CADT IDT

Game Development

W4 – Physics Engine

Lecturer : **Dr. VA Hongly**

EXPLAIN

Session Objective

- ✓ Rigid body
- ✓ Collider
- ✓ Joint
- ✓ Particle System
- ✓ Cloth

EXPLAIN

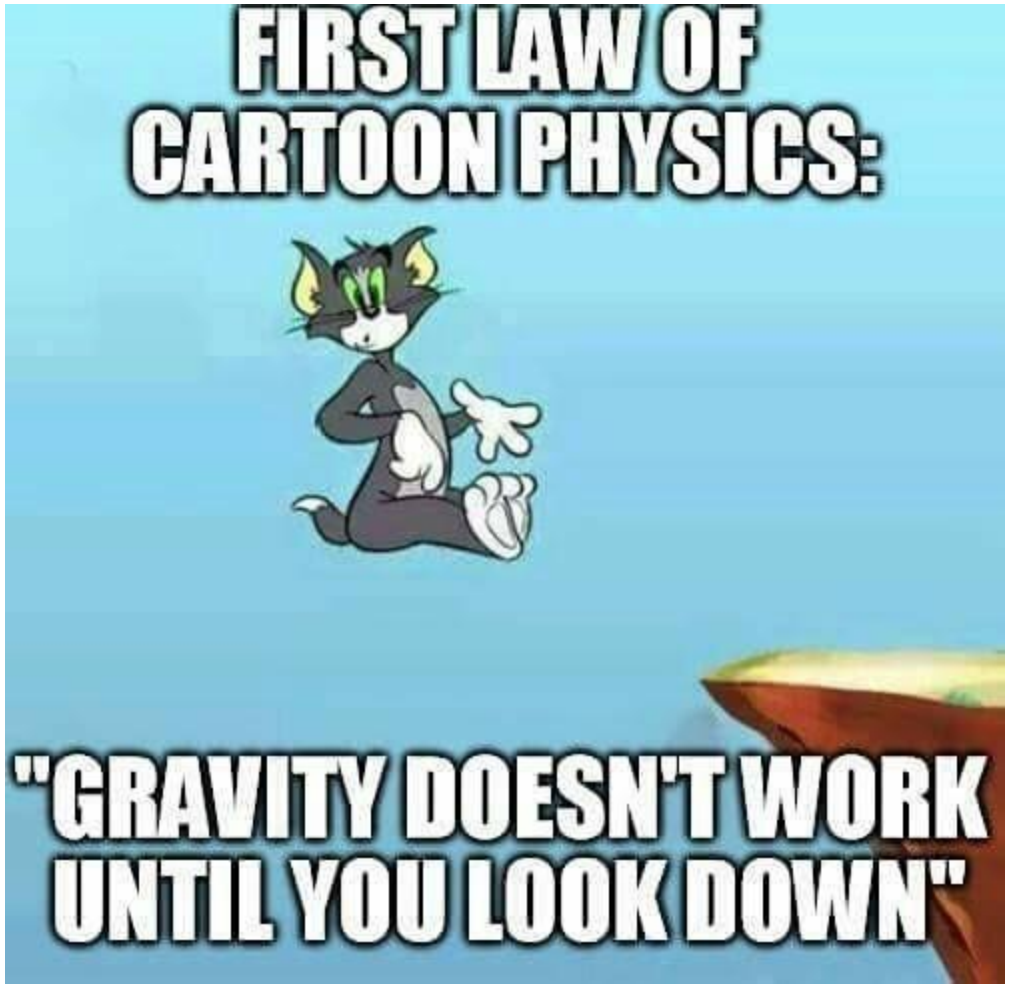
What is Physics?

- Rigid Body ?
- Gravity ?
- Collider ?
- Joint ?

Issac Newton

EXPLAIN

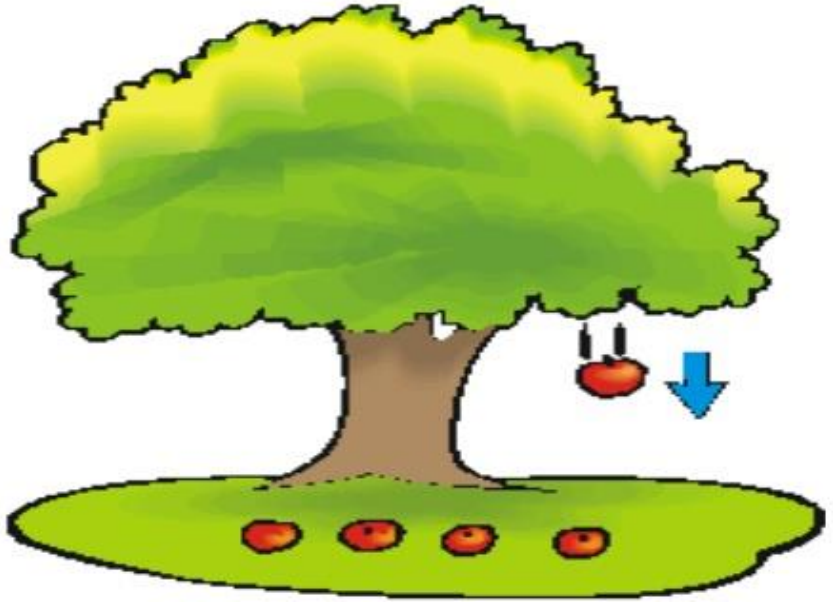
Gravity?



EXPLAIN

Gravity

Gravity- The force of attraction between objects of mass. Earth's gravity pulls objects towards its surface.



EXPLAIN

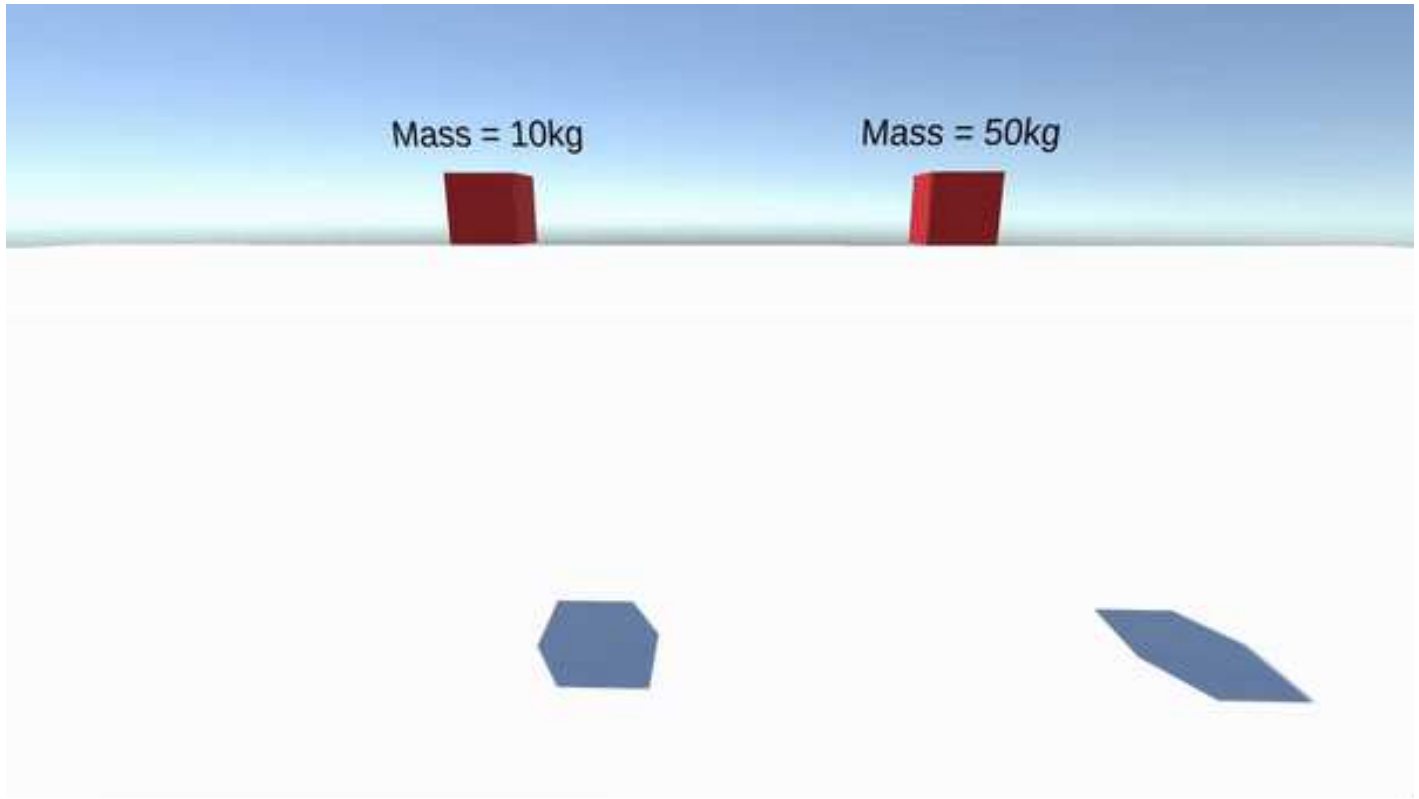
Rigidbody 3D

Type	Description
Mass, Drag, Angular Drag	• **Mass:** define the mass of the GameObject (in kilograms). • **Linear Drag** : Define the decay rate of a Rigidbody's linear velocity, to simulate drag, air resistance, or friction. *Ex: friction against drop.* • **Angular Drag** : decay rate of a Rigidbody's rotational velocity. *Ex: friction against rotate.*
Automatic Center of Mass	• If **Enable:** it predicted center of mass for the Rigidbody, based on its shape and scale.
Automatic Tensor	• If **Enable**: use the physics system's predicted tensor and tensor rotation for the Rigidbody. Ex:
Use Gravity	***Allow gravity force?***
Is Kinematic	***Allow object to move without force or velocity?***
Interpolate	• **None** : no smoothing movement • **Interpolate** : smoothing base on previous frame • **Extrapolate** : smoothing base on estimate position in next frame
Collision Detection	• **Discrete vs Continous**

https://docs.unity3d.com/2022.3/Documentation/Manual/class-Rigidbody.html

EXPLAIN

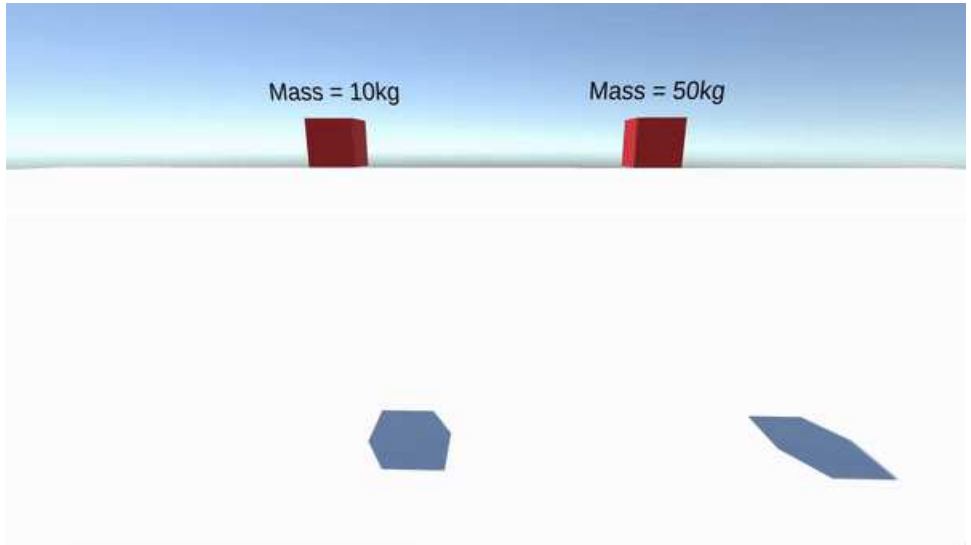
Rigidbody 3D

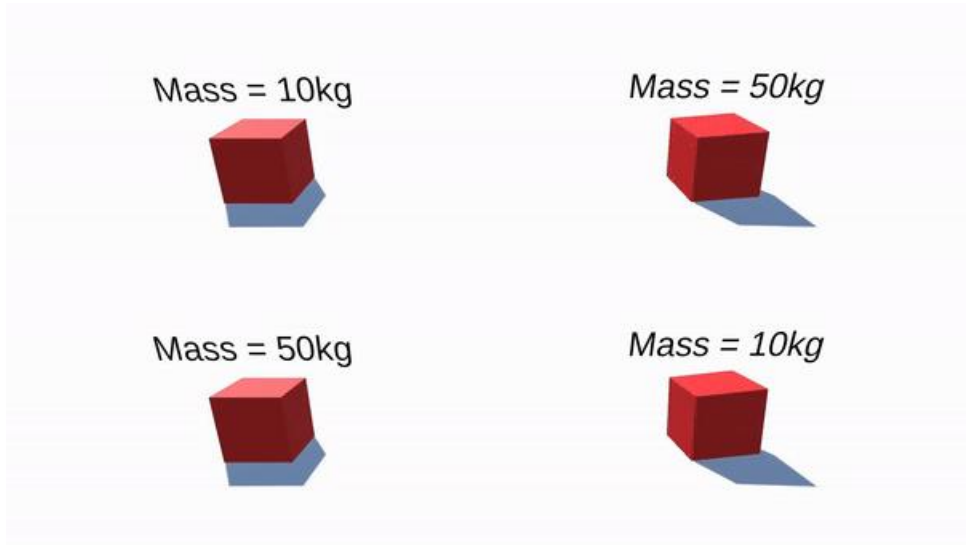


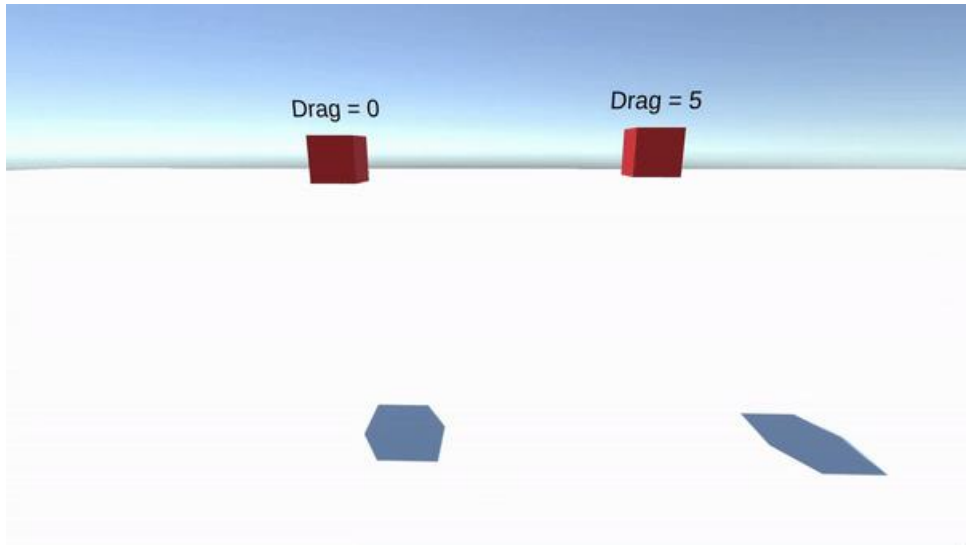
Gravity is independent of **Mass** in free fall

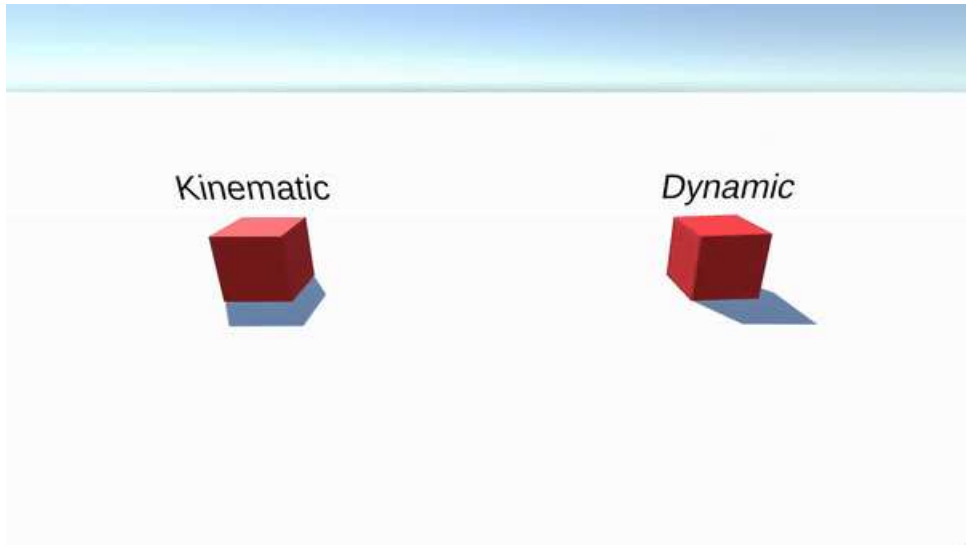
EXPLAIN

Rigidbody 3D



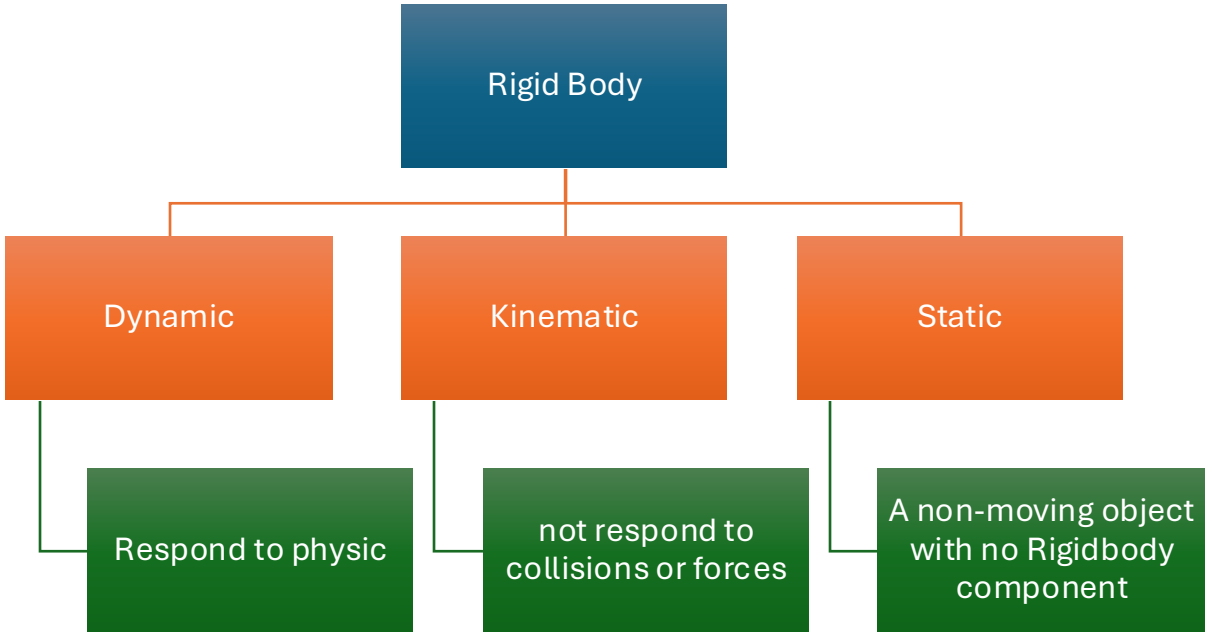






EXPLAIN

Rigid Body

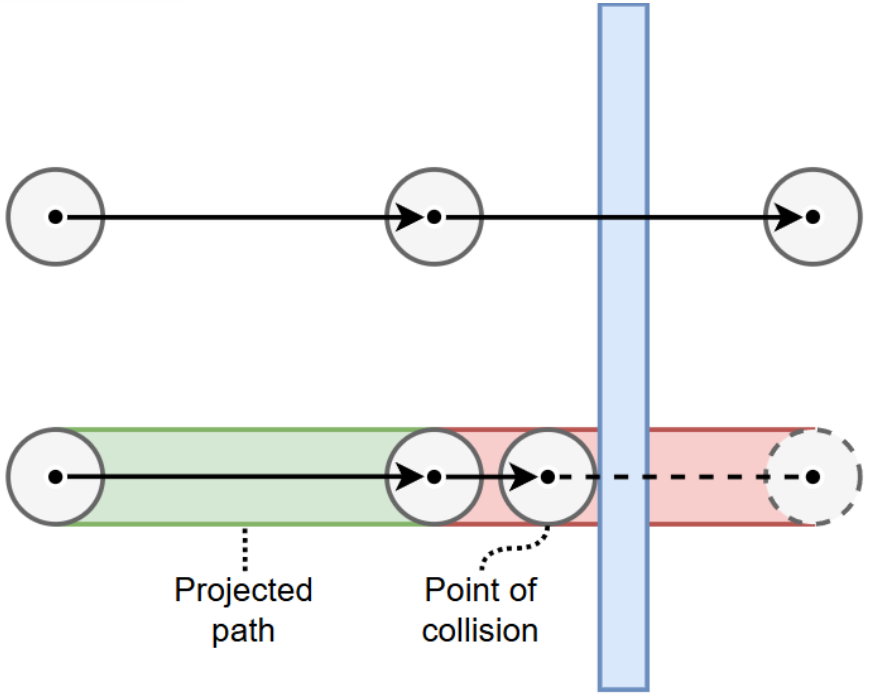


- Objects that need to move freely and interact with forces, like gravity and collisions.
- Objects that need specific movement or animations but shouldn't react to other forces or gravity.
- Immovable objects or background elements that never need to change position.

EXPLAIN

Rigidbody

- Proper use of collision modes:
  - Use **Discrete** if possible
  - If your object has fast angular motion, use **continuous speculative**
  - If ghost collision are a problem, use **continuous**
  - If multiple, fast, dynamic object are collided, use **continuous dynamic**



REVIEW


10 MIN

Rigid Body

Q1 – Give an example for **Dynamic** object use in a game.

Q2 – Give an example for **Kinematic** object use in a game.

Q3 – Give an example for **Static** object use in a game.

EXPLAIN

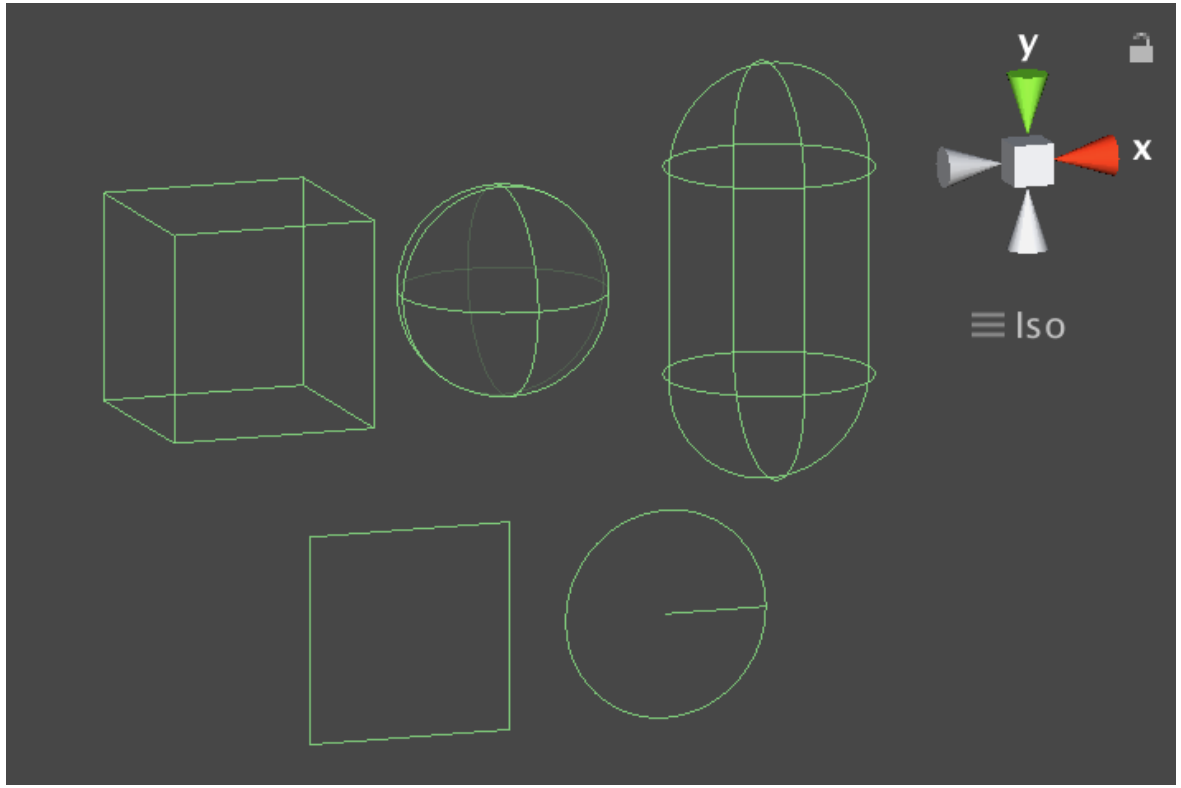
Game vs Real-life

Old Man Once Said: "*If you ever do this, your childhood was awesome!"*

EXPLAIN

Collider

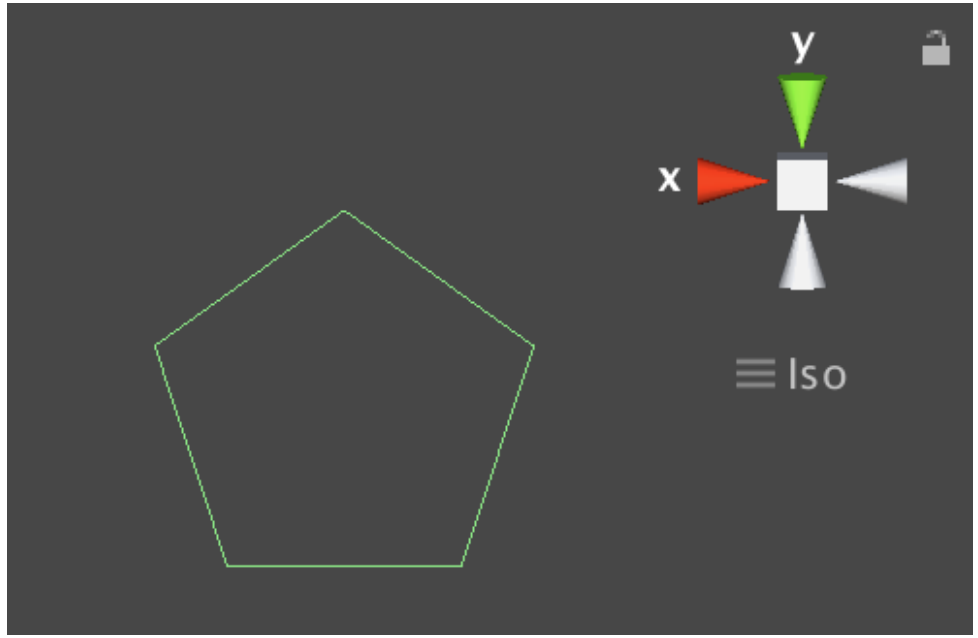
- Like invisible and touchable skin covering object
- Basic collider including:
  - Box Collider / Box Collider2D
  - Sphere Collider / Circle Collider2D
  - Capsule Collider / Capsule Collider2D



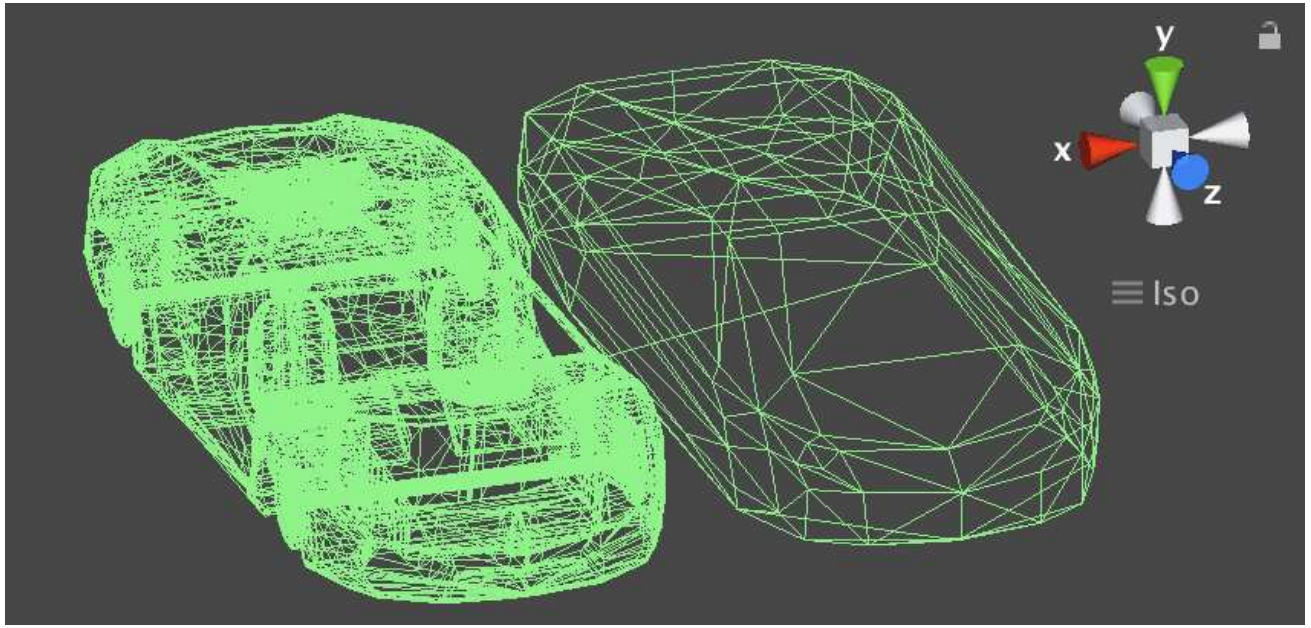
EXPLAIN

Collider (Cont.)

- Advance Collider
  - Mesh Collider
  - Polygon Collider2D



Polygon Collider2D

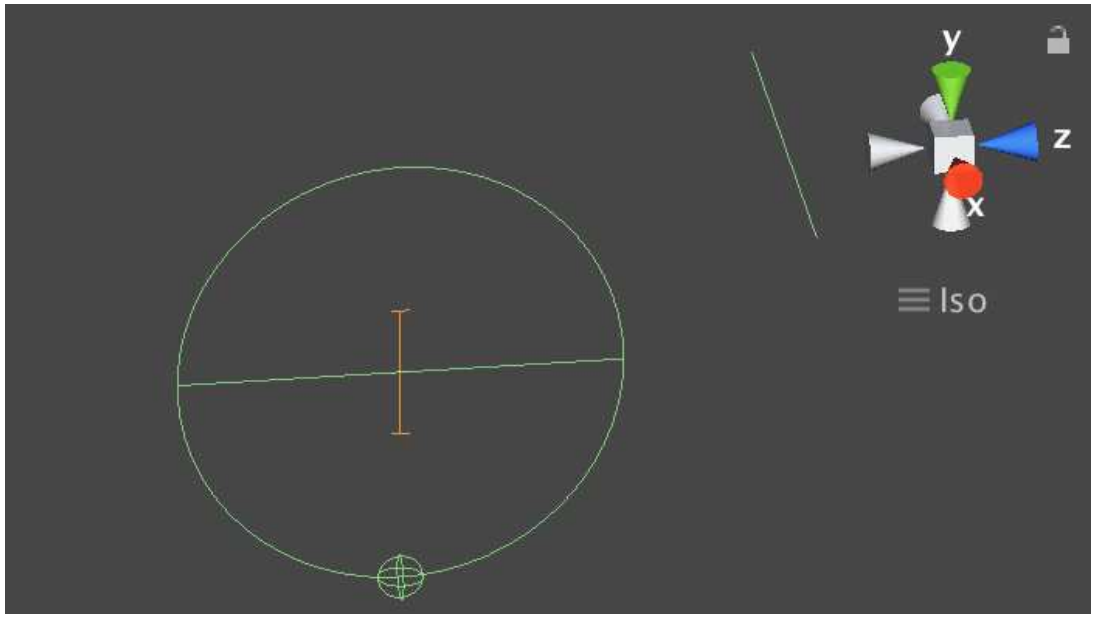


Mesh Collider of a car

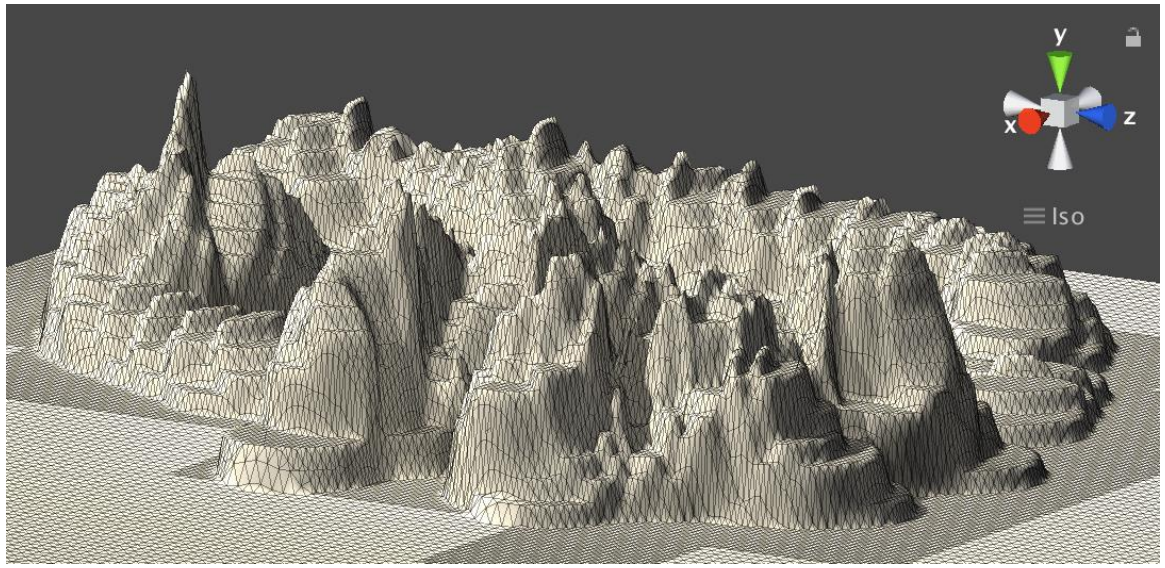
EXPLAIN

Collider (Cont.)

- Others
  - Wheel Collider
  - Terrain Collider
  - Edge Collider2D

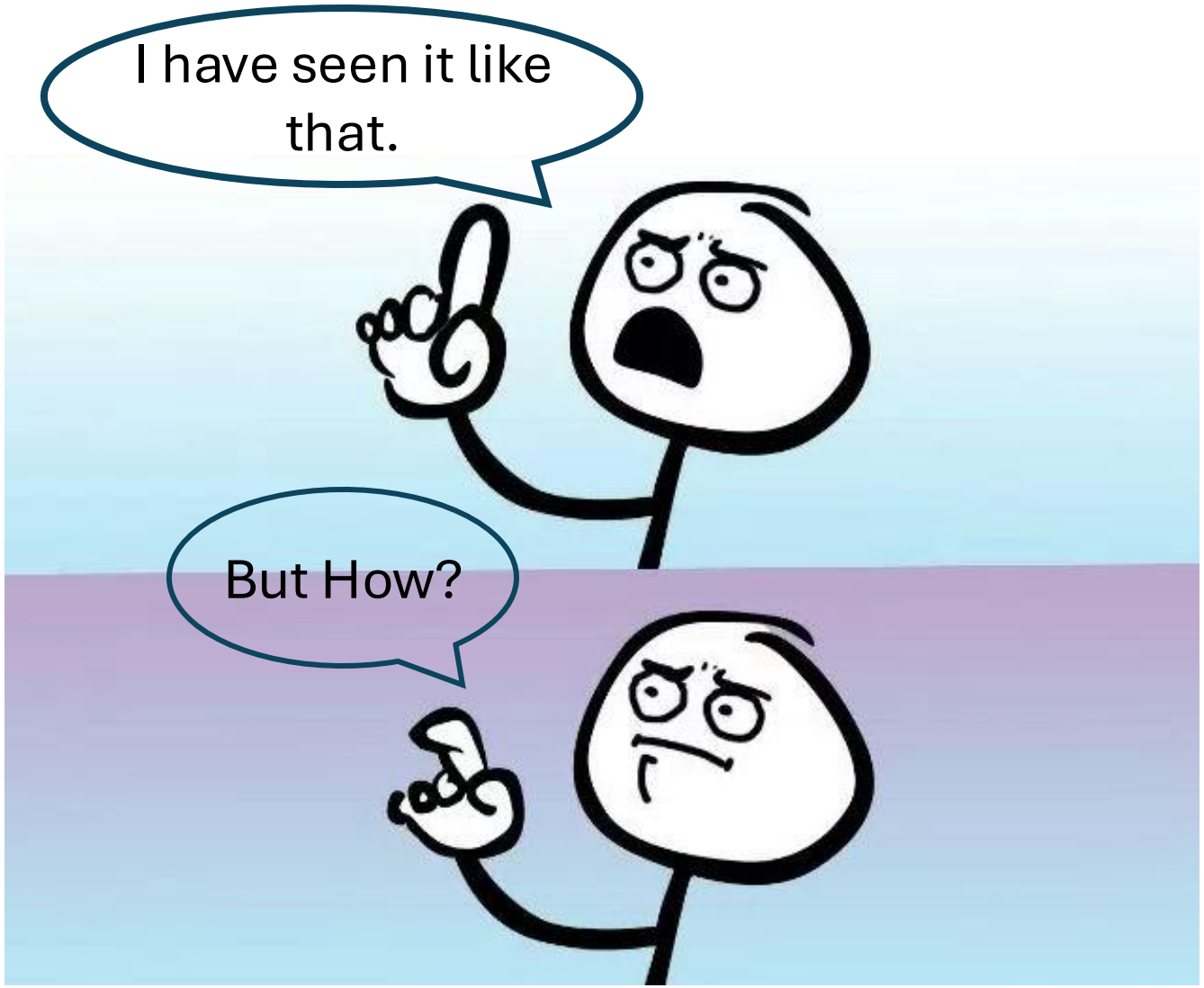


Wheel Collider and Edge Collider 2D(Line)



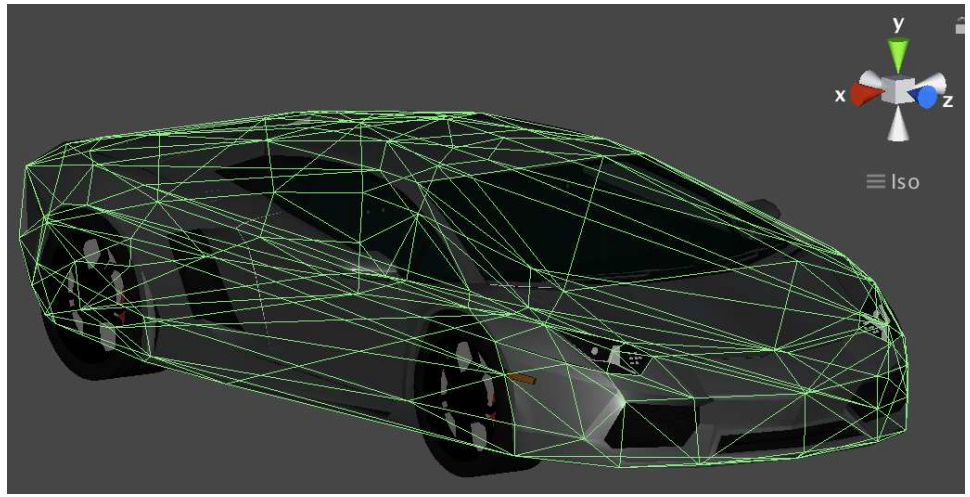
Terrain Collider

Some awkward you might have seen in game

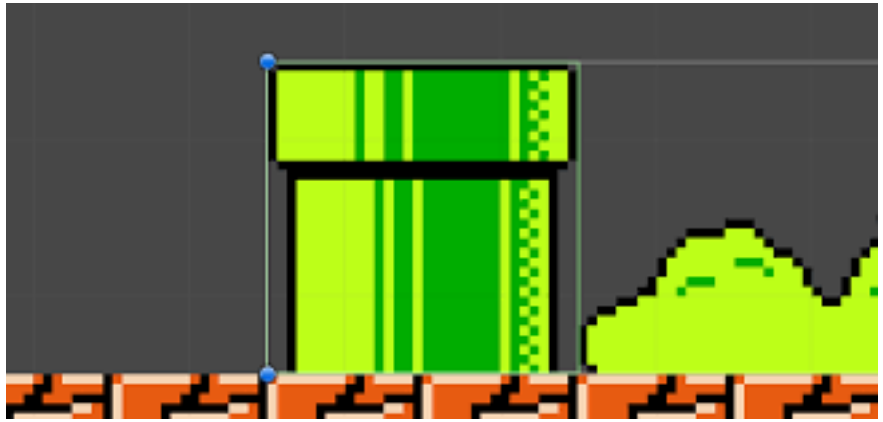


EXPLAIN

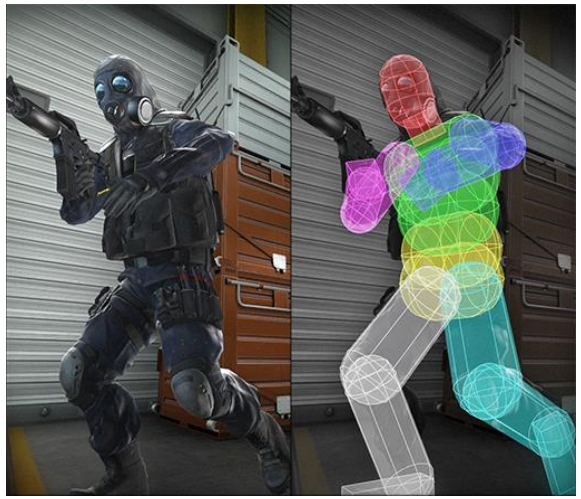
Some Collider Examples



Car with Mesh Collider



An obstacle with Box Collider2D



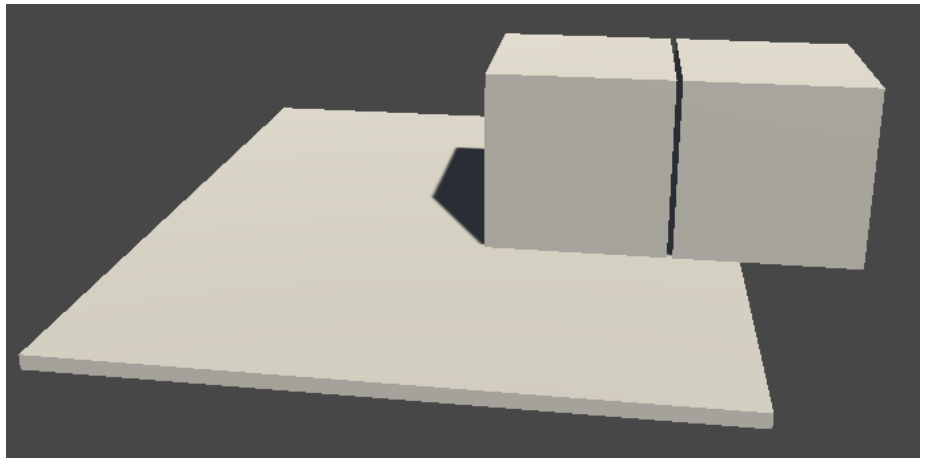
Character with Nested Capsule Collider

Character with Capsule Collider and Gun with Box Collider

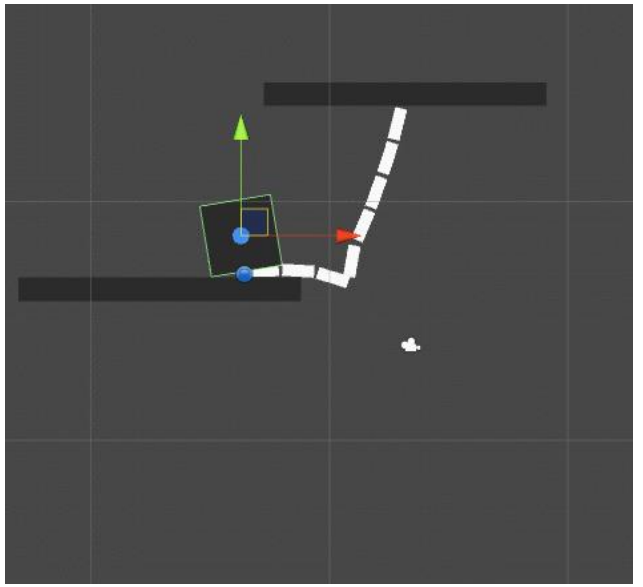
EXPLAIN

Joints

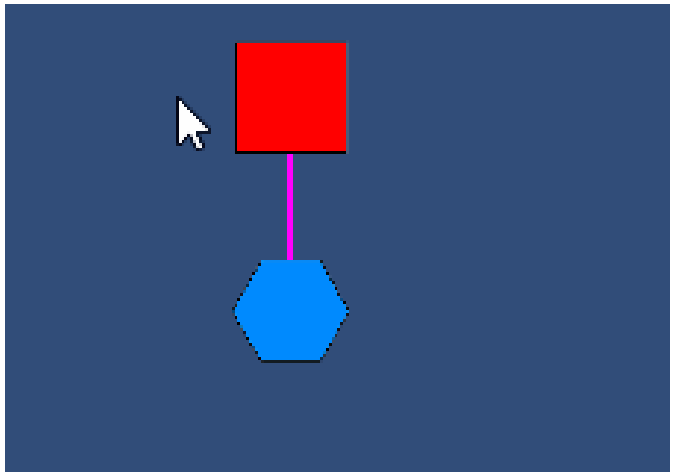
- There are 3 common types of joint in Unity
  - **Fixed Joint**
  - **Hinge Joint**
  - **Spring Joint**



2-Cube Connected via **Fixed Joint**



A series of cube connected via **Hinge Joint**



Pentagon connected to Cube via **Spring Joint**

EXPLAIN

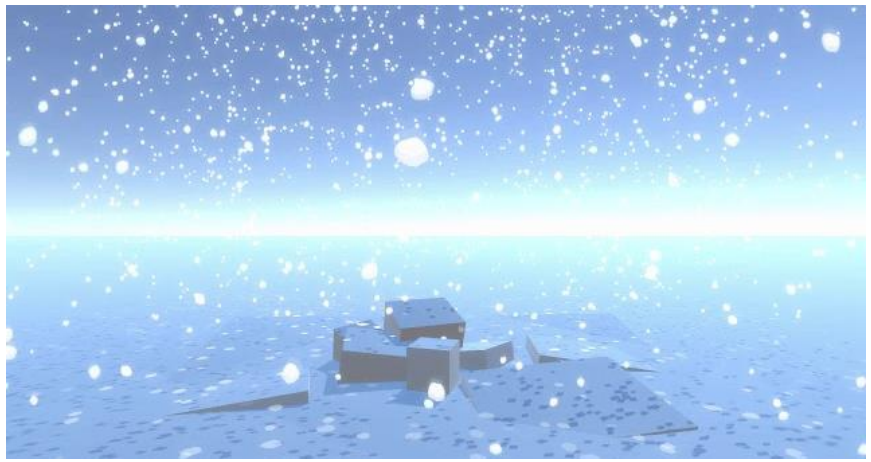
Casting

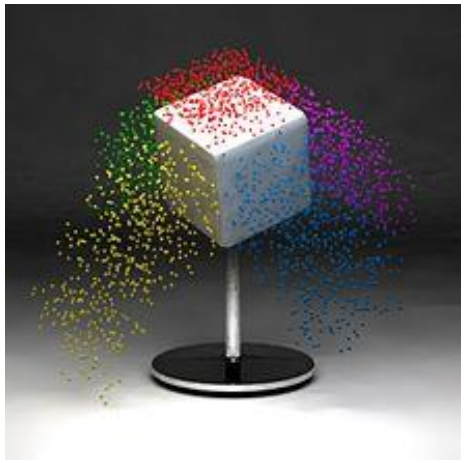
- **Casting** is an act of casting toward 2D or 3D collider to send or receive message to or from the physics object which implement event interfaces.
- Some Unity3D casting such as:
  - **Raycast**
  - **SphereCast**
  - **BoxCast**
  - **CircleCast**
  - **CapsuleCast**
  - **LineCast**

Enemy casting ray seeking for player

EXPLAIN

Particle System

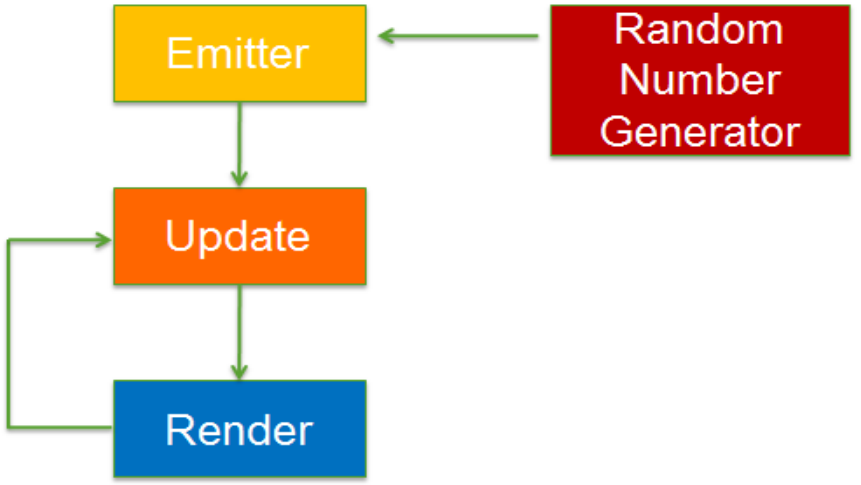




EXPLAIN

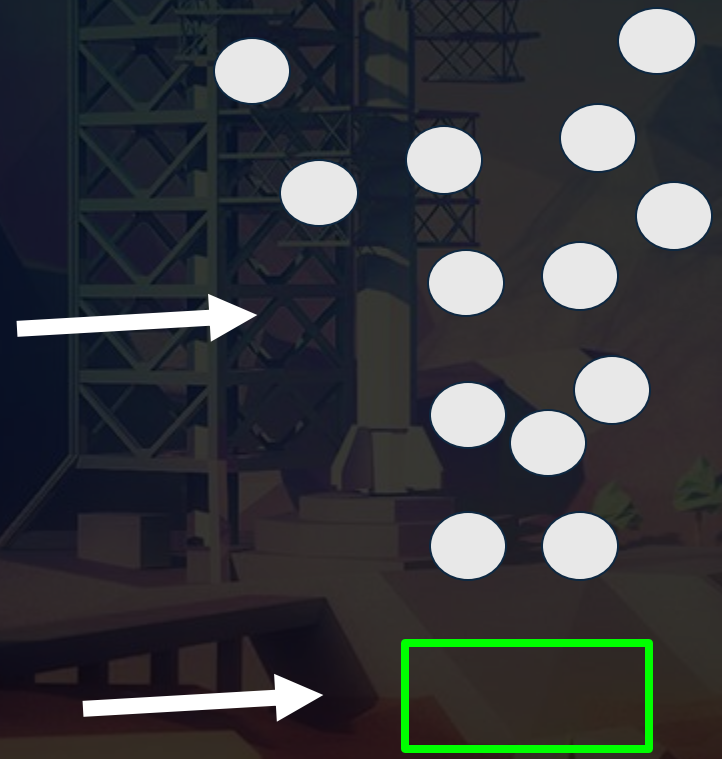
Particle System

- ✓ A system to control collection of a number of individual elements (point, Line, triangle, or texture) which act independently but
  - ✓ Share some common attributes:
  - ✓ Position (3D)
  - ✓ Velocity (vector: speed and direction)
  - ✓ Color + (transparency)
  - ✓ Lifetime
  - ✓ Size, Shape



Process of particle system

Particles System Component



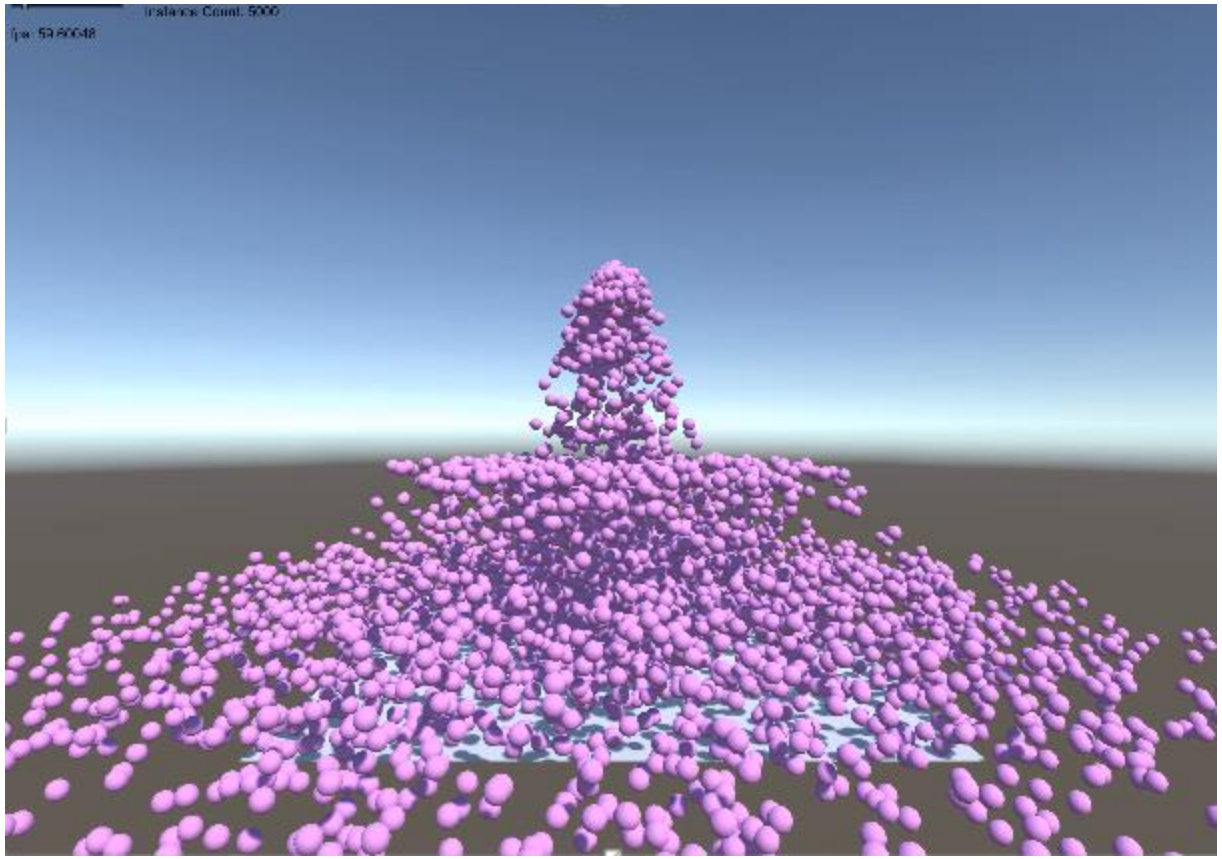
Particle System is a Component added to a Game Object

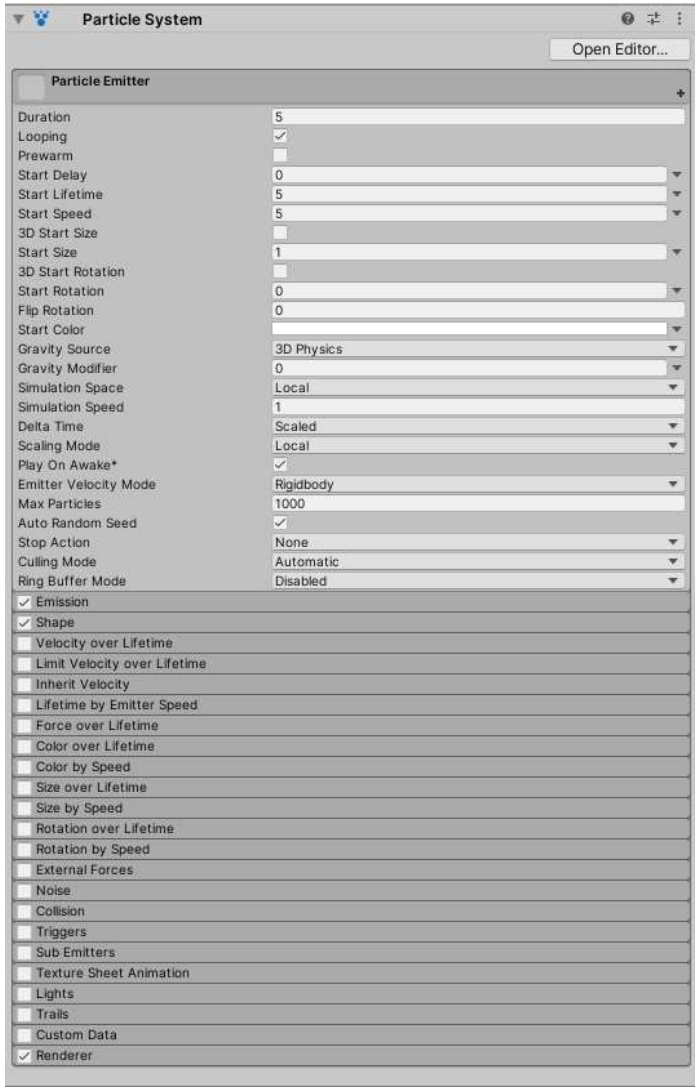
We use Modules for controlling behaviour

Each particle is not a Game Object

EXPLAIN

Particle System

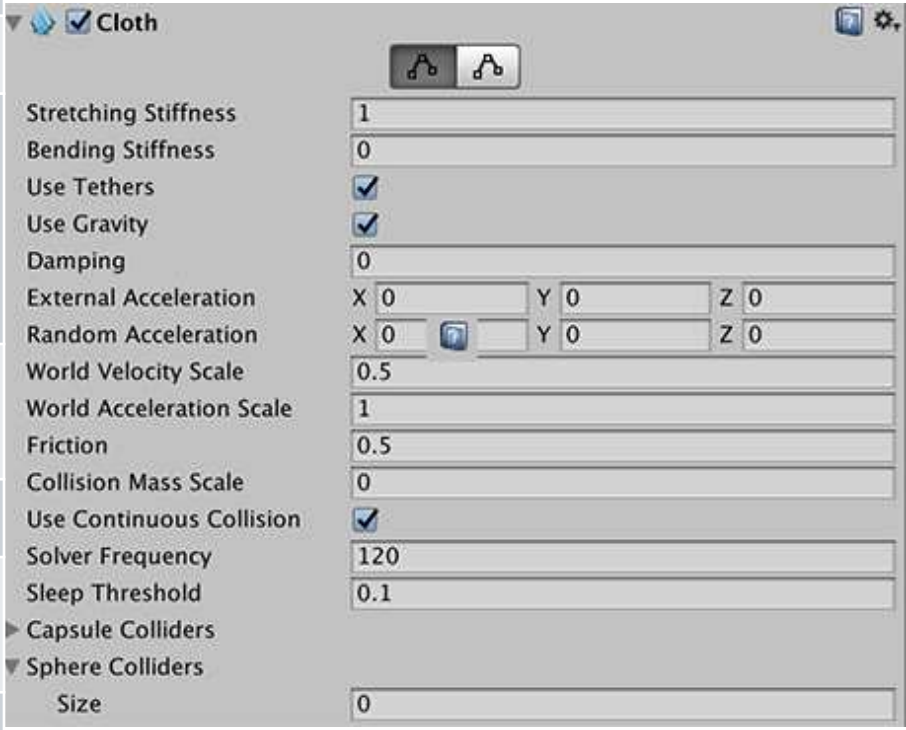




EXPLAIN

Cloth

Body Type	Explanation
Stretching Stiffness	Stretching stiffness of the cloth.
Bending Stiffness	Bending stiffness of the cloth.
Use Tethers	Apply constraints that help to prevent the moving cloth particles from going too far away from the fixed ones. This helps to reduce excess stretchiness.
Use Gravity	Should gravitational acceleration be applied to the cloth?
Damping	Motion damping coefficient.
External Acceleration	A constant, external acceleration applied to the cloth.
Random Acceleration	A random, external acceleration applied to the cloth.



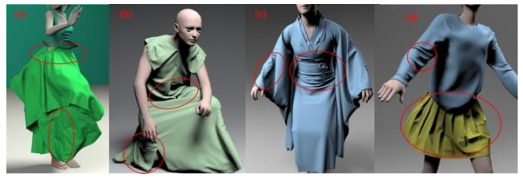
https://docs.unity3d.com/2022.3/Documentation/Manual/class-Cloth.html

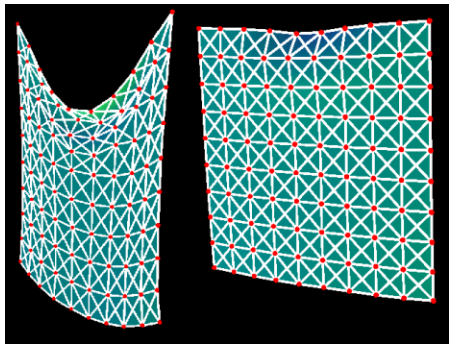
EXPLAIN

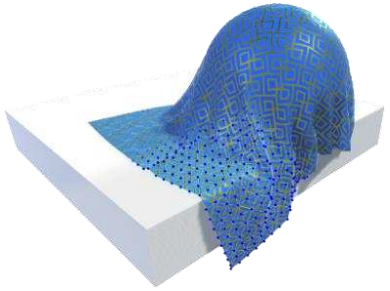
Cloth Simulation

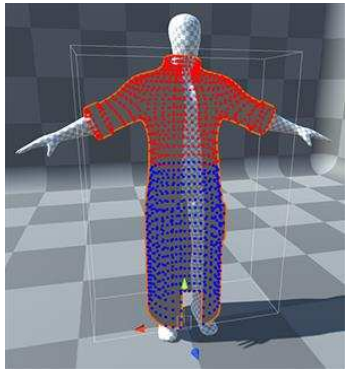
Physical Method:

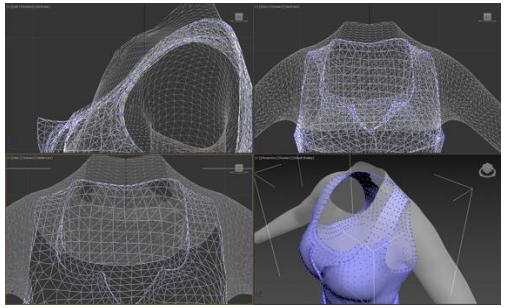
Treat the cloth model as a grid of **nodes** connected to each other by **springs**.











Obi Cloth

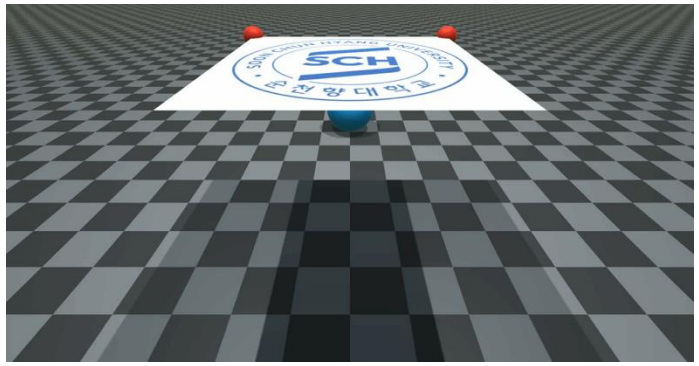
Cloth simulation improvement

EXPLAIN

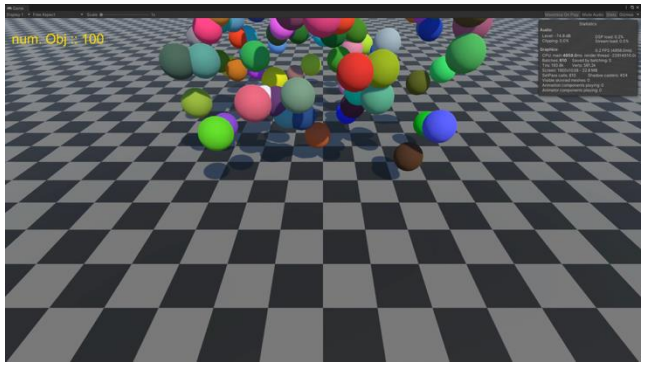
Cloth Simulation

Number of vertex	Number of Constraint	Mesh-based PBD	Unity3D's Cloth	Shader-based PBD
1,024	5,766	88.57 fps	146.42 fps	160.75 fps
4,096	23,814	54.45 fps	141.27 fps	151.57 fps
16,384	96,774	20.19 fps	105.34 fps	146.20 fps
65,536	390,150	5.34 fps	46.57 fps	130.07 fps

(Accelerate by NVIDIA GeForce RTX 2070 SUPER 8GB V-RAM)



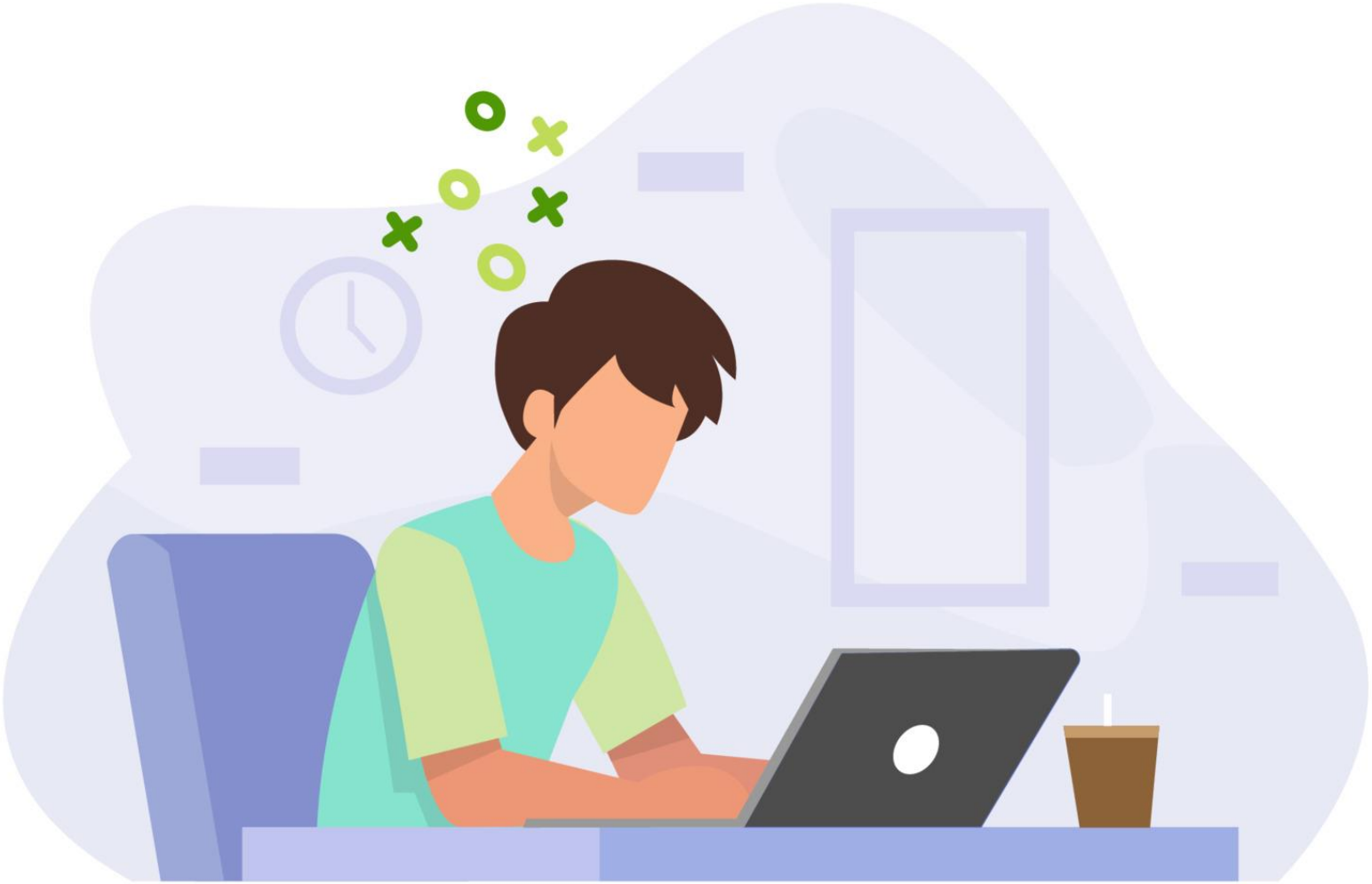
16,384 vertices mesh



100 object of soft body

DEMO

30 MIN



PRACTICE


1 hour

Your Turn

- Experiment with the settings (*to see how it affect*)
  - Key component:
    - Rigid Body
    - Collider
    - Joint
    - Particle System
    - Cloth (optional)
  - *Note: You may explore additional ideas for applying physics*

HOMEWORK

13 days

Homework:
Physics-Based Object Interaction

*Deadline: 08 November 2025 (**11:59 PM**)*

✓ Objective:

- Create a physics-based interactive object (2D or 3D) that demonstrates the use of **Rigidbody** and **Unity's physics engine features**. The object must:
  - Use **Rigidbody** to apply forces and simulate physics.
  - Respond to **user input** to apply movement using **AddForce()** or **AddTorque()**.
  - Detect collisions using **OnCollisionEnter()** or **OnTriggerEnter()**.
  - Include **customization of physics properties** (e.g., mass, drag) through the Inspector.

✓ Instructions:

- **Think of a Simple Game Concept:** Design a small game idea that involves physics such as:
  - A ball rolling through a maze.
  - A pinball game.
  - A target shooting game where objects fall or bounce.
- **Setup your scene:** game concept and game design
- **Create game mechanics (using script):** player movement, detect collision etc,.

Submission:

➢ Must submit the following files:

1. **Slide for presentation (pdf)**
2. **A demo video**
3. **Unity Package or GitHub Url**