

## Unity For Flash Developers Lesson - Tutorial 6 (Basic Physics)

In Flash, if you want to do something like simulate gravity or employ basic physics, you have to either roll your own code to do this or use one of the Flash physics engines out there like:

<http://box2dflex.sourceforge.net/>

<http://www.cove.org/ape/>

Although this gets the job done, it takes time and significant effort to get the job done.

### **Adding a Rigidbody component to enable Physics**

In Unity, if you want physics, you more or less just enable it.

One of the easiest ways to enable physics in Unity is to add a Rigidbody component to one of your game objects.

Once a Rigidbody has been added to your game object, things like gravity, and collision will become possible.

### **Use the Force Luke ... or whatever your name happens to be :-)**

If you added a Rigidbody to one of your game objects and hit the play button, you'll most likely notice it starts to fall, and if there is nothing for it to collide with, it will fall into the abyss.

If you don't want to fall into the abyss, you need to stop it from falling by putting something in its path it can collide with, like a simple plane with a mesh collider on it.

Now when you test your movie, you should see the game object fall and collide with your plane and eventually come to rest.

One of the cool things you can do with your physics-enabled game objects now is to apply forces to it.

will be using `AddForce`, to accomplish this.

<http://unity3d.com/support/documentation/ScriptReference/Rigidbody.AddForce.html>

Add the forces.js script to you camera or other gameObject of choice.

## //forces.js

```
var myRigidBody: Rigidbody;
```

```
function OnGUI () {  
    if (GUI.Button (Rect (10,10,150,100), "Apply Upward Force")) {  
        myRigidBody.AddForce (Vector3.up * 200);  
    }  
  
    if (GUI.Button (Rect (10,150,150,100), "Apply Force pos X")) {  
        myRigidBody.AddForce (100, 0, 0); //postive x direction  
    }  
  
    if (GUI.Button (Rect (10,300,150,100), "Apply Force neg X")) {  
        myRigidBody.AddForce (-100, 0, 0); //postive x direction  
    }  
  
    if (GUI.Button (Rect (10,450,150,100), "Apply Force in X and Y"))  
    {  
        myRigidBody.AddForce (100, 100, 0); //projectile motion!  
    }  
}
```