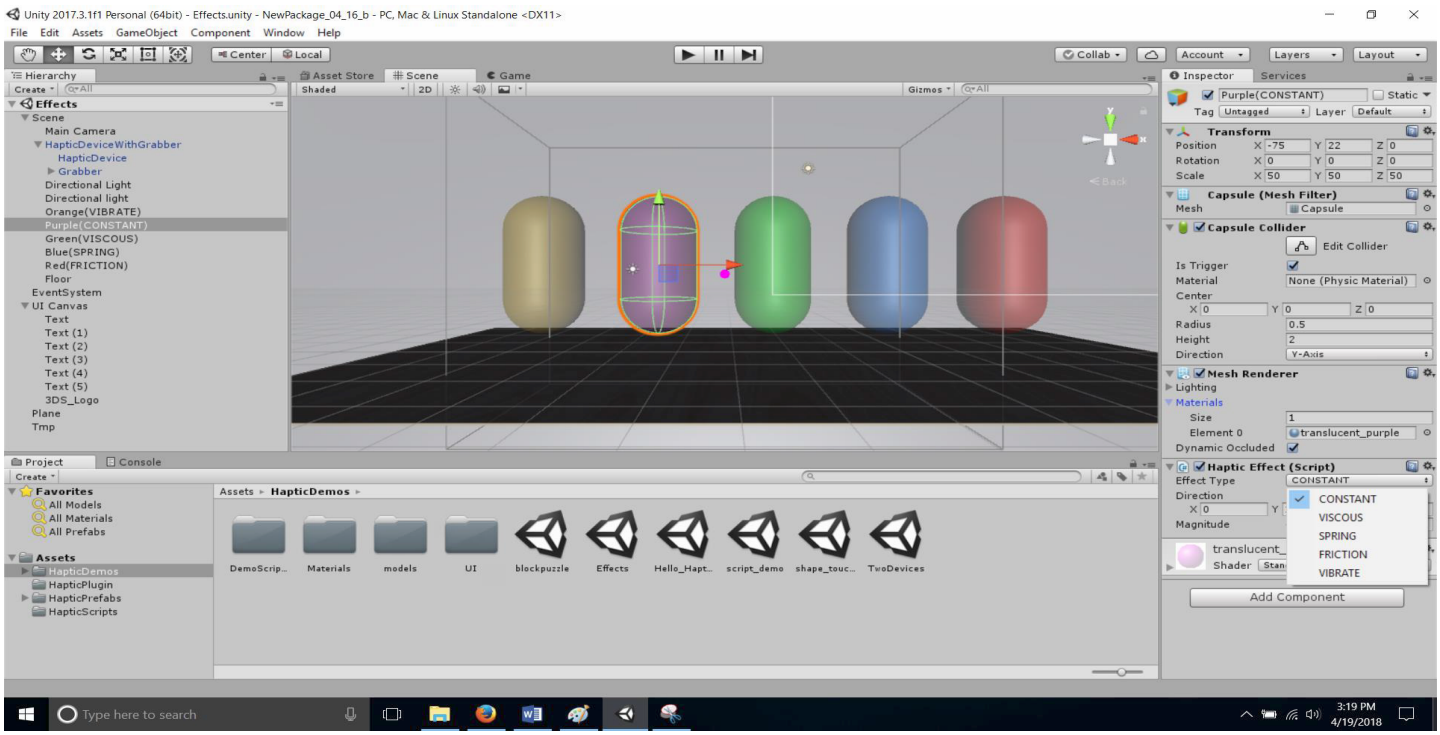
See the following image for reference.



Haptic effects can be added to any 3D object by adding the **Haptic Effect** script. There are five primary effects that the user ay experiment with:

- **Constant Force** – It adds a constant force vector to the total force sent to the haptic device. User can specify the direction and magnitude of the force vector.
- **Viscous** – It adds a viscous force to the total force sent to the haptic device. The viscous force is based on the current velocity of the haptic device and is calculated to resist the motion of the haptic device. Specifically the force is calculated using the expression F = -kV where f is the spring force, V is the velocity and k is the gain. User must specify : Magnitude and Gain
- **Spring** – It adds a spring force to the total force sent to the haptic device. The spring force pulls the haptic device towards the effect position and is proportional to the product of the gain and the distance between the effect position and the device position. Specifically, the spring force is calculated using the expression F = k(P-X) where F is the spring force, P is the effect position, X is the current haptic device position and k is the gain. User must specify : Gain, Magnitude and Position
- **Friction** – It adds a friction force to the total force sent to the haptic device. User must specify : Magnitude and Gain
- **Vibrate** – It adds a vibration effect to the total force sent to the haptic device. User must specify: Frequency, Direction, Magnitude and Gain.

These forces are shown in the following image.



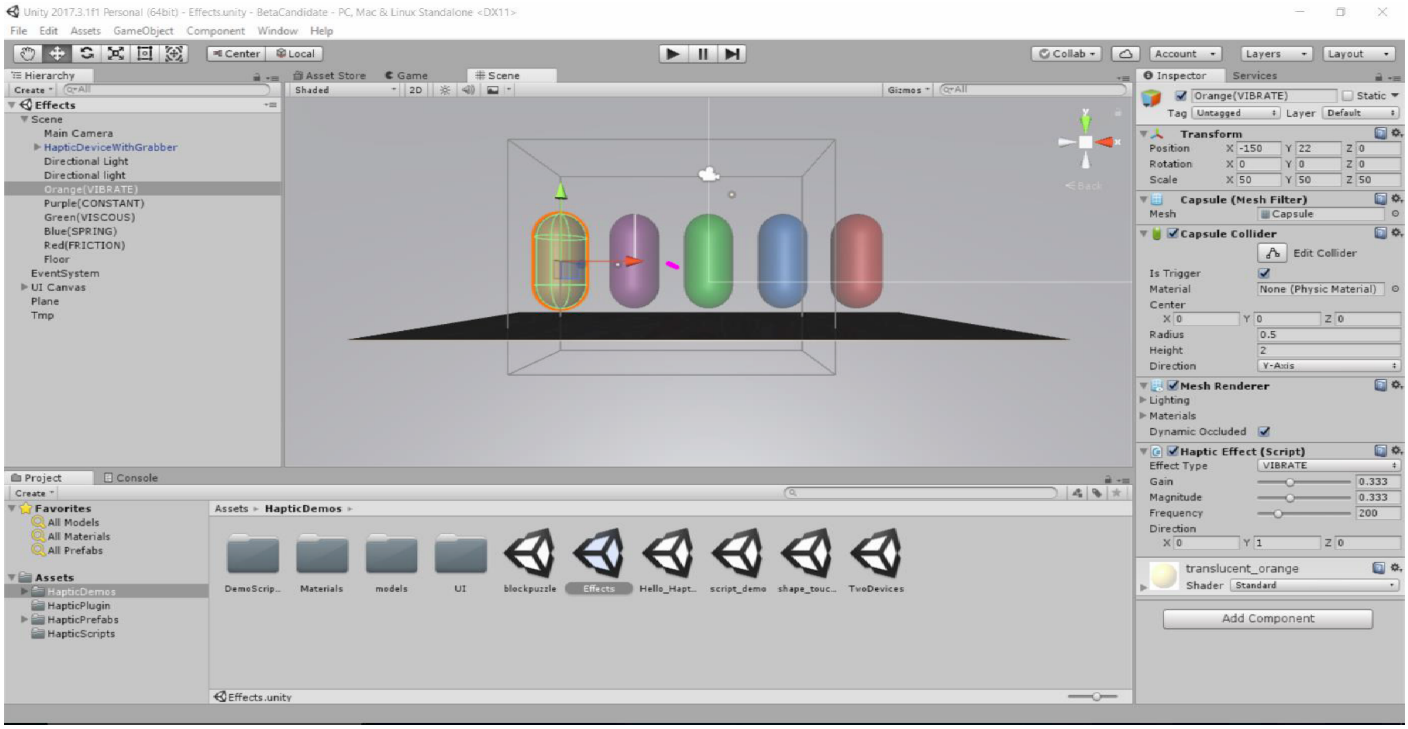
Tip: It is recommended that users learn more about these properties by reading the **Openhaptics Programming Guide**.

We will now disucss the implementation of these parameters in the demo.

Users can see that there are six components or 3D objects in the demo. Each of these objects explains how to add different haptic materials to 3D objects.

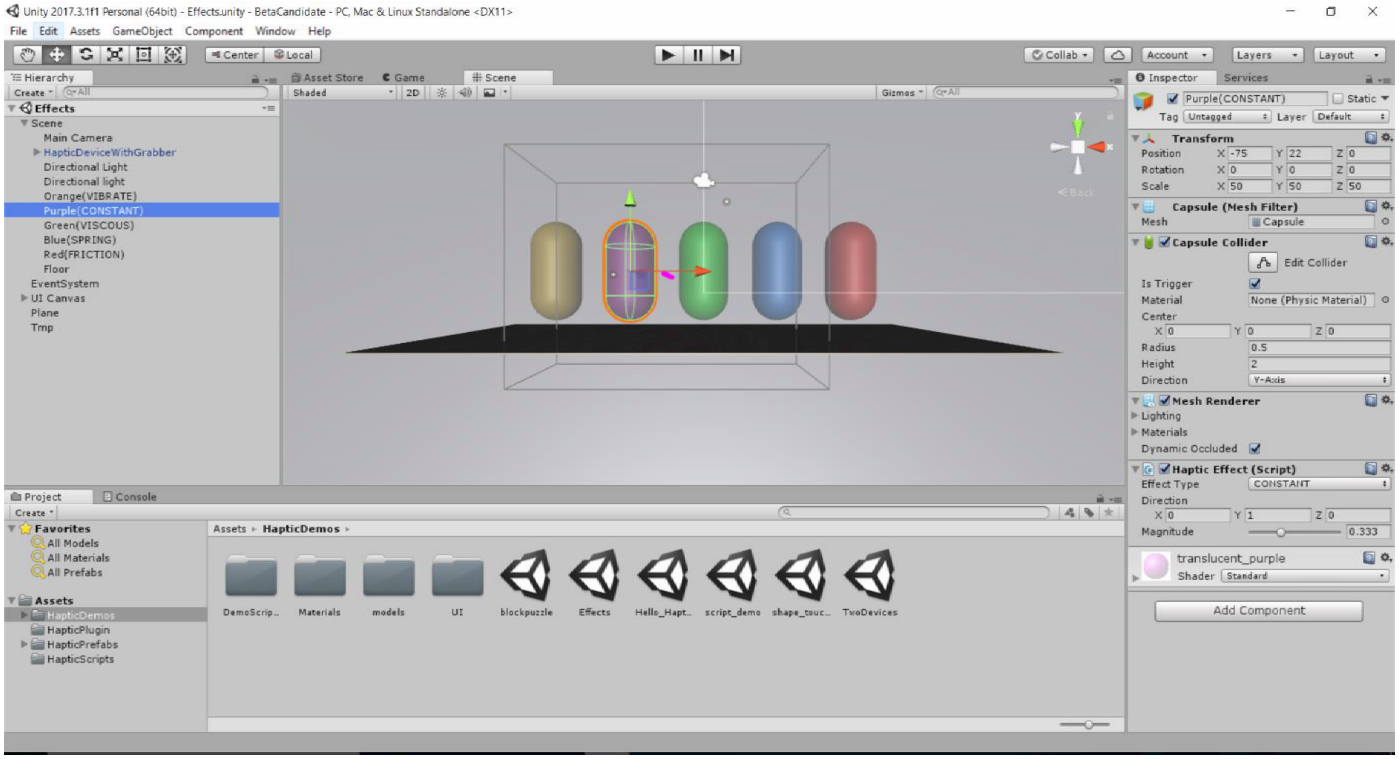
1. Orange (VIBRATE):

The parameters **Frequency**, **Direction**, **Magnitude** and **Gain** are available for modification.



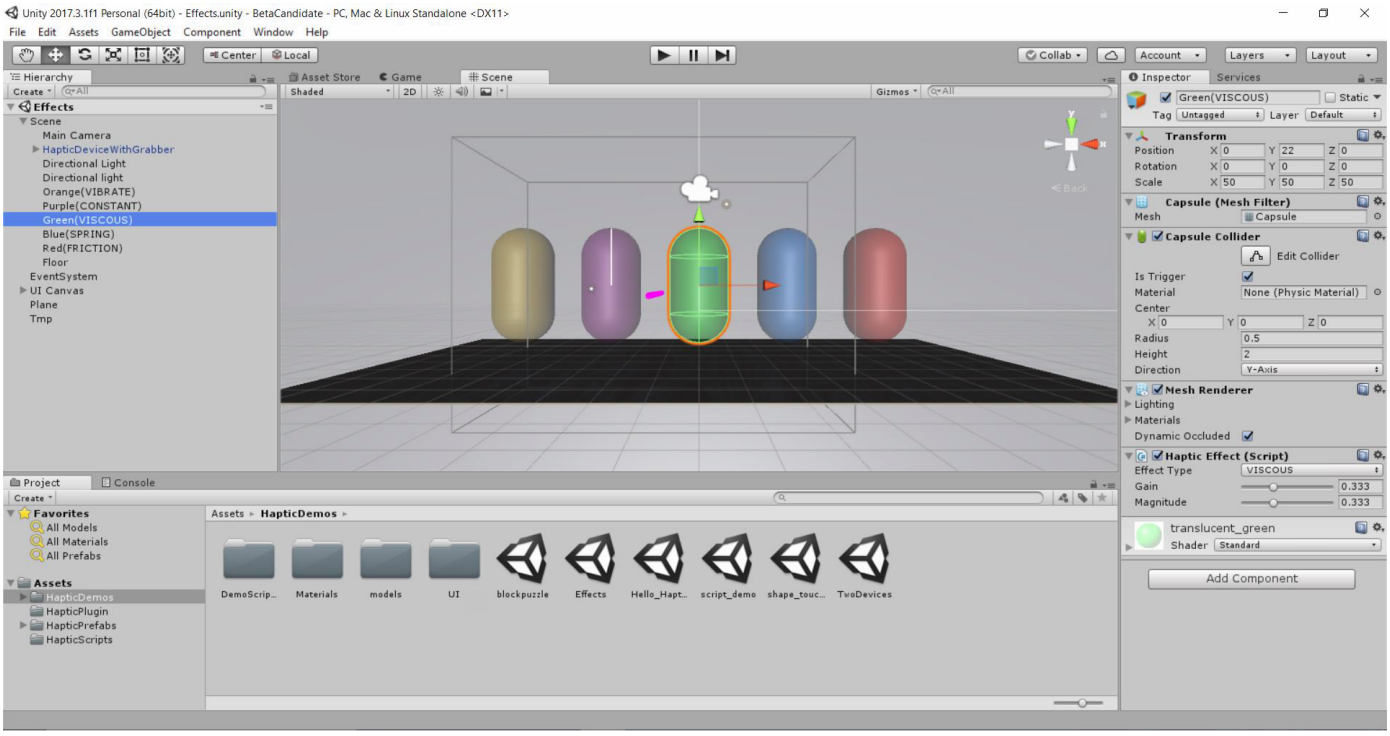
2. Purple (CONSTANT):

The parameters **Direction** and **Magnitude** are available for modification.



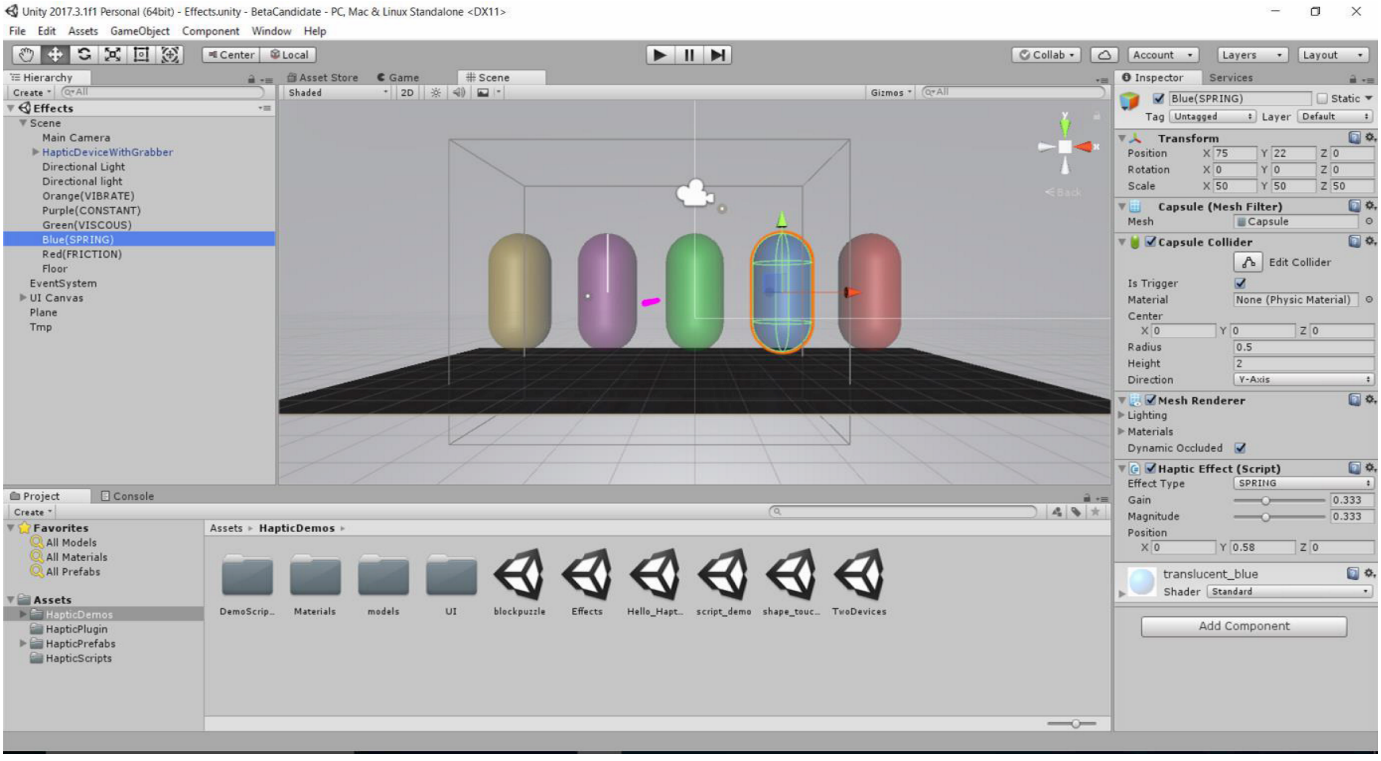
3. Green (VISCOUS):

The parameters **Gain** and **Magnitude** are available for modification.



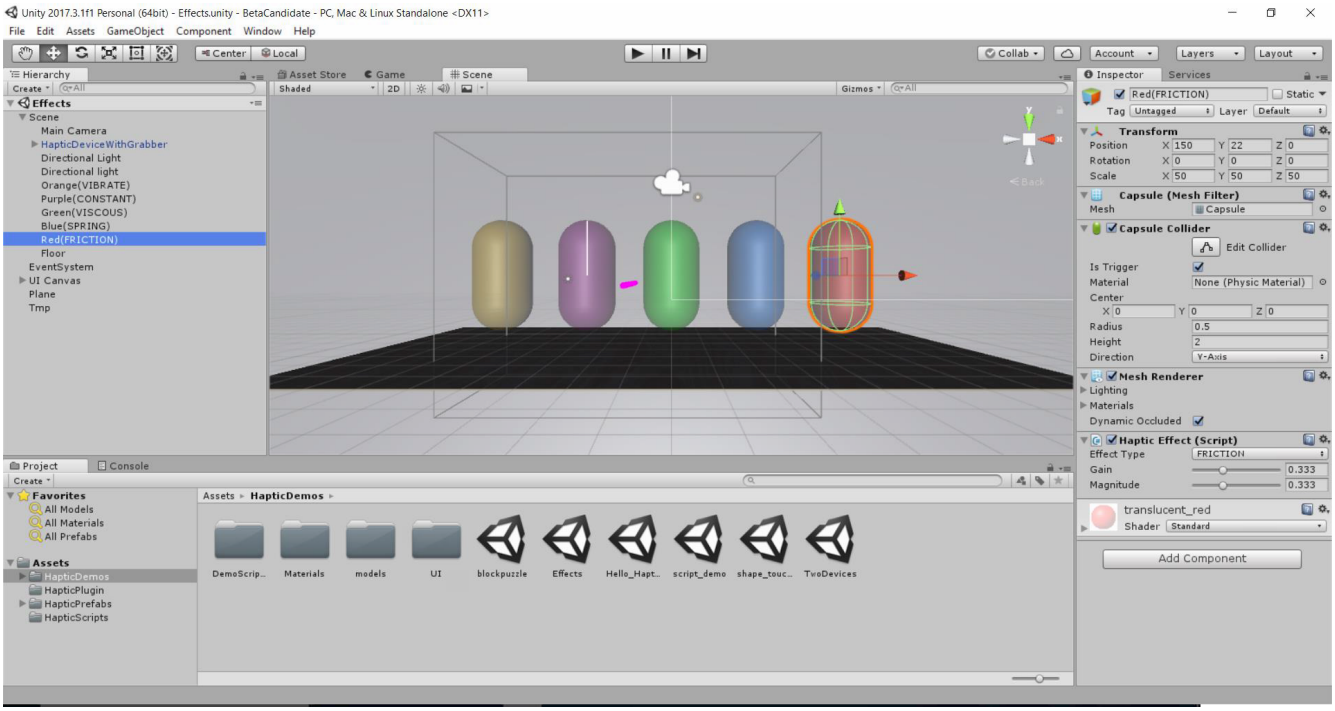
4. Blue (SPRING):

The parameters **Gain**, **Magnitude** and **Position** are available for modification.



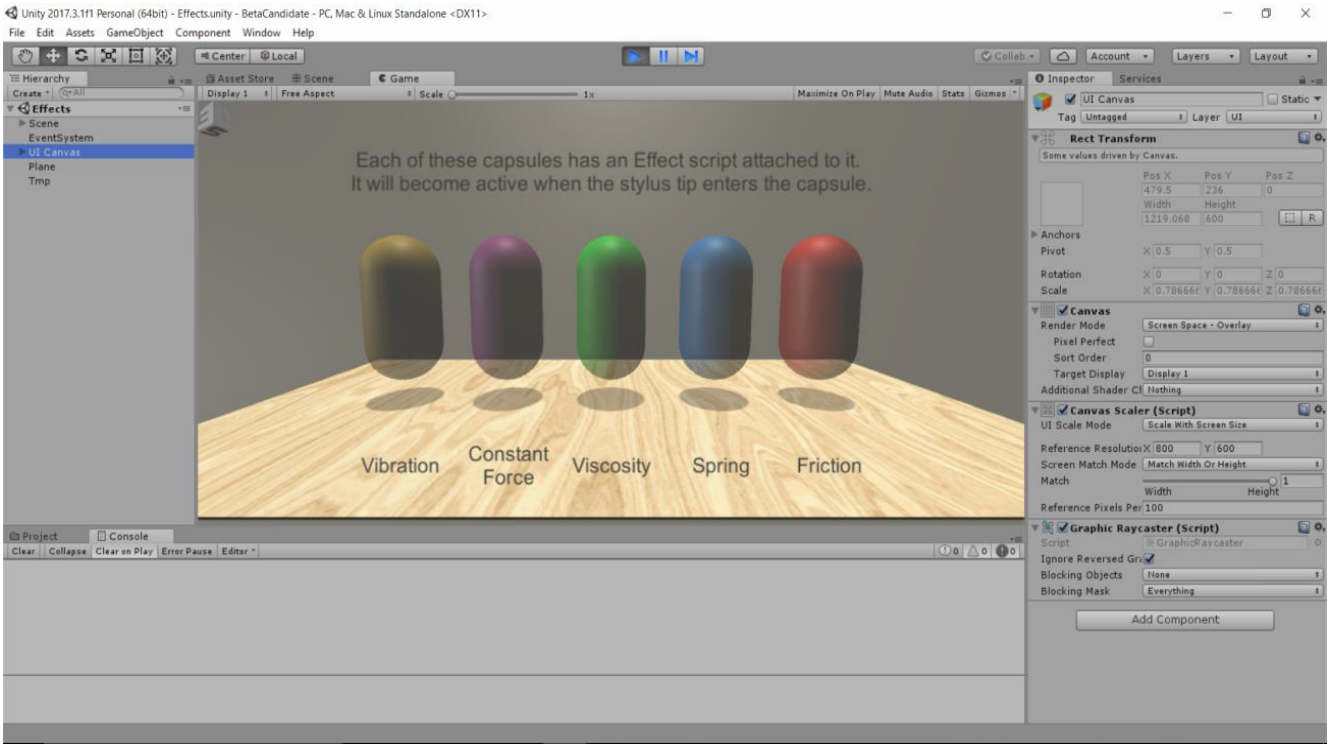
5. Red (FRICTION):

The parameters **Gain** and **Magnitude** are available for modification.



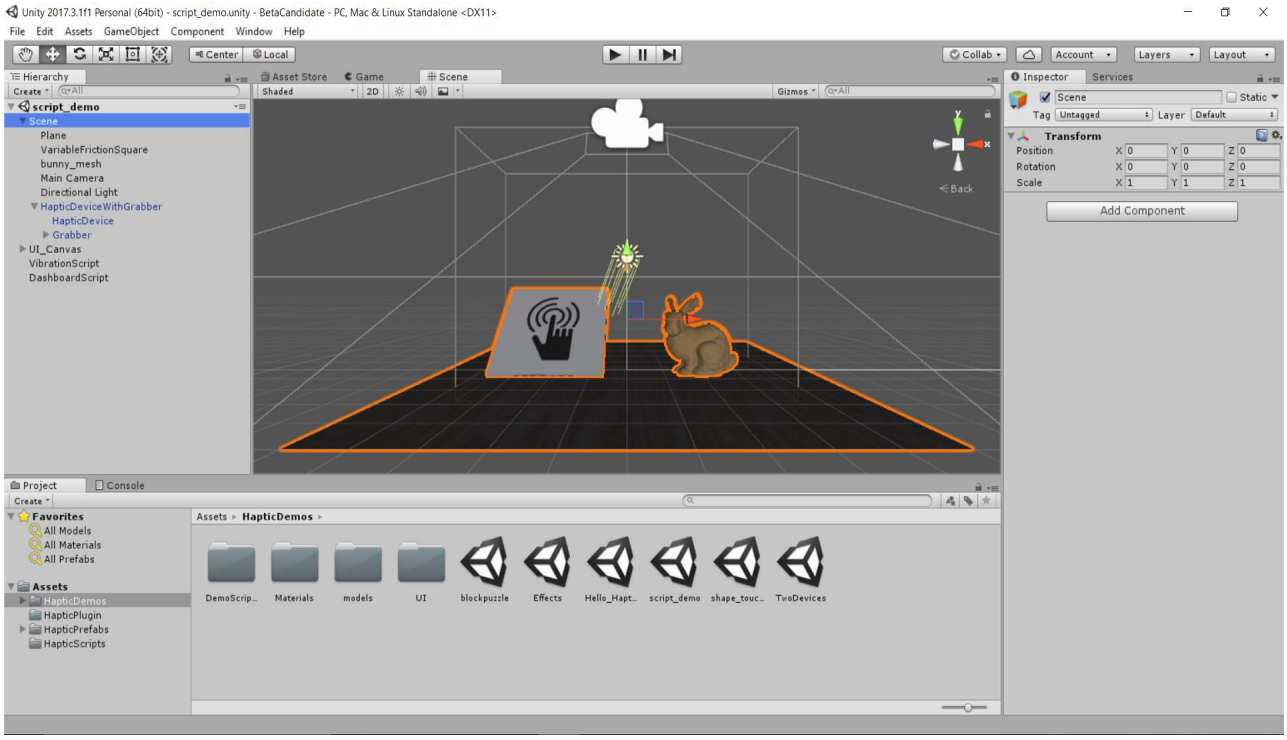
Run time View Of the Demo:

Click the **Play** button. Refer to the following image of the runtime view of the demo.



6 CHAPTER FOUR - SCRIPTS DEMO

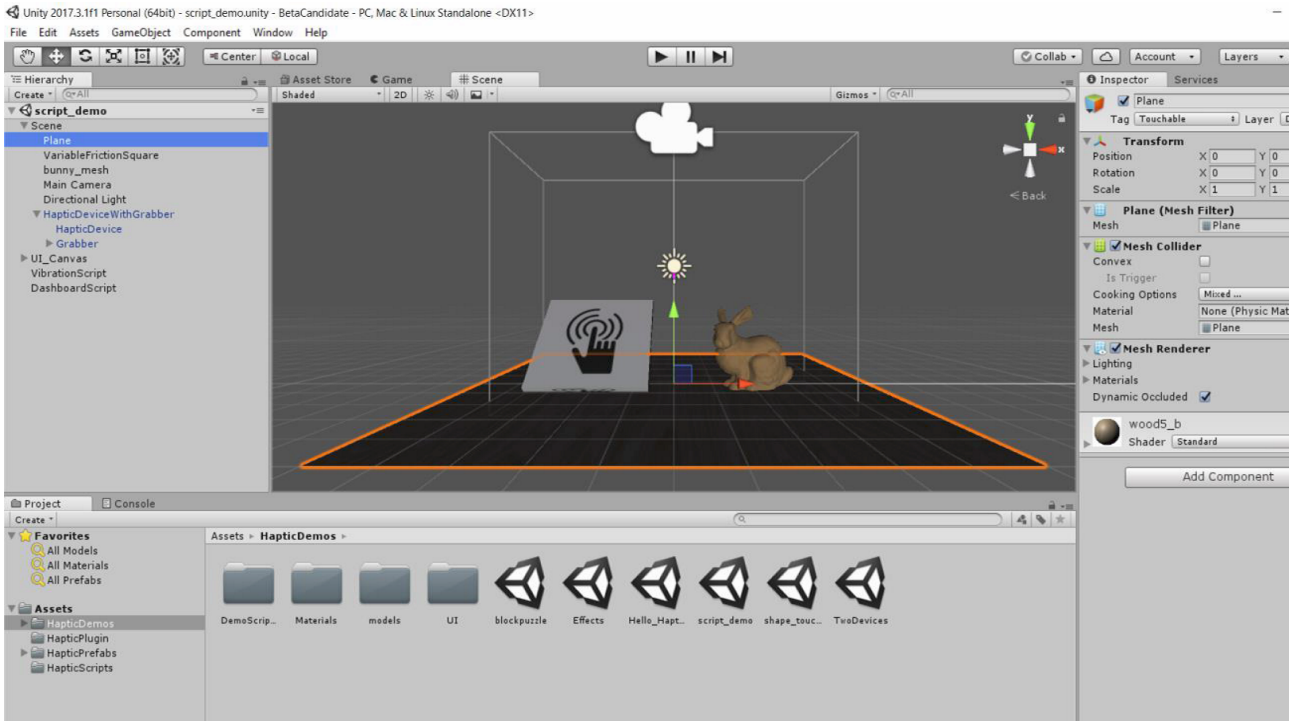
In the above few chapters, we have seen how to add various material properties (Surface Properties) and effects to any 3D objects using the plugin. In this demo, users can understand the actual implementation of effects and surface properties. Refer to the following image which demonstrates the usage of various haptic scripts.



As seen, the demo has three very simple 3D objects namely the Plane, Bunny mesh and Variable friction surface. Let us see below how these simple 3D objects can be manipulated using the haptic scripts to produce various surface properties and effects.

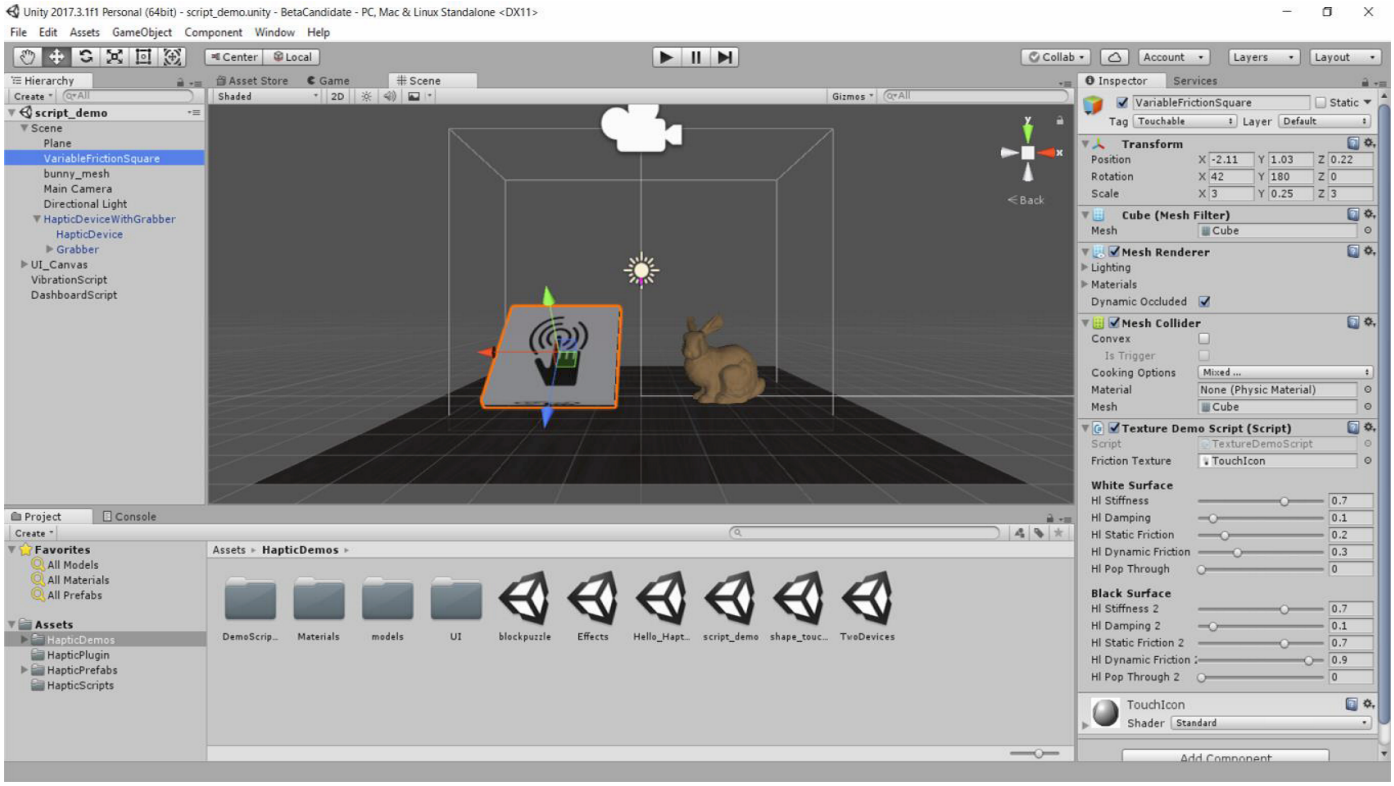
1. Plane:

A simple 3D object that is touchable and has no properties added.

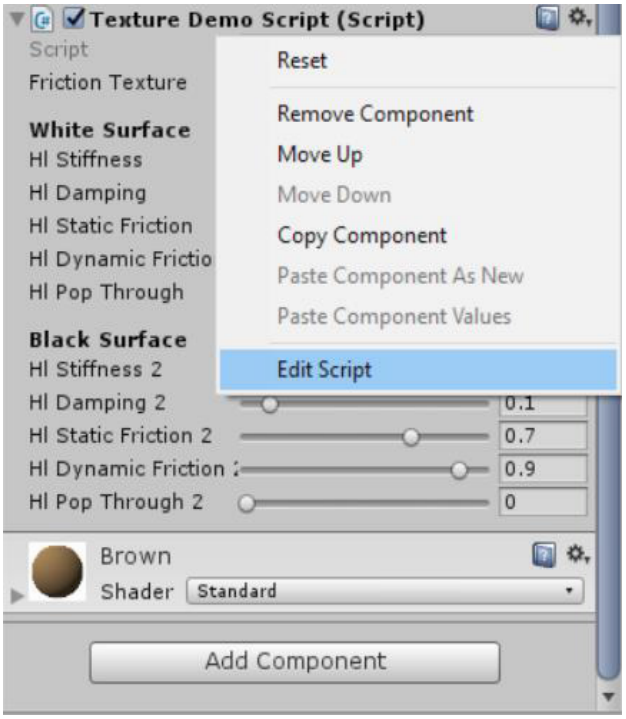


2. Variable Friction Surface:

This 3D object demonstrates the usage of haptic Surface Properties as seen in the screenshot below.

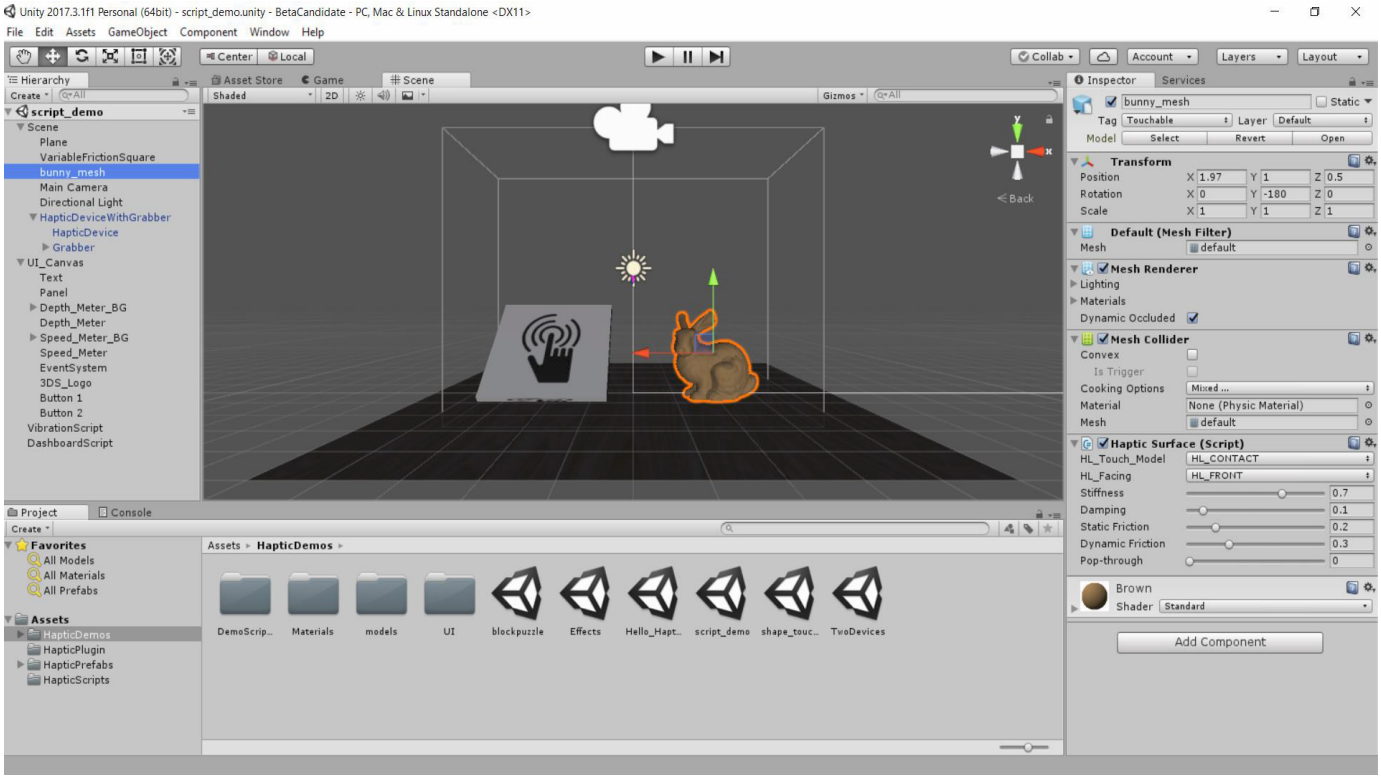


Tip: The script added here is a custom script created using the inherent Haptic Surface properties. Users can write / create their own scripts as seen here. The script used here is the **Texture Demo Script**. The script location: **Add component - > Scripts -> Texture** demo script. Users can edit the script by selecting **Edit Script**.



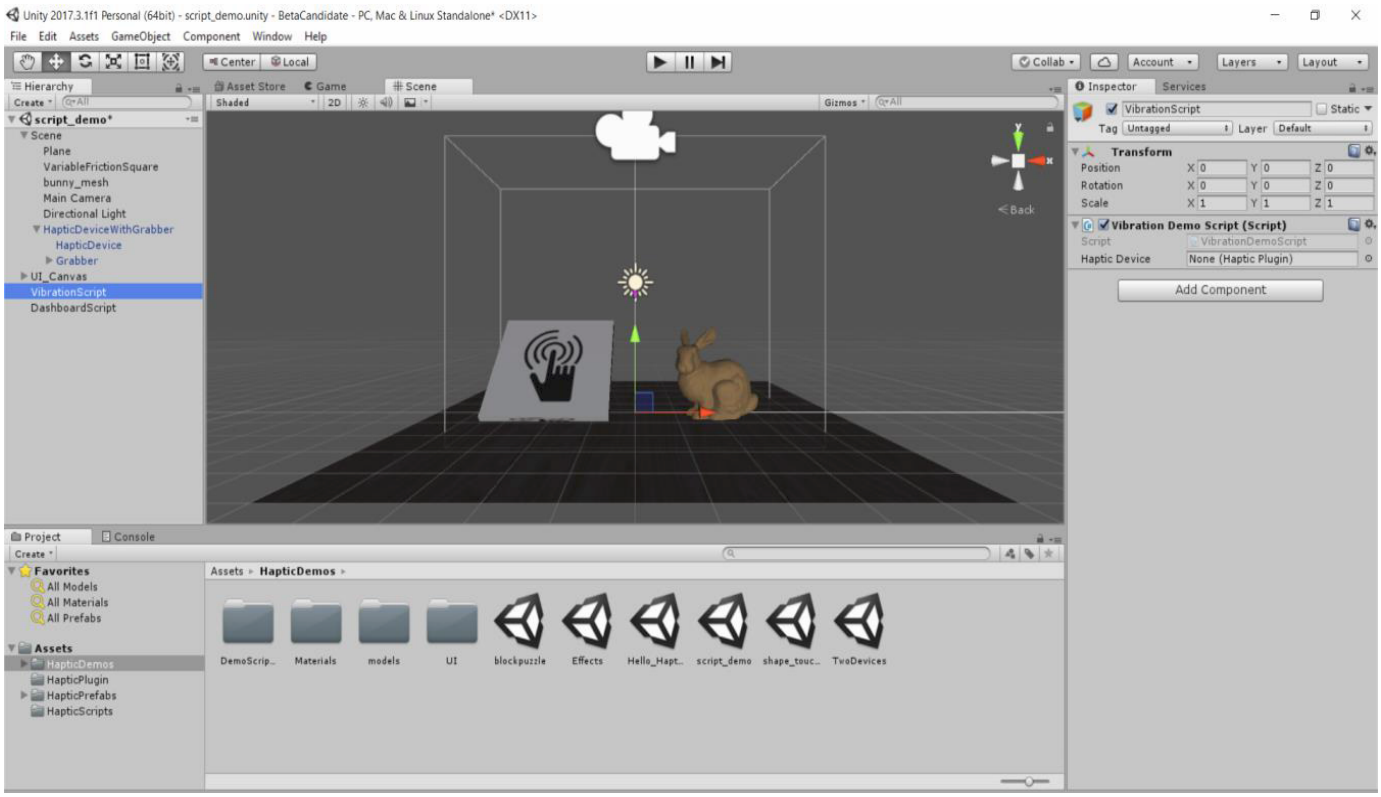
3. Bunny Mesh:

This 3D object demonstrates the usage the properties of haptic Surface Properties as seen in the screenshot below.

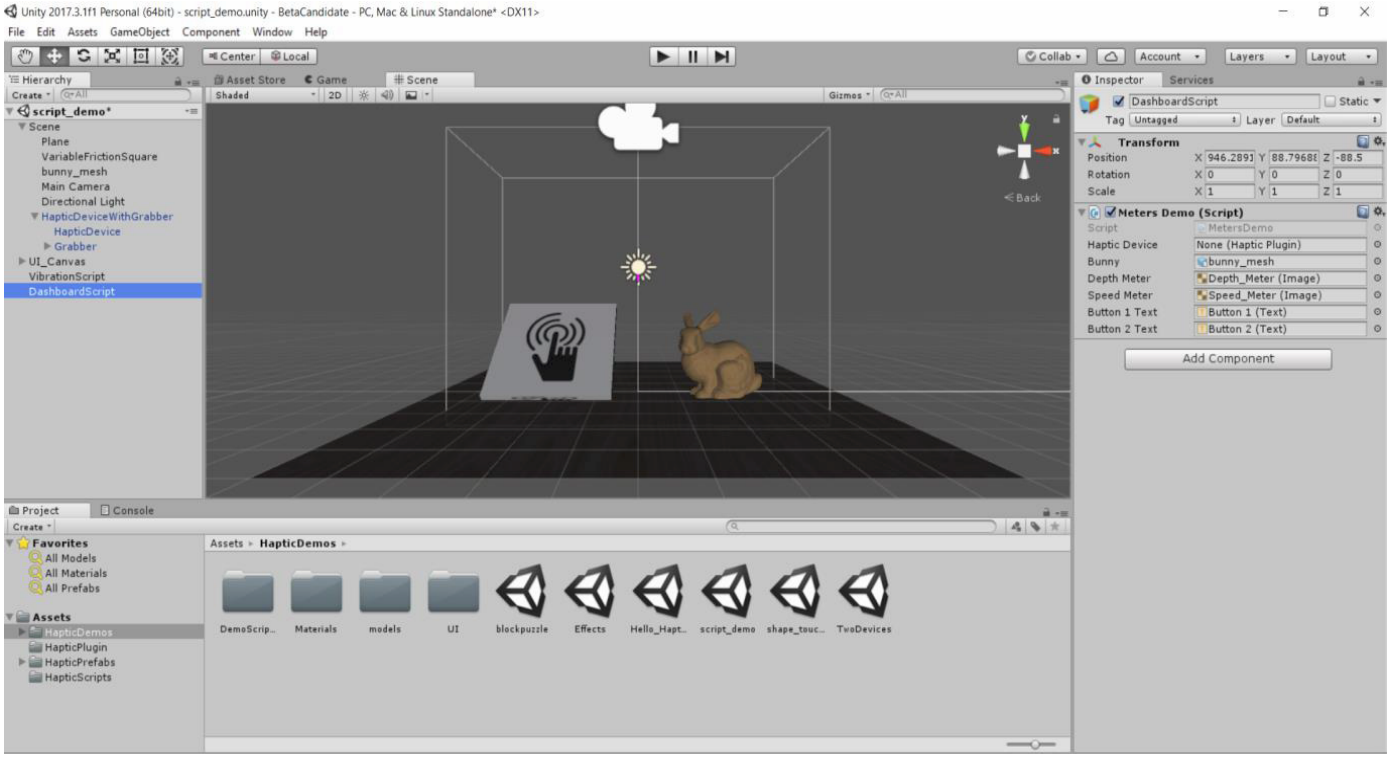


The following figures show additional scripts for user reference:

1) Vibration Script:



2) Dashboard Script



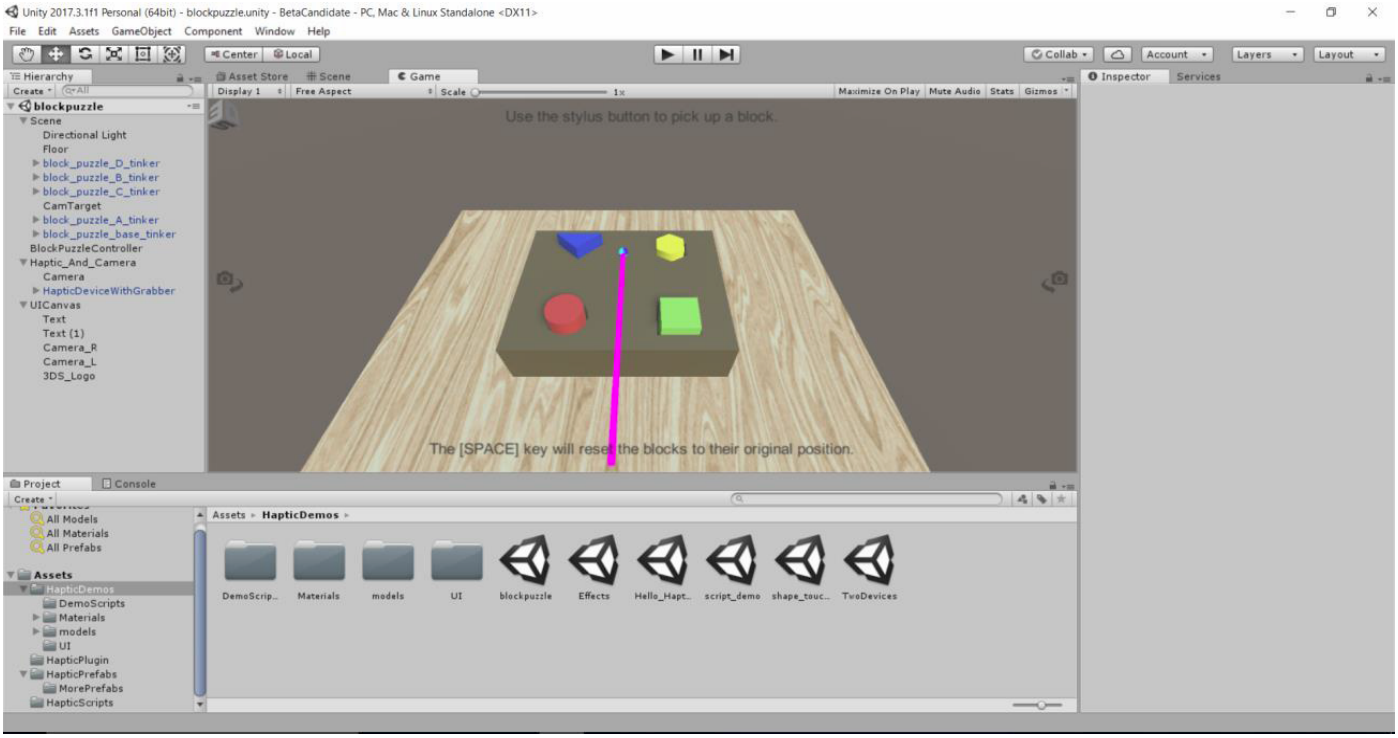
Tip: For more advanced programming details about the various scripts used in the **Scripts Demo**, users can refer to the **Scripts_Demo_Explained.pdf** found in the OpenHaptics Unity beta Plugin package (zip folder).

Run time View Of the Demo:

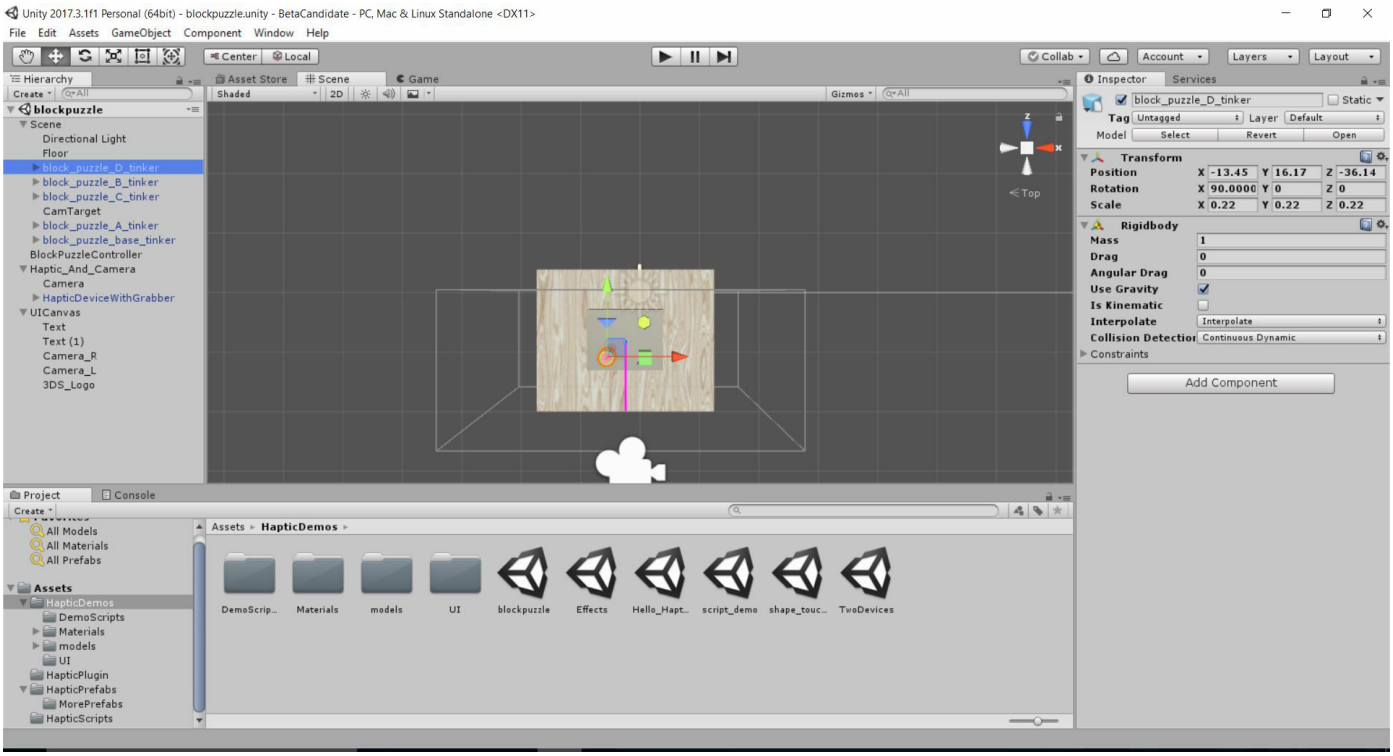
Click **Play** and experience the power of haptic scripts and the numerous opportunities to implement these properties.

Block Puzzle Demo

This demo thoroughly demonstrates the use of physics. The demo uses Unity physics. This demo encapsulates many properties discussed in the previous chapters as well as includes how to add physics to 3D objects.



The demo consist of four different shapes and associated blocks. Each block is defined as a rigid body (Unity rigid body) with appropriate properties. The camera in this demo can be adjusted and rotated. The blocks can be picked and placed using the haptic device. Refer to the the following image to view a block with its properties.

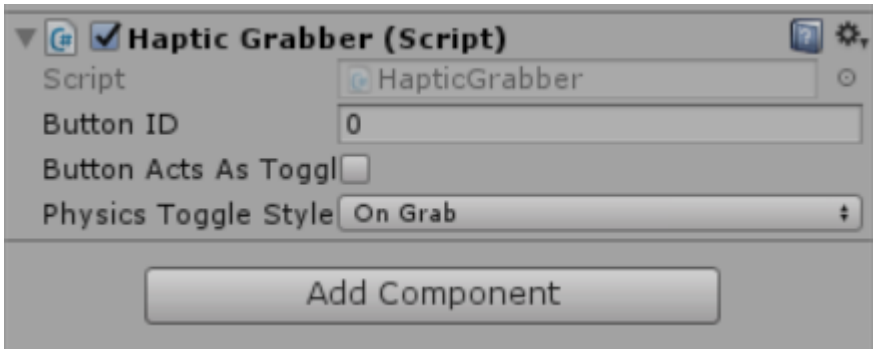


Click the **Play** button. The users can experience unity physics using the haptic devices.

The user can interact few ways with the physics objects:

- Grab/Pick the blocks using the haptic stylus and pressing the first stylus button.
- Turn or rotate the blocks and keep pressing the stylus button.
- Place the blocks into their defined places and also feel the interior cavities of the block places
- Collide the blocks
- Turn or rotate the camera. This is implemented in the script named “Camera Rotate Script.”
- The blocks can be reset using the “Space” key.

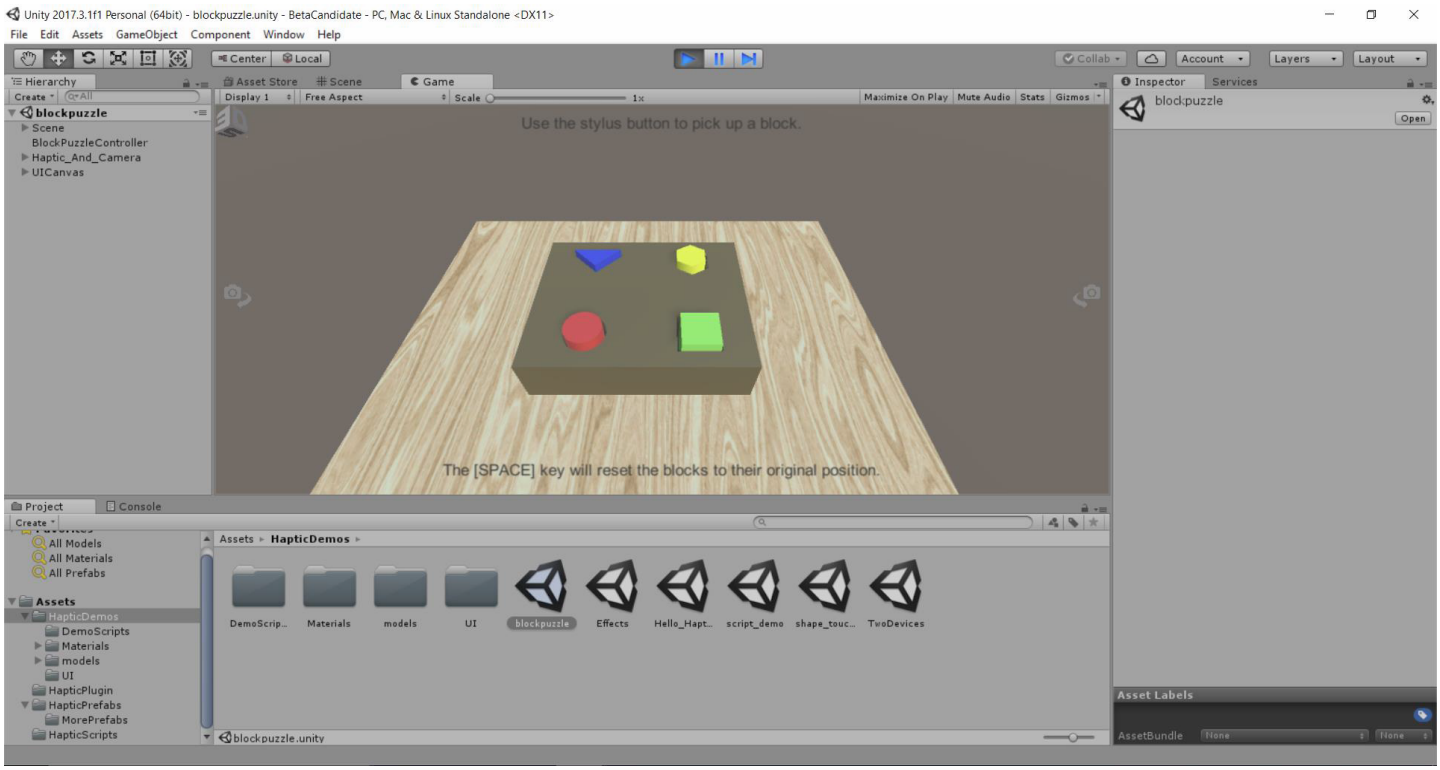
Tip: There are two haptic physics toggle modes; the user can take advantage of this through the haptic grabber script. The options are **On Touch** and **On Grab** found under **Physics Toggle Style**, seen in the following screenshot.



- **On Touch** : Allows the user only to touch and feel the objects and effects.
- **On Grab**: Allows the user grab and manipulate objects. The user can grab objects by pressing the stylus buttons. Which button to be used to grab objects can be specified by the field **Button acts as Toggle**. By default it is set to “0” referring to the first button. The user can change it to 1 and use the second button in the Touch Haptic Device. As far touch X the option will have to be only “0” as the stylus has only one button.

It is reccomended user have a clear understanding of the unity physics and appropriate properties to make full use of this demo and understand its features.

Run time View Of the Demo:

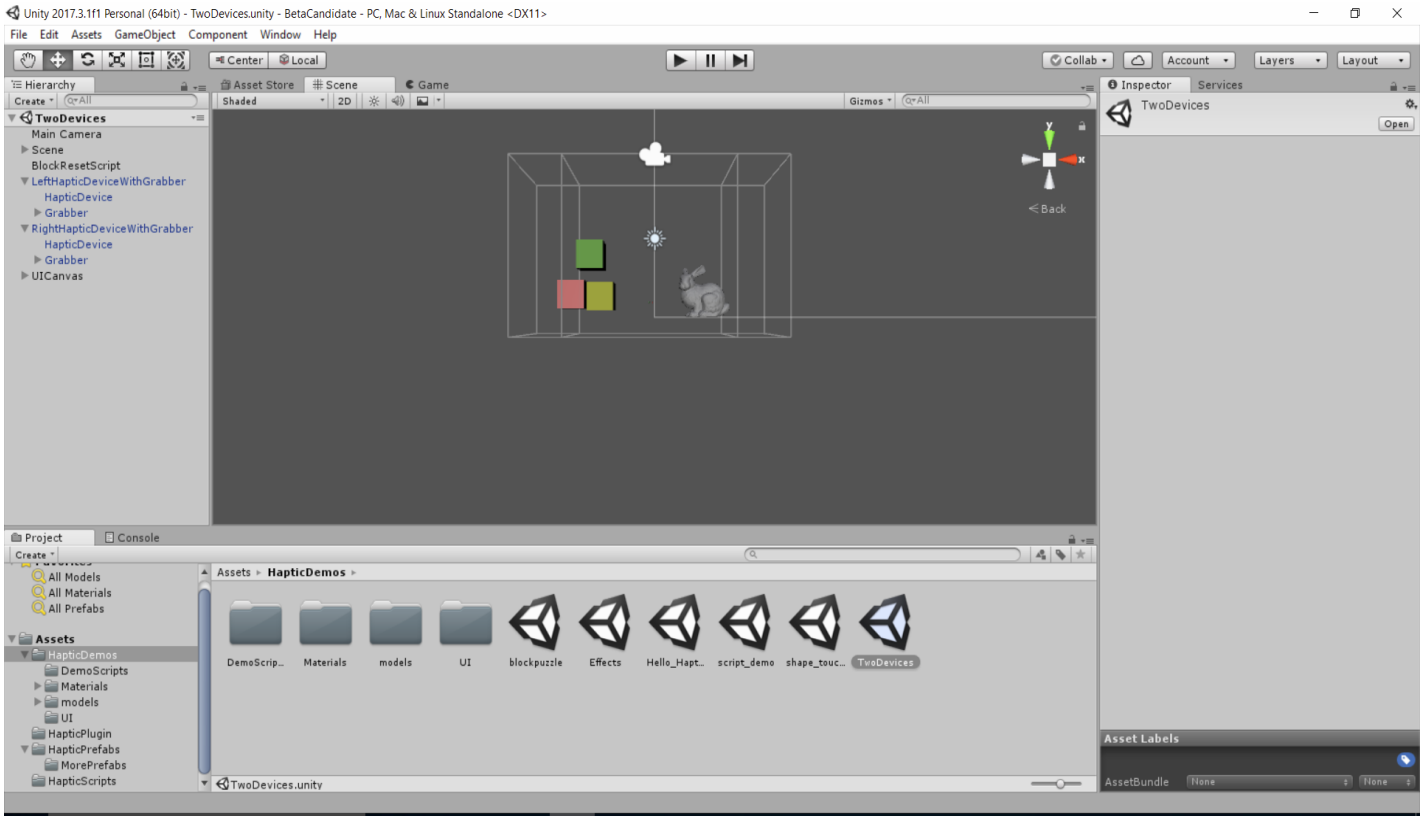


Two Devices Demo

This is a very simple demo that demonstrates the use of two haptic devices. The user must have two haptic devices connected in order to run this demo. Refer to the screenshot of the demo below.

As you can see, there are two device prefabs shown here:

- **LeftDeviceWithGrabber**
- **RightDevicewithGrabber**

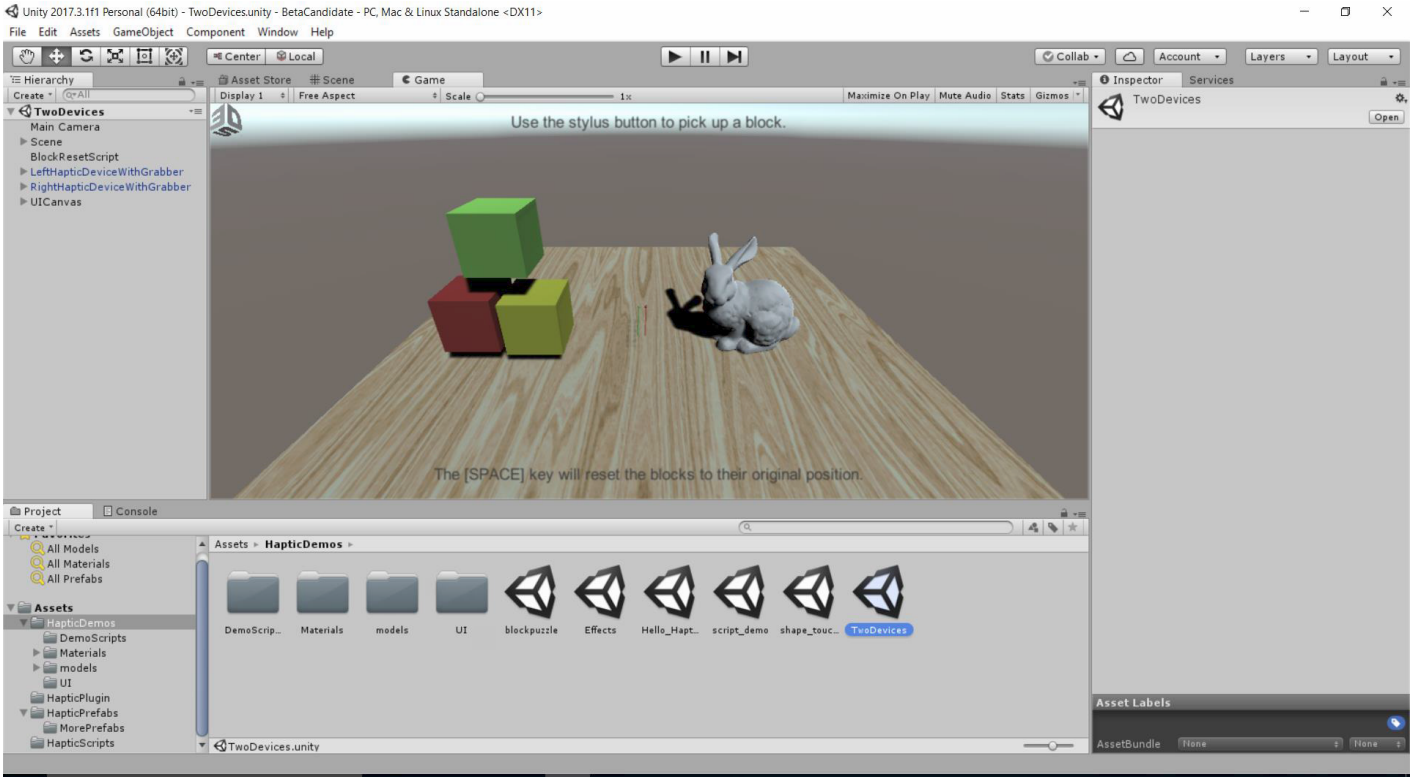


From the previous chapters, some of the functions and options are self-explanatory. The main focal point of this example is how to use 2 devices in Unity and in a demo. This demo encapsulates the basic features discussed earlier with the inclusion of 2 devices.

Tip: Users can create their own profile name for devices, as discussed in the Configuring the Devices section.

Run time View Of the Demo:

Refer to the following screenshot.



8 CHAPTER SIX – ADDING VR EXPERIENCE

The OpenHaptics Unity plugin supports adding VR experience to the user demos. A sample VR demo is included in a separate folder. In this beta release we have integrated only Oculus rift. The final release will have support for more VR headsets.

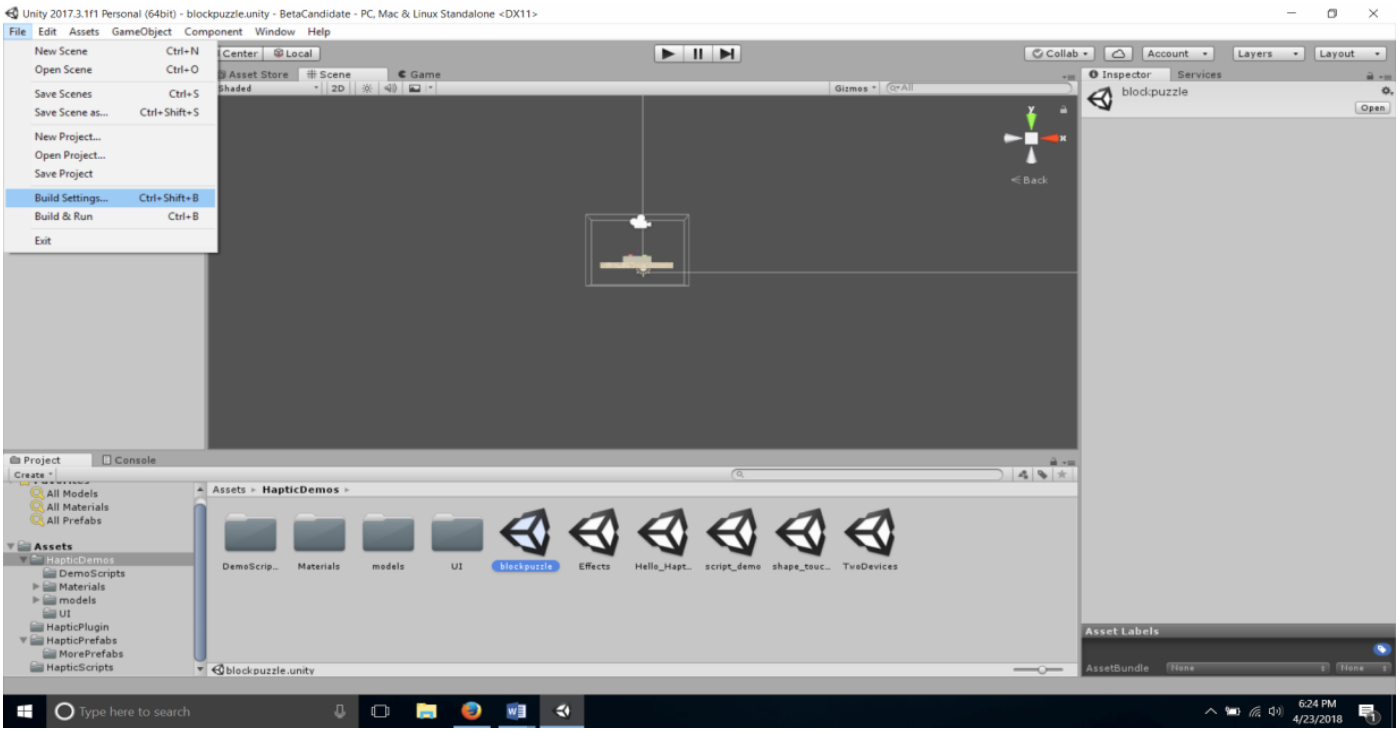
Filename of the VR demo inside the Unity beta package: **OpenHaptics.VR Demo Project**.

The guide for the VR demo is called "**VRDemo_Oculus_Guide.pdf**" and can be accessed from the OpenHaptics Unity Plugin Beta folder (zip folder).

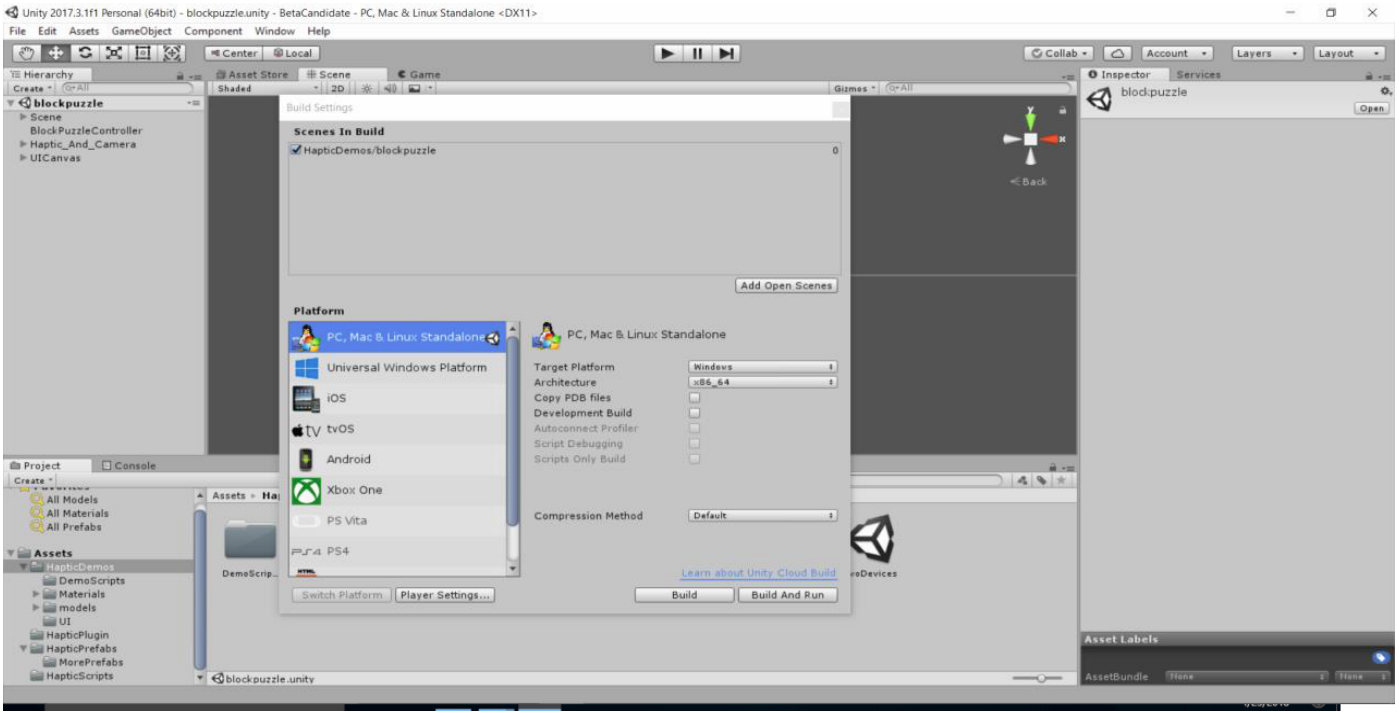
9 BUILD AND RUN DEMOS

Users can create their own demos and build the demos into an .exe file. These steps are outlined below:

1. Go to **File**-> **Build Settings**



2. Click on Add Open Scenes
3. Choose Platform -> Windows, Architecture: x86-64, Compression method: Default
4. Click on Build
5. Choose a path to save the .exe version of the demo. Refer to the following screenshot.



Appendix

I. For more help on scripting demo, refer to “Scripts_Demo_Explained.pdf”.

II. For more help on VR demos, refer to “VRDemo_Guide.pdf”.

III. For advanced documentation, refer to the oxygen generated reference guide in folder named “UnityPlugin_DoxygenDocumentation”. Just open the file named “index.html” to start using it.

IV. The demo / example executables can be found in start menu or as desktop shortcut as “OpenHapticsUnity_BetaDemos”. These are installed along with the Touch device drivers (Refer section 1. Prerequisites).

V. The OpenHaptics Unity plugin cannot be opened simultaneously with the other driver Applications like Touch Smart Setup, Touch Setup, Touch Diagnostic and Touch Demo.

VI. Users are advised to connect all devices before running any demo. This is a best practice to avoid ad hoc demo behaviors.

VII. If the haptic device is disconnected for any reason, the demo must be “stopped” completely before reconnecting the device. This way the plugin can effectively handle the connections.

VIII. It is advised that the users close the unity editor before running any of the built demo (.exe version of the demo created)

IX. As a prerequisite the OpenHaptics Unity plugin does not support any Intel graphics cards.

3D Systems, Inc. 333 Three D Systems Circle Rock Hill, SC 29730
www.3dsystems.com