



Introduction to Algorithmic Problem Solving

Game Development

Lesson 3 - Physics and Interactivity

Unity Physics

Unity has an in-depth **physics system** that's usually pretty good.

Common terms:

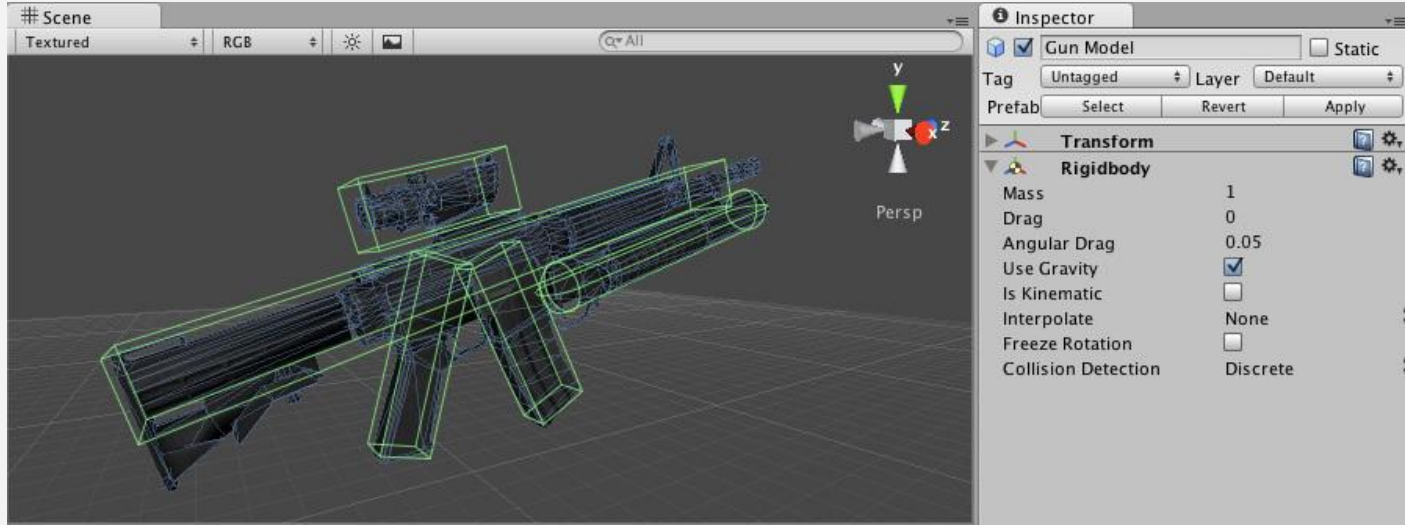
- Colliders
- Rigidbodies
- Tags
- Physics Layers



Collider

Invisible geometry that defines the **area of influence** for a **physical object**.

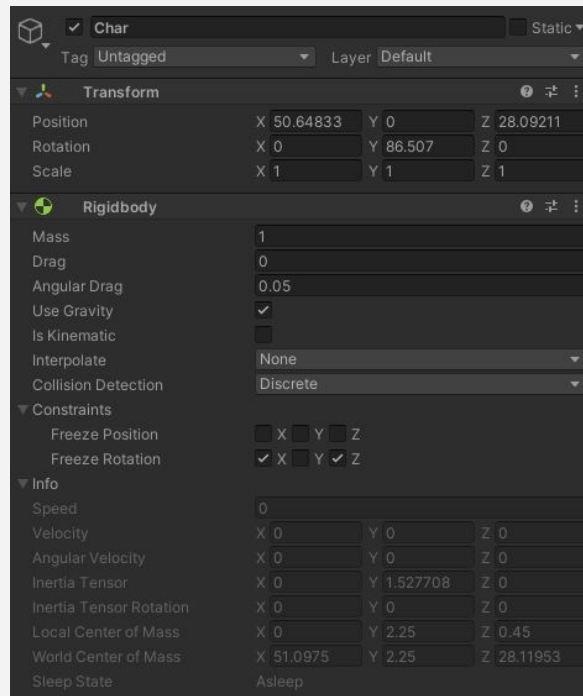
- Can be **solid**, which means it will not move through other colliders.
- Can be a **trigger**, which other colliders can move through.



Rigidbody

A **rigidbody** defines how a **Game Object** is affected by the **physics system**.

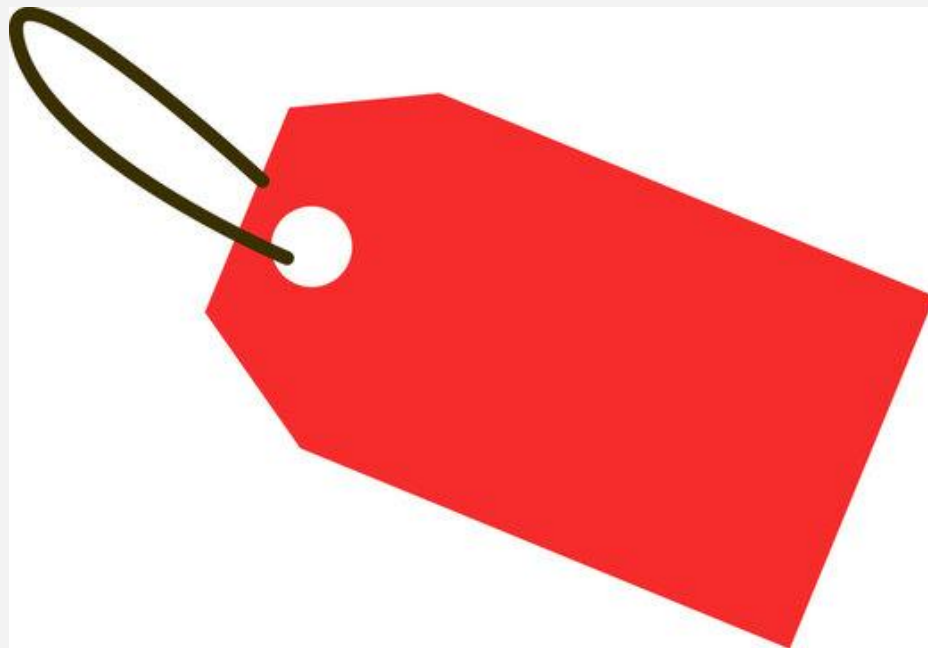
- Can be **dynamic**, which means it is affected by **gravity** and responds to **forces**.
- Can be **kinematic**, which means it still acts upon other objects but has to be **moved in code**.
- Can be **static**, which means it **can't move**.



Tags

A **tag** is a quick way for Unity to know **what an object is**.

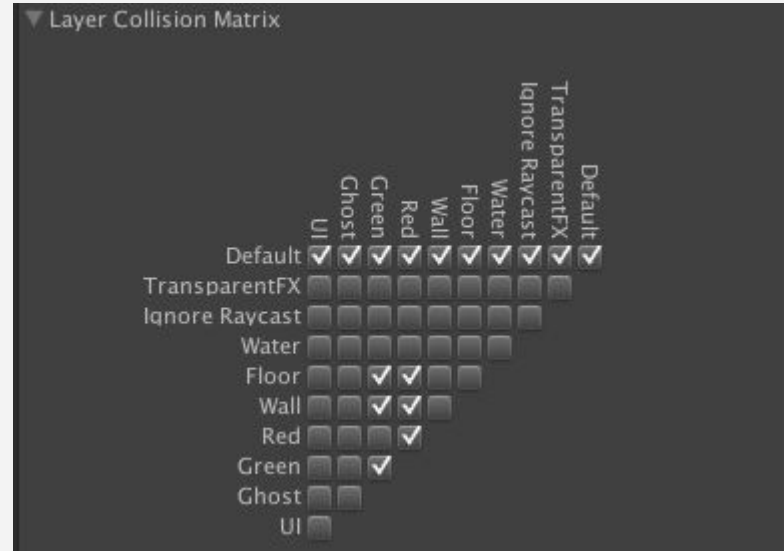
- A **tag** is just a **string** assigned to a Game Object.
- Useful in physics, because we can check them in code to see what kind of object we are dealing with.



Physics Layers

Physics layers define **which objects can collide** with each other.

- You can **enable** or **disable** interactions between each **physics layer**.
- Allows for much **greater control** over your game.



Getting Physical

By the end of today, you will be able to use the Unity physics system to control how objects interact with each other in your games.



Today's Plan:

15 min - Demonstration

15 min - Collaborative problem solving

75 min - Writing code