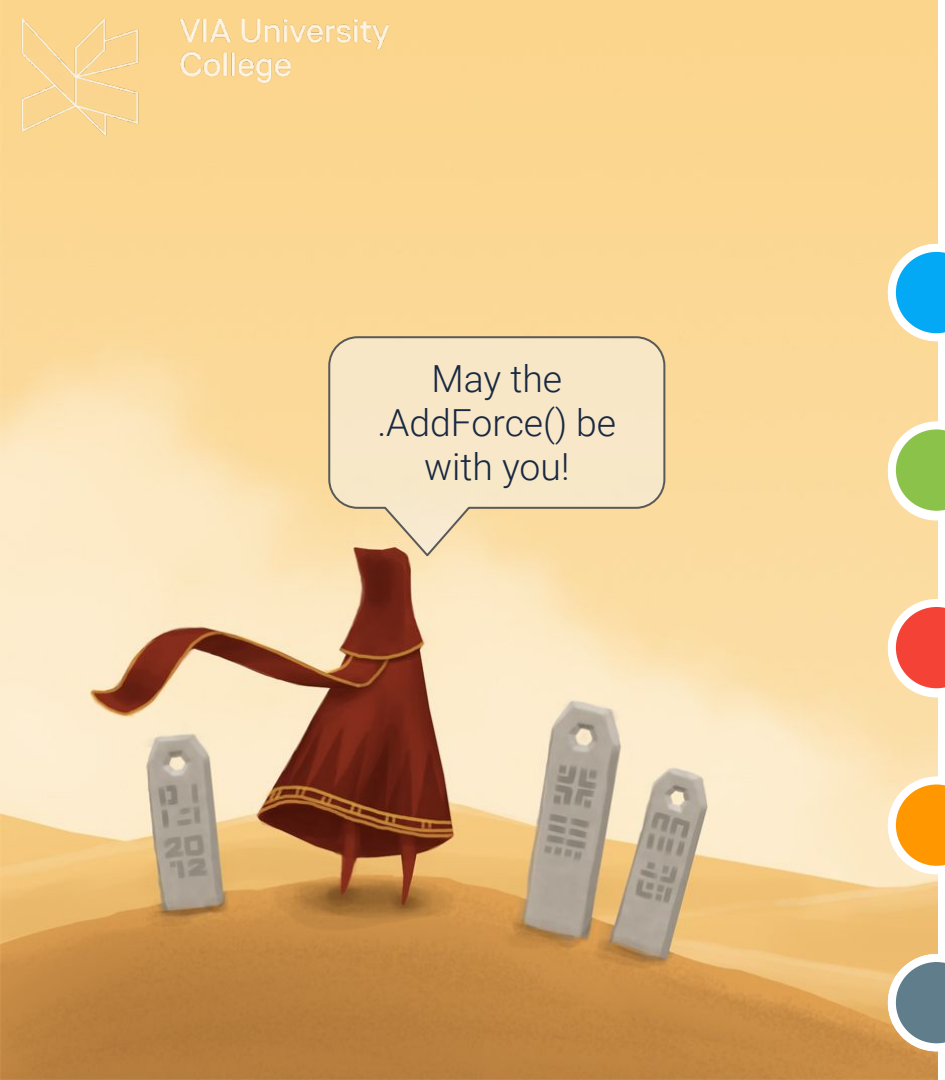
VIA University College



Physics

Handling physics simulations in Unity

Basic Physics in Unity

Rigidbodies, FixedUpdate, colliders and physic materials

Physics Events

See what Unity event functions the physics system exposes

Layers

Get more control over which objects can interact with each other

Raycasting

Cast rays using vectors and react according to what you hit

Exercises

Basketball, Volleyball, Moon Lander, Billiard and more!

Last Week

What is a Vector?

What can Vectors be used for?

How are Vectors used in Unity?

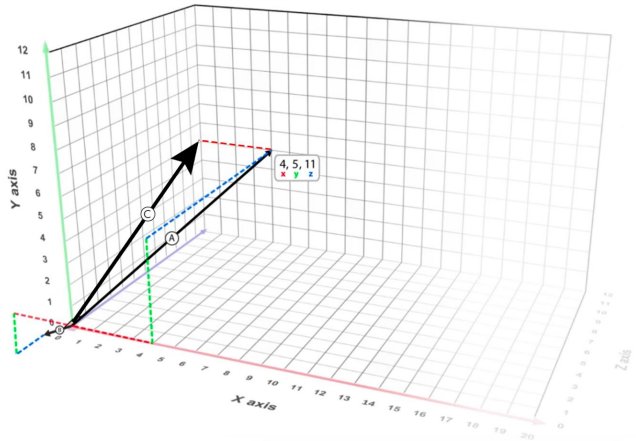
What is Linear Interpolation?

How do we do basic input in Unity?

What types of input does Unity support?

What is the Input Manager?

What is an axis?



What is Physics?

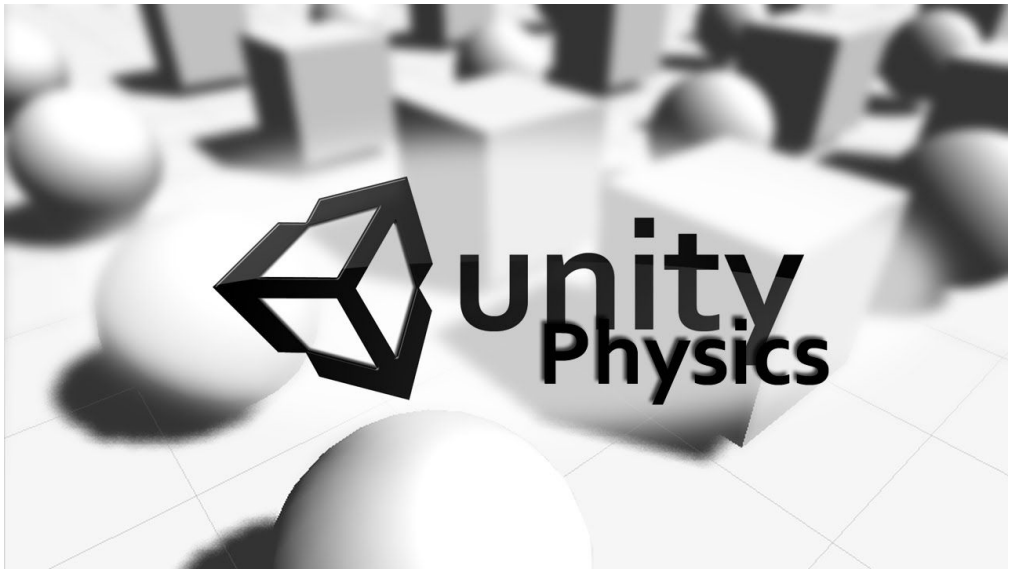
Forces:

- Gravity
- Push
- Something bumps into something else
- Directional
- Rotational
    - Both are just vectors

Built-in physics engine

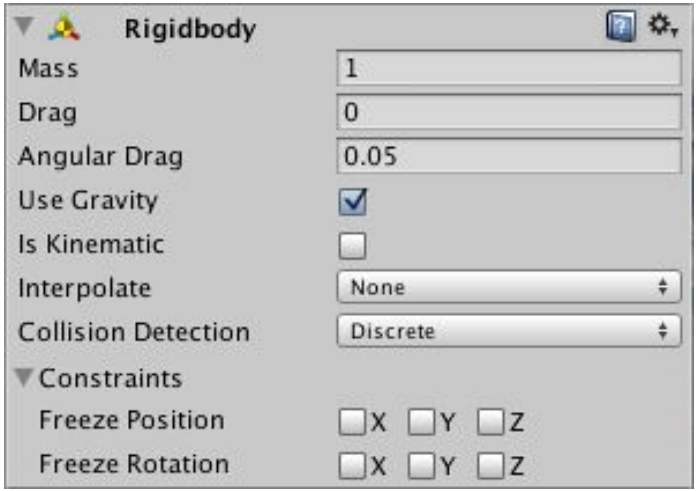
Unity's built-in physics engines provide components that handle the physical simulation for you.

There are actually two separate physics engines in Unity (3D & 2D)

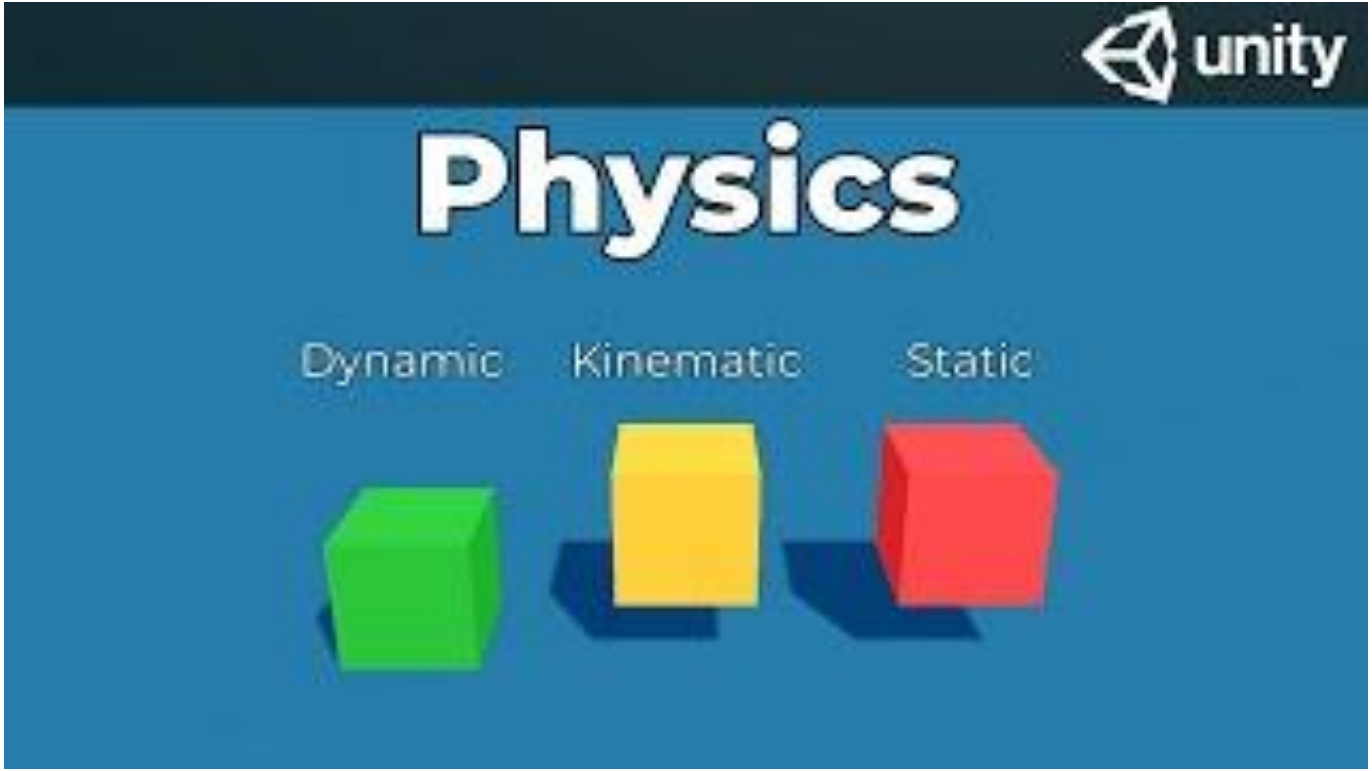


Rigidbody

- Enables GameObject to be affected by physics
- Commonly used to enable gravity on objects and collision between objects
- Mass, drag and angular drag
- Check “Is Kinematic” if you don’t want this GameObject to be affected by the physics engine. (Other objects will still be affected by it though!)
- Constraints: Limit how your object will be affected
- Physics settings: Edit > Project Settings > Physics

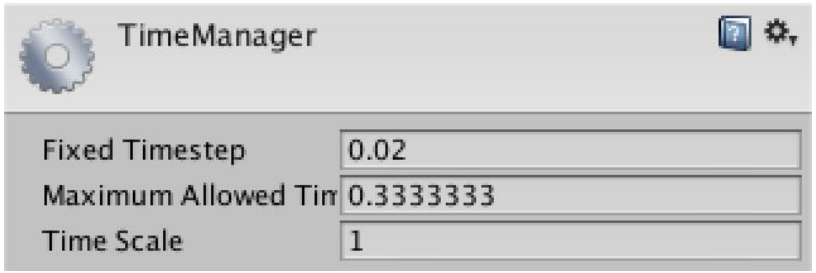


Physics Objects



FixedUpdate

- Event function used for all code that utilizes physics.
- Called at a set interval (frame rate independent)
- But remember - it is not guaranteeing per second movement - you still need deltaTime for that.



FixedUpdate

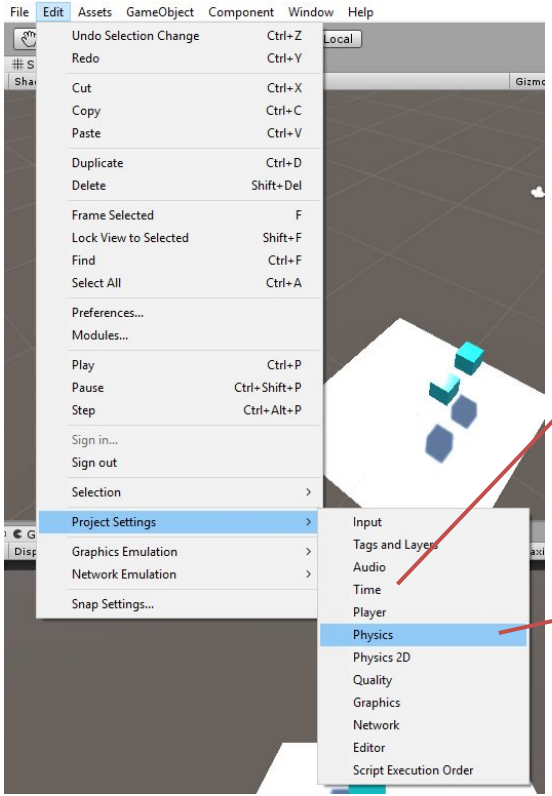
FixedUpdate, Example

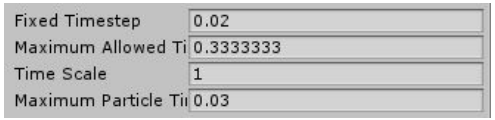
If you wanted to do your own gravity, you would add a force in FixedUpdate:

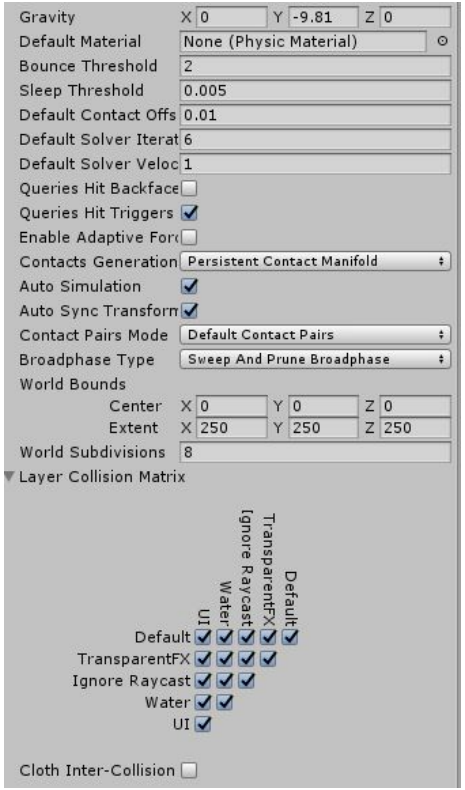
```
void FixedUpdate() { // called at a set interval
    rb.AddForce(Vector3.down * 30f); // gravity of 30, usually 9.82f
}
```

Physics Settings

Settings for gravity and FixedUpdate can be changed

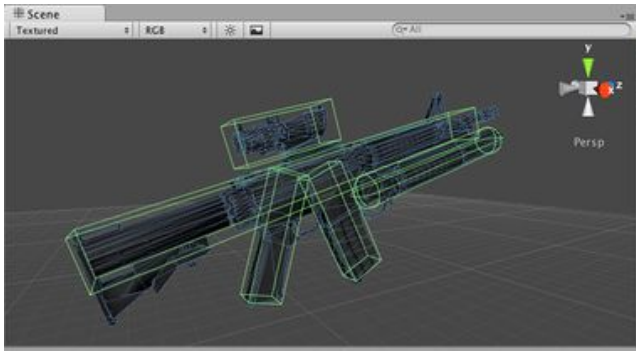


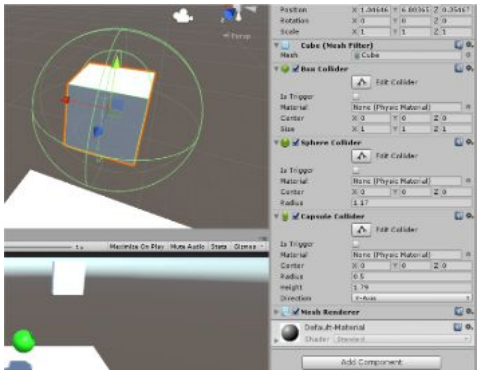




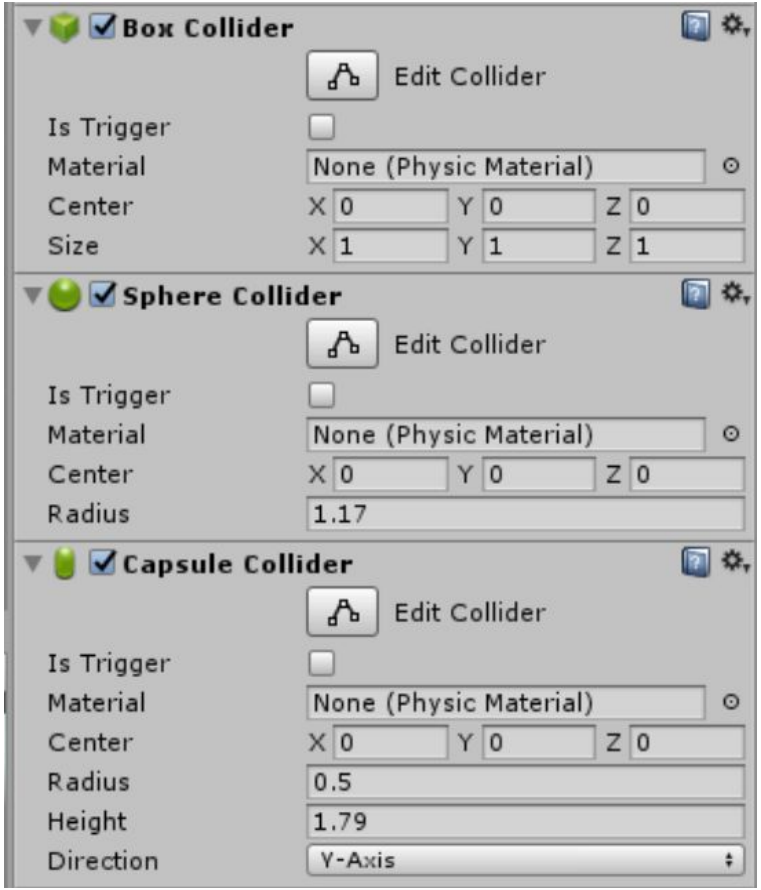
Colliders

- The **Rigidbody** component **needs a collider** component to interact with other physics objects.
- Colliders come in various shapes and types
  - Sphere, Capsule, Box, Mesh, etc...
- Keep Colliders as simple in geometry as possible!
- Child GameObjects with Colliders affect Parents with Rigidbodies.

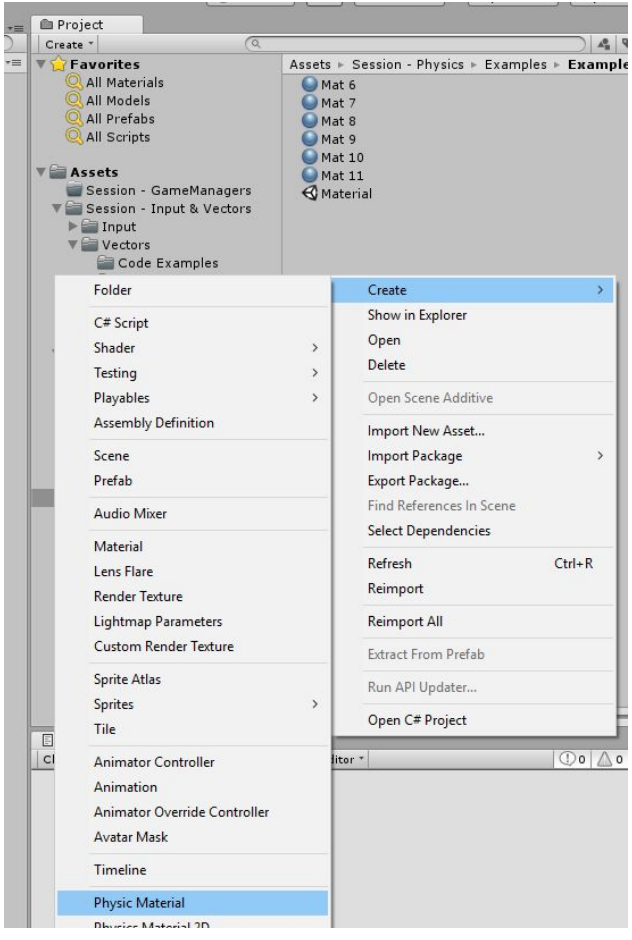




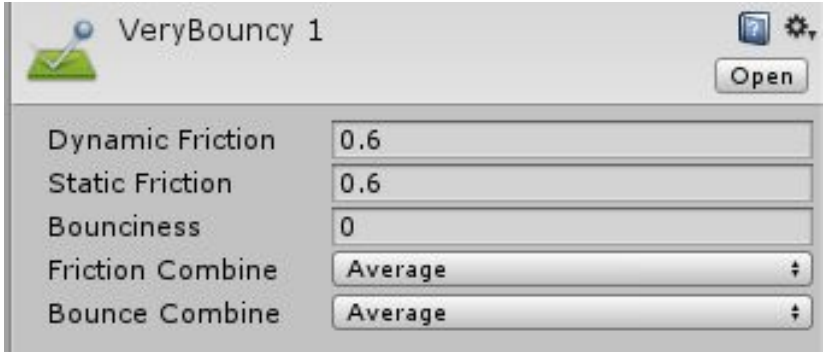
Colliders


Box Collider
Edit Collider
Is Trigger
Material
None (Physic Material)
Center
X 0 Y 0 Z 0
Size
X 1 Y 1 Z 1
Sphere Collider
Edit Collider
Is Trigger
Material
None (Physic Material)
Center
X 0 Y 0 Z 0
Radius
1.17
Capsule Collider
Edit Collider
Is Trigger
Material
None (Physic Material)
Center
X 0 Y 0 Z 0
Radius
0.5
Height
1.79
Direction
Y-Axis

Physic Materials



Physic materials are used to define how two objects interact, when they collide



Adding Physics Forces

To move GameObjects through physics, use methods of the Rigidbody Component attached to the GameObject you want to move.

- **AddForce(Vector):**
  - Starts the movement of a physics object, or change the speed or the direction of its movement.
  - Continuously adding force -> increase velocity
  - Stop adding force -> decrease velocity (if you have drag, gravity...)
  - ForceMode: Force (default), Impulse, Acceleration, VelocityChange

- **Bonus info:**
  - Use AddRelativeForce() to add force relative to local coordinate system
  - Rigidbody.velocity can be used to change the velocity directly instead of just adding to the current force.

Adding Physics Torque

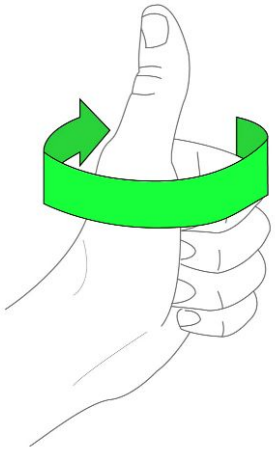
- **AddTorque(Vector3)**

Torque sounds fancy, but it's just rotation through physics!

It is similar to addForce, but rotates the object clockwise around a defined vectors axis.

Remember: Tweak the angular drag to change how much force you need to rotate an object!

Left Hand Screw Rule



Careful!

- The transform-methods you learned last week teleports game objects from one location to the next with no intermediate positions being calculated. This can be dangerous if mixed with the physics system!

- If you want more controlled movement, especially suited to kinematic rigidbodies:
  - MovePosition(Vector)
  - MoveRotation(Quaternion)

Collision Events

- When two Colliders collide, 3 types of collision events are called by Unity:
  - One for the first frame of the collision (Enter)
  - One for the continuous contact (Stay)
  - One for losing contact (Exit)

The Collision contains data to extract.

For instance “col.gameObject” to get the GameObject your GameObject collided with.

```
void OnCollisionEnter(Collision col) {
    //Called the frame the collision occurs
}

void OnCollisionStay(Collision col) {
    //Called every physics step the colliders stay in contact
}

void OnCollisionExit(Collision col) {
    //Called once the colliders loses contact
}
```

Colliders as Triggers

- Triggers are zones that can be entered to trigger some code.
- Convert a Collider to a Trigger by checking “Is Trigger”.
- When a Collider is a Trigger, other objects cannot bump into it.

These events are called when other colliders overlap with the Trigger:

- OnTriggerEnter(Collider other)
- OnTriggerStay(Collider other)
- OnTriggerExit(Collider other)

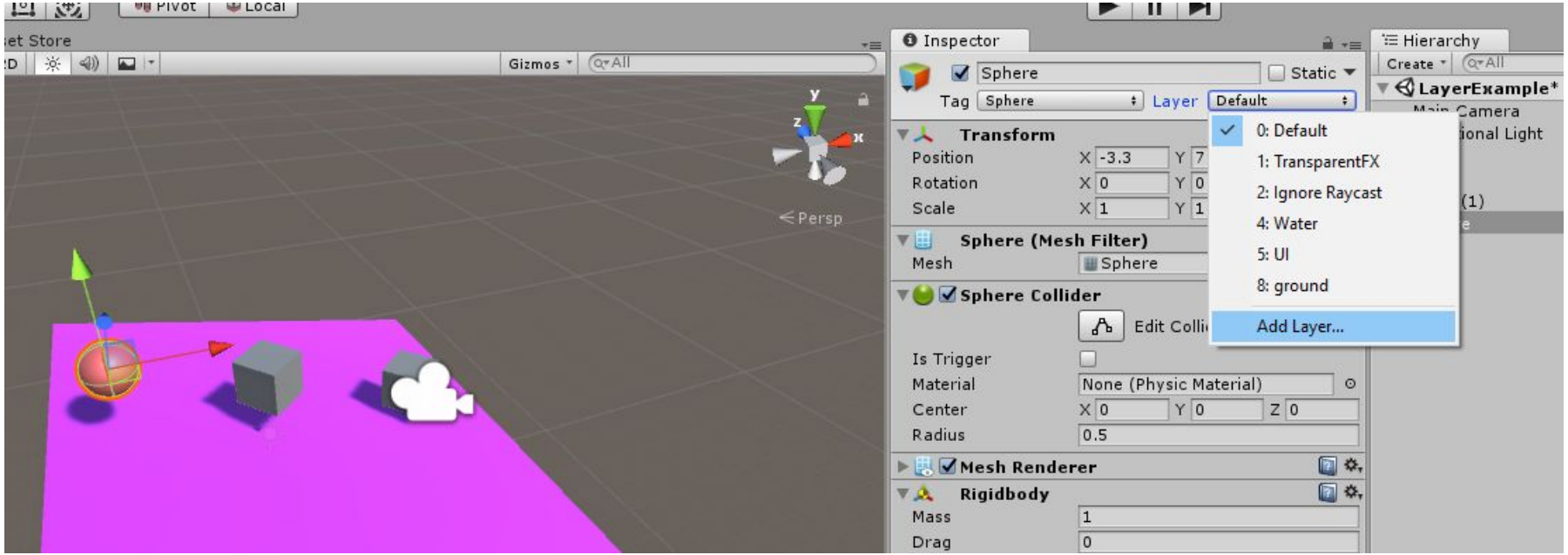
Notice the similarity to the event functions of standard collisions, but notice that the argument here is a Collider and not a Collision!

As with standard collisions - one of the colliding objects must have a rigidbody.

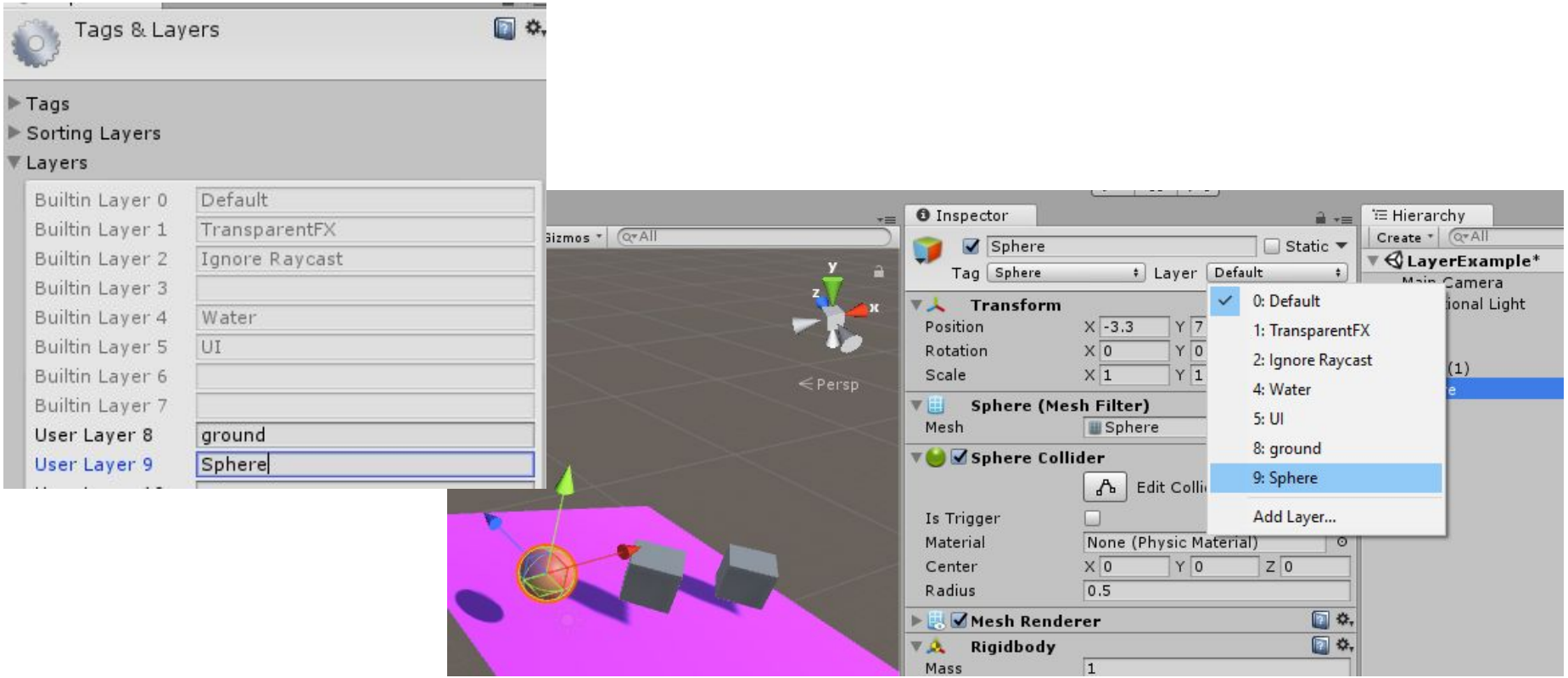
Layers

Layers are used to define which objects can interact with each other

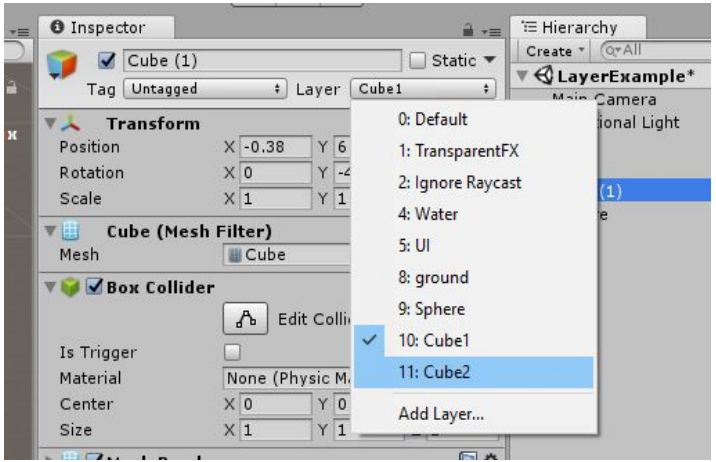
You can attach a Layer to an object:

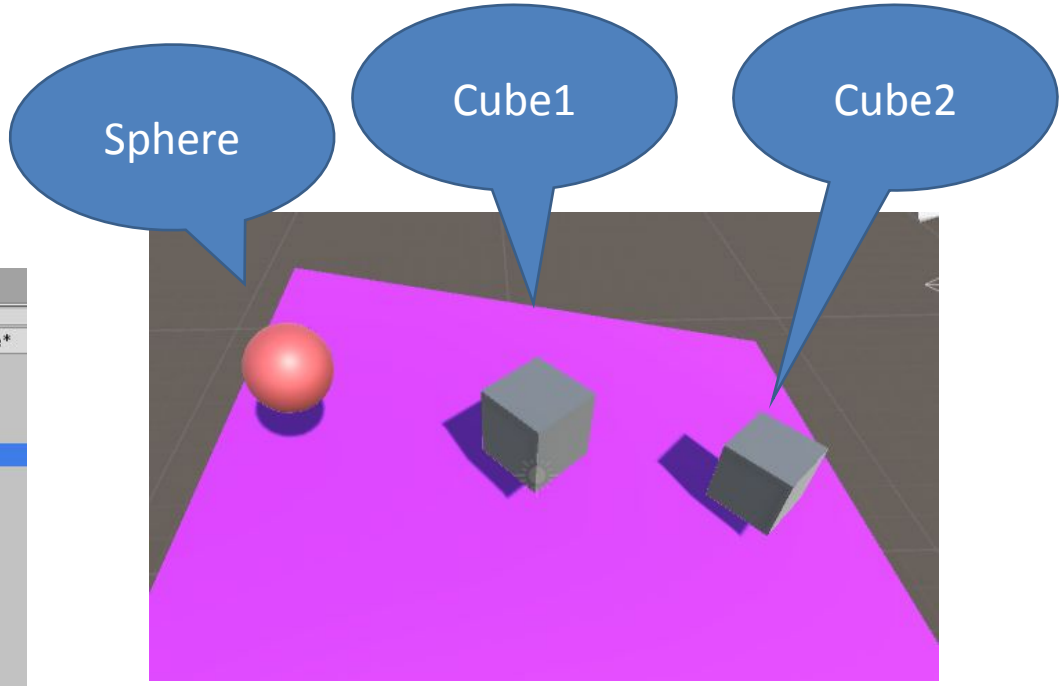


Layers

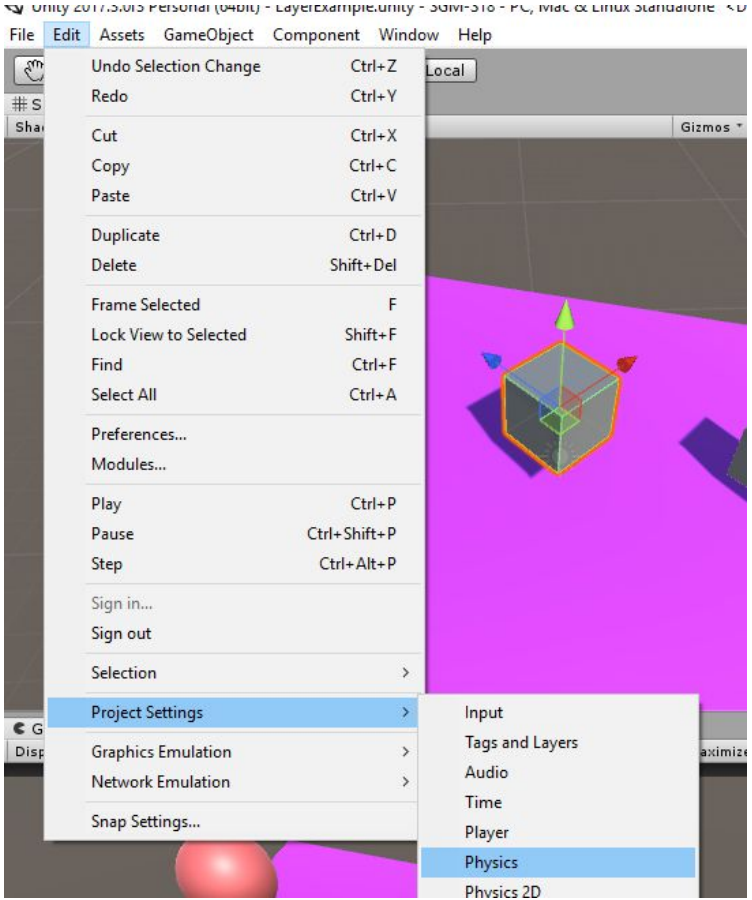

Tags & Layers
Tags
Sorting Layers
Layers
Builtin Layer 0
Default
Builtin Layer 1
TransparentFX
Builtin Layer 2
Ignore Raycast
Builtin Layer 3
Builtin Layer 4
Water
Builtin Layer 5
UI
Builtin Layer 6
Builtin Layer 7
User Layer 8
ground
User Layer 9
Sphere
Persp
Inspector
Sphere
Static
Tag
Sphere
Layer
Default
Transform
Position
Rotation
Scale
Sphere (Mesh Filter)
Mesh
Sphere
Sphere Collider
Edit Colli
Is Trigger
Material
None (Physic Material)
Center
Radius
0.5
Mesh Renderer
Rigidbody
Mass
1
0: Default
1: TransparentFX
2: Ignore Raycast
4: Water
5: UI
8: ground
9: Sphere
Add Layer...
Hierarchy
Create
LayerExample*

Layers

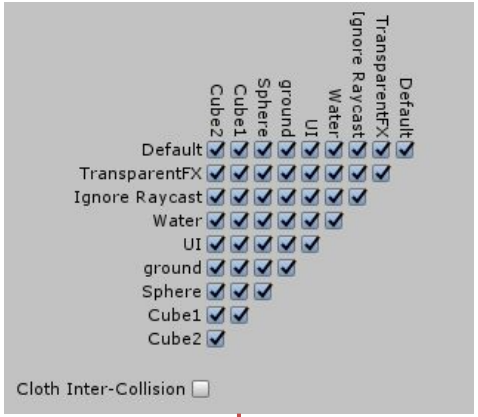

Inspector
Cube (1)
Static
Tag
Untagged
Layer
Cube1
Transform
Position
Rotation
Scale
Cube (Mesh Filter)
Mesh
Cube
Box Collider
Edit Colli
Is Trigger
Material
None (Physic M
Center
Size
Hierarchy
Create
LayerExample*
0: Default
1: TransparentFX
2: Ignore Raycast
4: Water
5: UI
8: ground
9: Sphere
10: Cube1
11: Cube2
Add Layer...

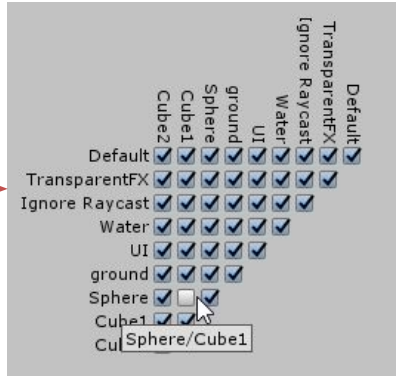

Sphere
Cube1
Cube2

Define Which Layers Can Interact



Layers





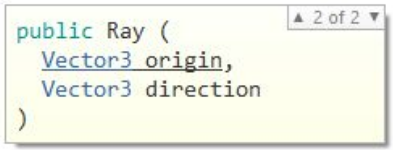
What is a Ray

- We know about vectors:

```
Vector3 vec = new Vector3 (1f, 2f, 0f);  // direction
Vector3 point = new Vector3 (-1f, 0f, 2f);  // point
```

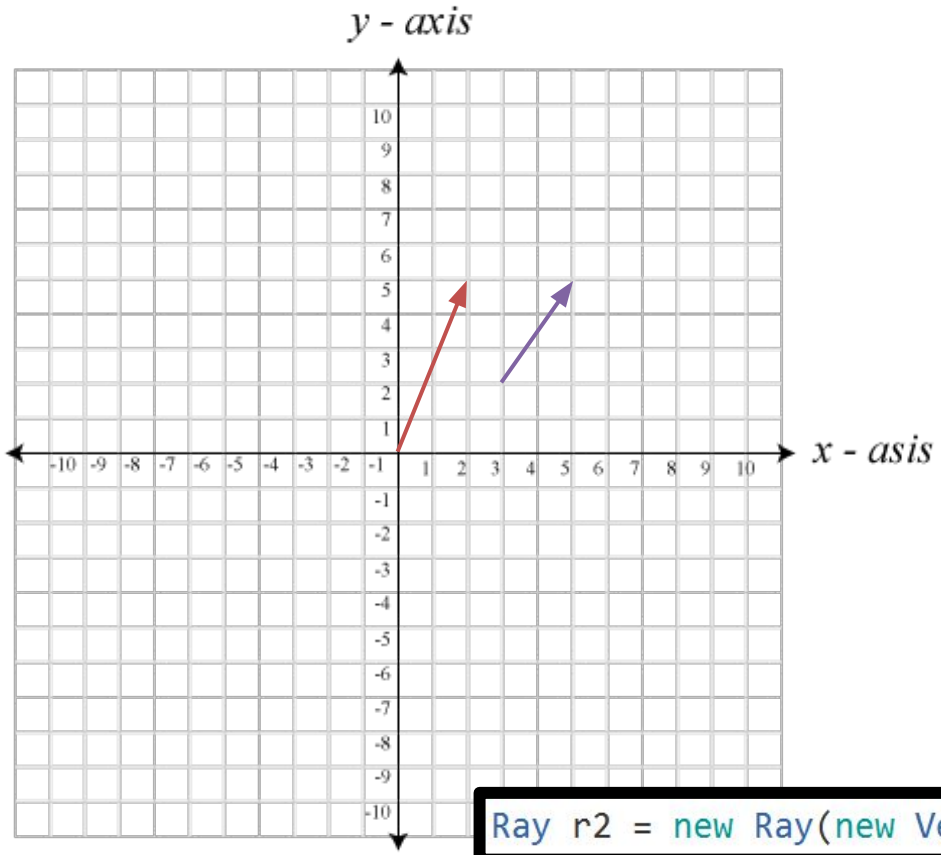
- A ray is a combination of a point of origin, and a direction:

```
Ray ray = new Ray(
```




```
Ray ray = new Ray (new Vector3 (0f, 1f, -1f), new Vector3 (0f, 2f, 1f));
```

Rays




```
Vector2 v2 = new Vector2 (2f, 5f);
```

```
Ray r2 = new Ray(new Vector2(3f, 2f), new Vector2(2f, 3f));
```

Raycasting

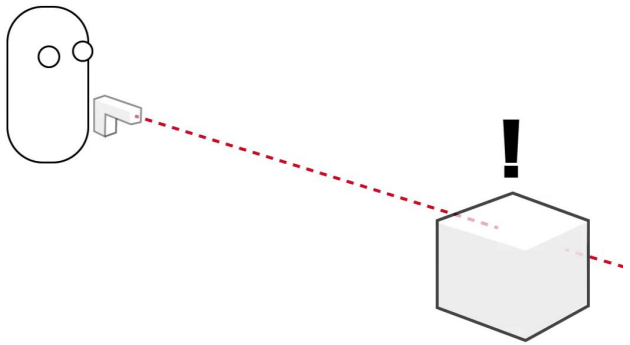
Shooting an invisible ray from a point in a specified direction to check for colliders in its path.

Physics.Raycast(Vector3 origin, Vector3 direction, RaycastHit hitInfo, float distance, int LayerMask);

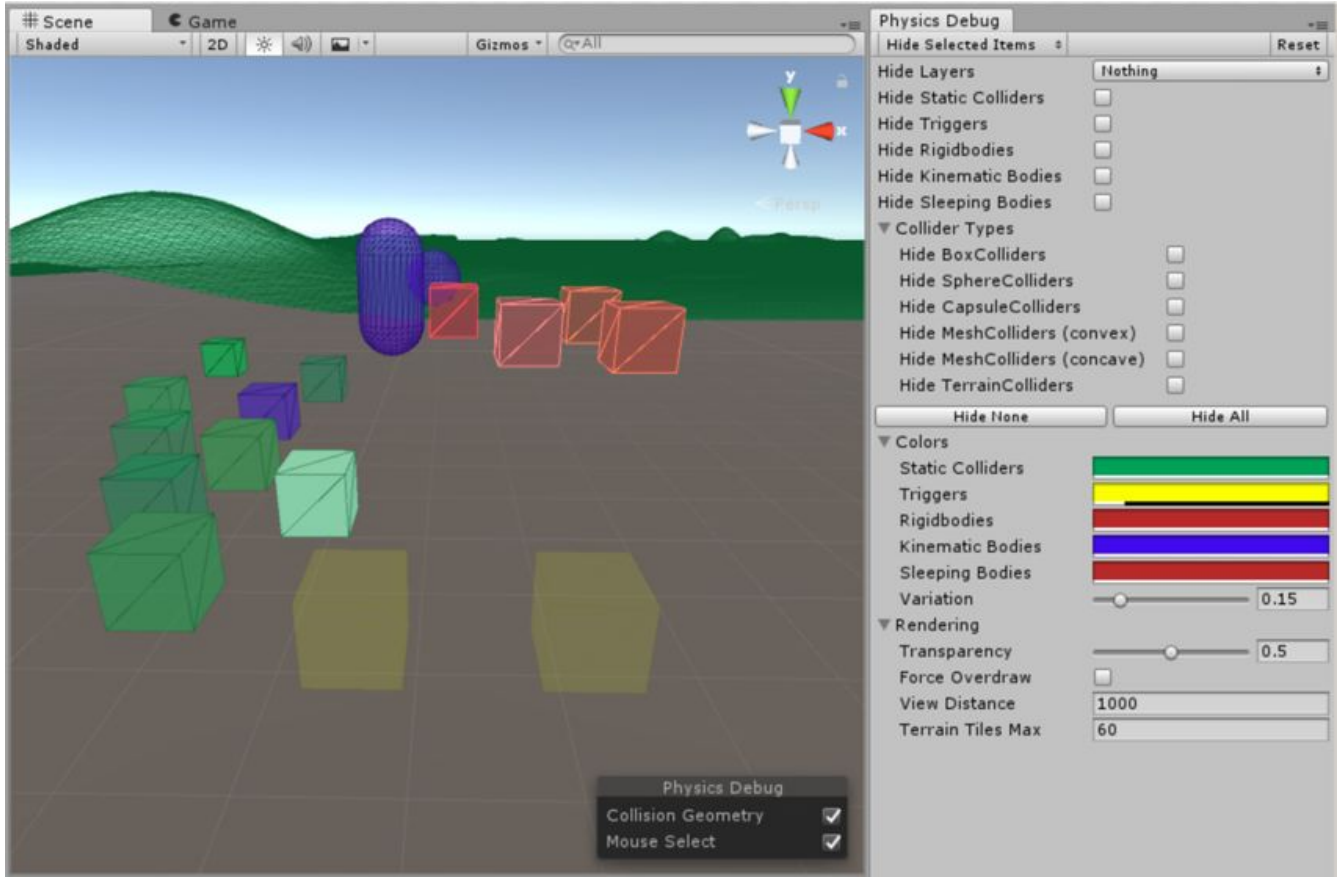
- Use the information from these collisions to create your behaviour
  - hitInfo.collider to get the object the ray hit.
- Use Debug.DrawRay(origin, direction/length) to cast a visual ray.

```
RaycastHit hit;
Ray landingRay = new Ray(transform.position, Vector3.down);

if(!deployed)
{
    if(Physics.Raycast(landingRay, out hit, deploymentHeight))
    {
        if(hit.collider.tag == "environment")
        {
            DeployParachute();
        }
    }
}
```



Physics Debugger



Exercises

Exercises are available on itslearning along with an example unitypackage.