Intro to Unity

Hands-On Challenges

Introduction to Unity Hands-On Challenges

Challenge P: Build and Run

At long last, you've come to the end of the line. You should have a game that is ready for new players. Your job now is to build the game and run it on a device.

Getting Started

Open the starter challenge project. If you've been following this series from the beginning, open your last saved project.

> **Note:** Unity will open to empty scene. Find the scene that you saved, or if you are using the Starter Project, open Main.scene.

Once you have the game set, build out a project, and put it on a device! Congrats, you're ready to submit to the app store!

Uber Haxx0r Challenge

So, if you've moved your game to a touch device, then you will have noticed that the game is entirely unresponsive to touch controls. This creates a bit of problem for your user.

Your job is to implement touch control.

First, check out the API documentation over here:
http://docs.unity3d.com/ScriptReference/Input.html

Next, you have to keep in mind that a touch point is returned in screen points whereas your objects exist in world space. You have to convert that touch from touch to world space.

Next, you have to make sure the correct object is selected. You will have to raycast from the touch point and if it hits the paddle, then you know the users is "touching" it.

Next, you have to move the paddle with the user's finger. To do that, you have to find out the finger position in world space. Here's some code that will do that for you:

```
Ray ray = Camera.main.ScreenPointToRay(Input.GetTouch(0).position)
;
Plane plane = new Plane(Vector3.up, transform.position);
```

```
float distance = 0;
if (plane.Raycast(ray, out distance)){
    Vector3 pos = ray.GetPoint(distance);
         transform.position = new Vector3 (pos.x ,
              transform.position.y, transform.position.z);
}
```

Finally, make sure the paddle is deselected when the user lifts his or her finger off the device. Like iOS, touches in Unity have several different states. You will have to check against the `TouchPhase` state in the documentation.

http://docs.unity3d.com/ScriptReference/TouchPhase.html

Of course, you should have the user start the game with a tap. Then, build and run on your device, and you should be able to run it in a mobile world.

> **Note:** There are some current issues with Unity 4.6 beta so expect to encounter some weirdness.