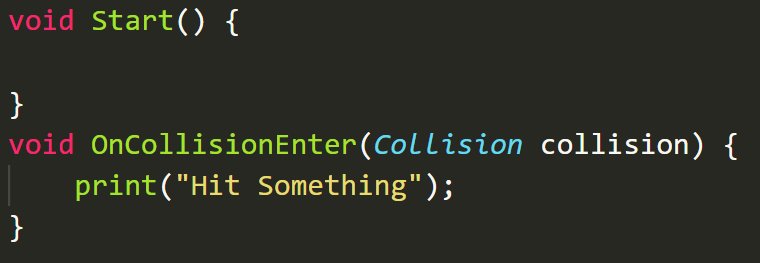
**Triggers and Collisions**

**Objective:** in this lesson we’re going to have the ship trigger messages every time they hit something.

Remember in the last project we had our player colliding with the terrain. Lets look at the code there. Specifically the **OnCollisionEnter** method.



1. So we’re going to copy this method to our **Player.cs** script right under the Start Method and only have it display a message.



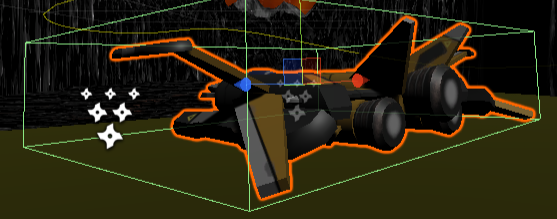
1. Lets go to Unity and hit play and try to trigger this message. **We should notice that we can’t get this message to display.**

So collision messages don't just automatically and magically get passed. We were lucky in the last project, but in this game, we're going to have to understand collisions a bit better.

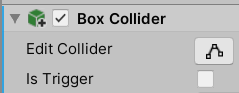
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Static Collider** | **Rigidbody Collider** | **Kinematic Rigidbody Collider** | **Static Trigger Collider** | **Rigidbody Trigger Collider** | **Kinematic Rigidbody Collider** |
| **Static Collider** |  | Collision |  |  | Trigger | Trigger |
| **Rigidbody Collider** | Collision | Collision | Collision | Trigger | Trigger | Trigger |
| **Kinematic Rigidbody Collider** |  | Collision |  | Trigger | Trigger | Trigger |
| **Static Trigger Collider** |  | Trigger | Trigger |  | Trigger | Trigger |
| **Rigidbody Trigger Collider** | Trigger | Trigger | Trigger | Trigger | Trigger | Trigger |
| **Kinematic Rigidbody Collider** | Trigger | Trigger | Trigger | Trigger | Trigger | Trigger |

There are 2 types of collisions. The ones we’ve seen before in yellow are your traditional collision and in green are Triggers. Lets understand the difference.

1. Lets go to our **Spaceship** and click **Add Component**. Look for **Box** **Collider**.

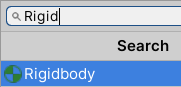


So when you add a box collider to something, you've got an option for **Is Trigger** or not.

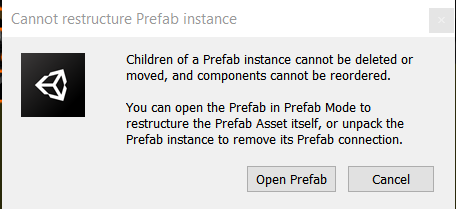


If it's not a trigger, then physics will be processed, and it will bounce off things. If it is trigger, physics will not be processed.

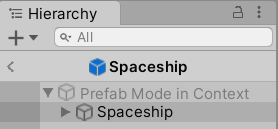
Having said that, you won't have any physics working, unless you have a **Rigidbody** component that we've added previously in the last game.

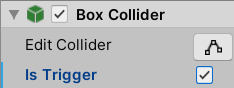


1. Lets **add a RigidBody component** to our ship.
2. Lets reorder our components. First lets **Apply all the overrides to our prefab**. Move the Player script to the bottom. You’re going to get a message telling you the prefab connection is going to get broken.



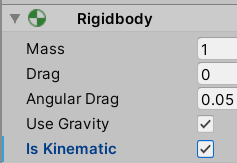
1. Click **Open Prefab** and on this screen move the script to the bottom. When Done click on the Arrow at the top left.



1. Minimize every component except **Box Collider** and **RigidBody**.
2. If we check the box, it's a trigger. 
3. If it has a **rigid body** that's on it, it is a **RigidBody Collider**



1. if the rigid body has kinematic selected, it's a **Kinematic RigidBody Collider**. DON’T SELECT KINEMATIC RIGHT NOW.

.

**Rigidbodies** provide physics to the object, **kinematic** means the study of movement without the physics. Basically, it's saying that we move this thing in script, we're not letting the physics engine move it.

1. Lets look at our Table again

If either of the GameObjects *(look at the Column and Row names)* have a trigger *(the ones in green)*, then you're dealing with triggers. The moment you put a trigger on either of them, you get a trigger event, an **OnTriggerEnter** event.

Link to Documentation

<https://docs.unity3d.com/ScriptReference/Collider.OnTriggerEnter.html>

And it's very, very similar to **OnCollisionEnter**. The only difference is instead of a collision coming through, you get a collider, a slightly simpler piece of information. All you get told is about the collider that you collided with, rather than details about the collision and the collision velocity, and all that stuff, all right?

But the fundamental difference between the two, apart from you get different information, is no physics is processed with the trigger. So if there's a **trigger** on either row or column, you get a trigger, that's the pattern.

And if there is not a trigger on either, then you're either a **Static Collider** which is is any collider that does not have a Rigidbody on it. if I was to take the rigid body off the ship then it would be a Static Collider.

And then the next thing you see is a **Rigidbody Collider**, that's saying that I have a Rigidbody on me.

And the next one is **Kinematic Rigidbody** which again just means the study of movement without the physics. Basically, it's saying that we move this thing in script, we're not letting the physics engine move it.

Okay, so once you learn to read this table, then we can say to ourselves, okay, why aren't we colliding, let's say, with the terrain?

Well, let's take a look. What is the ship right now? Well, right now, that I've added a **Rigidbody** onto it, it is a **Rigidbody Collider**, not Kinematic.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Static Collider** | **Rigidbody Collider** | **Kinematic Rigidbody Collider** | **Static Trigger Collider** | **Rigidbody Trigger Collider** | **Kinematic Rigidbody Collider** |
| **Static Collider** |  | Collision |  |  | Trigger | Trigger |
| **Rigidbody Collider** | Collision | Collision | Collision | Trigger | Trigger | Trigger |
| **Kinematic Rigidbody Collider** |  | Collision |  | Trigger | Trigger | Trigger |

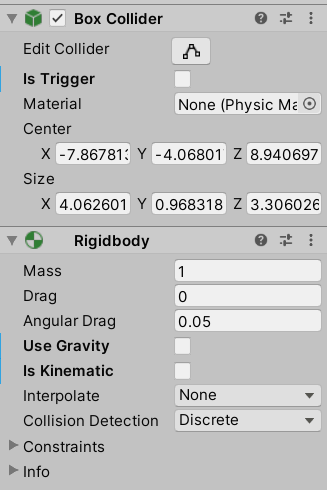
So under what circumstances would I get a collision or trigger? Well, any circumstances, actually, whatever the thing that I'm providing with, as long as it has a collider on it.

1. So you check on your terrain, terrain collider is turned on.

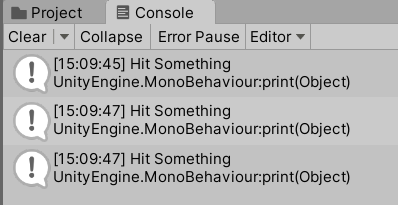


So just by adding a Rigidbody to the ship, we should now be colliding with a terrain.

1. Make sure this is how your Spaceship looks right now. **Turn Gravity off** or else your ship starts falling off screen.



1. Press play and collide with terrain to see the message



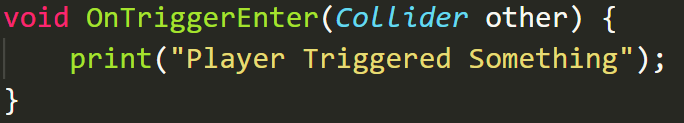
Right now our ship will fly through the hill after a detection. I don't think we want that. I think we want the ship to just have receive a trigger message, and then to be a total blow up.

1. We need to go from it's Rigidbody Collider hitting a Static Collider of the terrain, well, the terrain in this first column is always a Static Collider, it can't be anything else. So what are our options?

|  |  |
| --- | --- |
|  | **Static Collider** |
| **Static Collider** |  |
| **Rigidbody Collider** | Collision |
| **Kinematic Rigidbody Collider** |  |
| **Static Trigger Collider** |  |
| **Rigidbody Trigger Collider** | Trigger |
| **Kinematic Rigidbody Collider** | Trigger |

In order for our collisions to turn into a trigger we need to change the Spaceship for RigidBody Collider into a RigidBody or Kinematic RigidBody

1. Turn on **Is Kinematic** and **Is Trigger**
2. Open your **Player**.**cs** script and lets add the **OnTriggerEnter** method under the **OnCollisionEnter** method.



1. Play test it and make sure the message is popping up.

**CHALLENGE:**

* Which cell are we in for enemy interaction?
* What do we need to change to get messages?
* The message should print to the console.

**Solution**

1. Find the folder where your enemy ships are and **select them all**

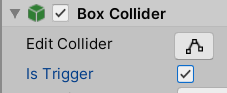


If you ever can’t find it right click on **Prefab** in Hierarchy and select **Prefab> Select Asset**

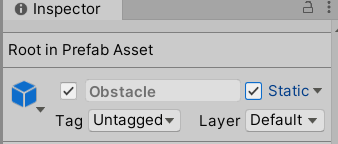
1. Click **add component** and find **Box Collider**
2. Now we know our Ship is a **Kinematic RigidBody Collider** so what do our enemies need to be to trigger? Well anything lol

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Static Collider** | **Rigidbody Collider** | **Kinematic Rigidbody Collider** | **Static Trigger Collider** | **Rigidbody Trigger Collider** | **Kinematic Rigidbody Collider** |
| **Kinematic Rigidbody Collider** | Trigger | Trigger | Trigger | Trigger | Trigger | Trigger |

1. I’m going to **turn ON** the **Is Trigger** option for all the enemy ships cause I don’t want them exhibiting physics.



1. Now if we press Play and crash through our enemy ships we should see the Triggered message.
2. Lets make sure our obstacles have the same thing. Select on your Obstacle (the horizontal rock) and find in Assets. Add a **Box Collider** to it. Also mark the Obstacle as **Static** on the top right. I'm going to mark these as static, cause they're not going to be moving. And that's just a lighting, etc., optimization you can see here. But in general, if things aren't moving, you mark them as static.



1. Press **Play** and now everything should be Triggering the Triggered message.
2. Delete the **OnCollisionEnter** method.