LAPLAND UNIVERSITY OF APPLIED SCIENCES

Game Physics

In Unity

A1301794 – Mert Ertugrul MUSUL

TABLE OF CONTENTS

GAME PHYSICS : UNITY

There is nothing real or natural about how things behave in computer games. They don't know anything about the natural physical phenomenon that exists in reality. If you place a bunch of objects in a scene in Unity, these objects will not naturally know what they are supposed to do. They won't even be affected by gravity unless you explicitly tell them what gravity is. Game engines (like Unity) require Physics to be simulated using a **Physics Engine**. The Physics Engine tries to simulate physics in your game so that the objects in your game demonstrate natural physical phenomenon.

Unity contains powerful 3D physics engine NVIDIA® PhysX® Physics. Create immersive and visceral scenes with clothes and hair that blow in the wind; tires that screech and burn; walls that crumble; glass that shatters, and weapons that inflict major damage.

Unity's easy-to-use integrated 2D physics system uses the same system of rigidbodies, joints and colliders as our feature-rich 3D solution. Actually, however, it's driven by Box 2D, for an industry-hardened stable and versatile 2D solution.

It's easy to mix 2D and 3D physics in Unity. By doing so you can work in completely new ways and create innovative game formats with ease. Want to render a 3D mesh in 2D space? No problem, just add a 2D Rigidbody Component.

In this "Game Phyics" project, models are made and scripts are written by me. Textures are license free to use.

And I'm listing sources that I get help while preparing my report.

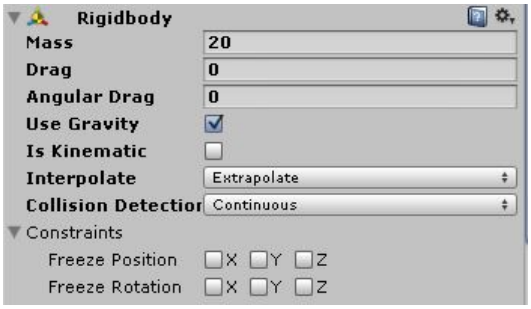
Sources :

http://docs.unity3d.com/

http://3dgep.com/physics-in-unity-3-5/#Introduction

http://unity3d.com/unity/quality/physics

CHAPTER 1: GRAVITY



The Rigidbody component is required for objects to act under the influence of Physics. You can apply forces and torques to the Rigidbody component from scripts or you can directly manipulate the linear and angular velocity of the Rigidbody if you want more control over it's behavior.

A Rigidbody component requires a Collider component to be present on the GameObject for correct Physics simulation.
A Rigidbody can either be Physics controlled or Kinematic controlled.

In Unity, The gravity applied to all rigid bodies in the scene. So we add Rigidbody Component to the ball. And give it a mass.

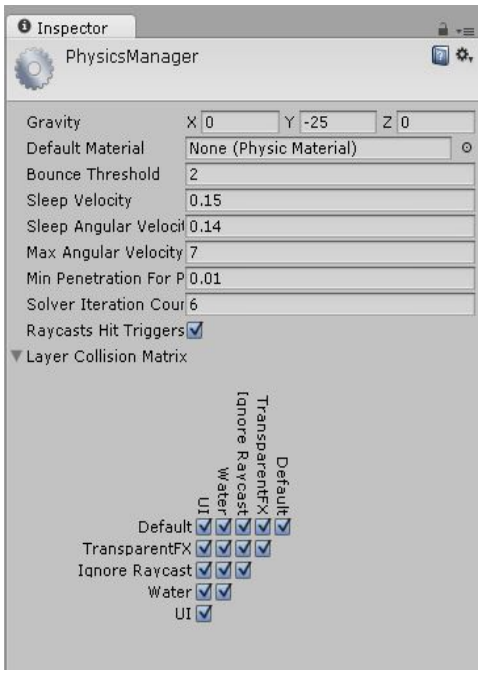
Colliders are used to define the collision shape of objects in your scene. Unity provides three primitive collider types: Box, Sphere and Capsule. Unity also provides a Mesh Collider component that can be used to represent a simplified version of your complex meshes. For vehicles, Unity also provides a Wheel Collider component and a Terrain Collider component that can be used on terrains.

Any Collider type (Box, Sphere, Capsule, Mesh, and Terrain) can be used as a Trigger by setting the Is Trigger property to true. If the Collider is a Trigger we call it a Trigger Collider. Trigger Colliders will not participate in physics simulations but they will fire the OnTriggerEnter, OnTriggerStay, and OnTriggerExit events on the GameObject they are attached to as long as the collider that intersected with the trigger is a Rigidbody Collider.

If you place a Rigidbody on a Trigger Collider, it will be effected by gravity (if theUse Gravity property is true on the Rigidbody component) but it will not collide with other colliders in your scene. If you have Rigidbody objects in your scene that are falling through the floor, make sure the collider component does not have the Is Trigger property set to true. Trigger Colliders can be used for many different things. You can place a Trigger Collider near a doorway that opens the door when the player enters the collider and closes the door when the user exits the Trigger Collider.

The Sphere Collider is similar to a Box Collider except it resembles the shape of a sphere instead of a cube.

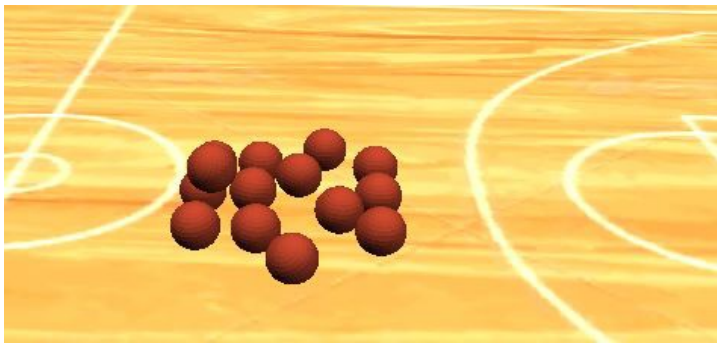
The Sphere Collider has the following properties:
•Is Trigger: If true, this collider will be considered a Trigger Collider and will no longer be influenced by physics but it will fire Trigger events on otherRigidbody colliders.
•Material: The Physics Material that determines the friction and bounce of this object. The Physics Material is irrelevant if this is a Trigger Collider.
•Center: The X, Y, Z, position of the Collider relative to the GameObject's position.
•Radius: The radius of the collider relative to the GameObjects's scale.



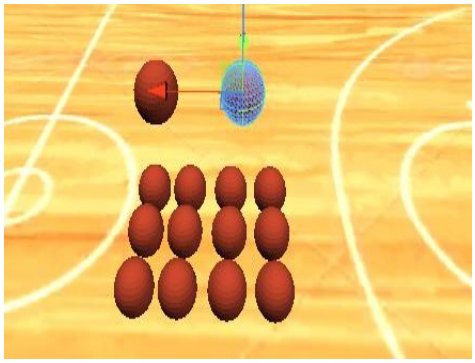
Property:	Function:
Gravity	The amount of gravity applied to all **Rigidbodies**. Usually gravity acts only on the Y-axis (negative is down). Gravity is world units per second squared.
Default Material	The default **Physics Material** that will be used if none has been assigned to an individual **Collider**.
Bounce Threshold	Two colliding objects with a relative velocity below this value will not bounce. This value also reduces jitter so it is not recommended to set it to a very low value.
Sleep Velocity	The default linear velocity, below which objects start going to sleep.

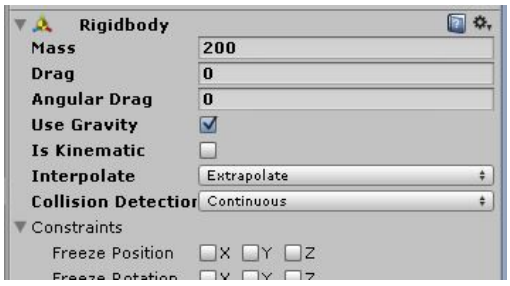
Property:	Function:
Sleep Angular Velocity	The default angular velocity, below which objects start going to sleep.
Max Angular Velocity	The default maximimum angular velocity permitted for any Rigidbodies. The angular velocity of Rigidbodies is clamped to stay within **Max Angular Velocity** to avoid numerical instability with quickly rotating bodies. Because this may prevent intentional fast rotations on objects such as wheels, you can override this value for any Rigidbody by scripting **Rigidbody.maxAngularVelocity**.
Min Penetration For Penalty	How deep in meters are two objects allowed to penetrate before the collision solver pushes them apart. A higher value will make objects penetrate more but reduces jitter.
Solver Iteration Count	Determines how accurately joints and contacts are resolved. Usually a value of 7 works very well for almost all situations.
Raycasts Hit Triggers	If enabled, any Raycast that intersects with a Collider marked as a Trigger will return a hit. If disabled, these intersections will not return a hit.
Layer Collision Matrix	Defines how the layer-based collision detection system will behave.

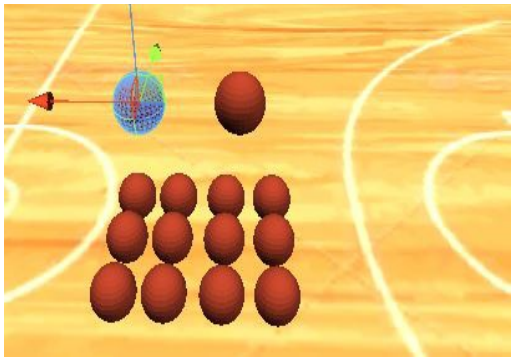
CHAPTER 2: MASS

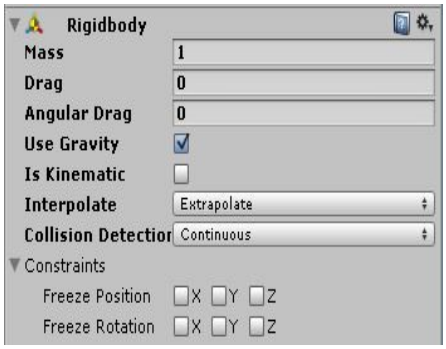


When we set different mass between two balls, their impacts are different as well.







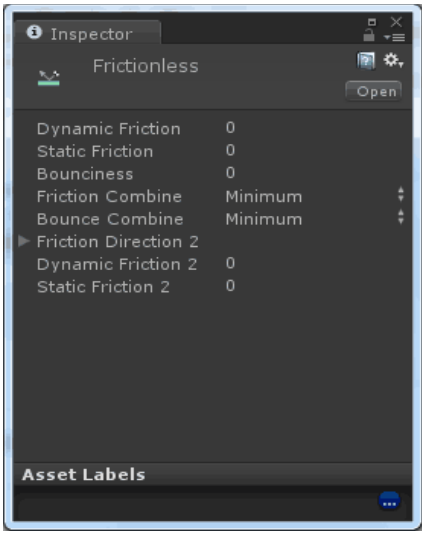


Mass: The mass of the Rigidbody. The mass of a Rigidbody should be based on the relative size and density of the object it is attached to. For example, a large feather will have a much lower mass than relatively small rock or stone. The mass has a direct influence on how much force is required to move the Rigidbody. A very large mass will require more force to move it than a similar sized object with less mass. If you remember the formula $F = ma$ from your basic physics lessons, then you will know that in order to get am object that weighs 10 kg to a speed of 5 meters per second (m/s) in just one second, you must apply a force of 50 newtons ($N = kg\frac{m}{s^2}$) assuming we neglect all other external forces such as friction, drag, and gravity. An object that has a mass of 5 kg only requires 25 newtons to accelerate it to a speed of 5 meters per second in one second.

CHAPTER 3: FRICTION

Friction is not effected by the mass. So, the three boxes slide the same distance.

We define the friction with a Physic Material component,



The Physic Material has the following properties:

- Dynamic Friction: The Dynamic Friction property determines the friction that is applied while the object is in motion. This property usually has a value in the range 0 to 1. A value of 0 will cause no friction to be applied to the surface and a value of 1 will cause the object to slow down very quickly when it is in contact with another surface.
- Static Friction: The Static Friction property determines the friction that is applied while the object is not moving. A value of 0 will allow the object to start moving very easily. A value of 1 will take a lot of force to get the object to start moving.
- Bounciness: Determines how much the object will bounce when it hits another collider. With a value of 0, the object will not bounce at all (all kinetic energy will be absorbed). With a value of 1, the object will bounce indefinitely (all kinetic energy will be maintained). With a value greater than 1 will cause the object to bounce higher (it will gain kinetic energy).
- Friction Combine: Determines how the friction of the two colliding objects is combined:

- Average: The applied friction is the average friction between the two colliders.

- Multiply: The applied friction is the product of the friction of both colliders.

- Minimum: The applied friction is the minimum friction of either collider.

- Maximum: The applied friction is the maximum friction of either collider.

- Bounce Combine: Determines how the bounciness of the two colliding objects is combined:

- Average: The applied bounciness is the average bounciness between the two colliders.

- Multiply: The applied bounciness is the product of the bounciness of both colliders.

- Minimum: The applied bounciness is the minimum bounciness of either collider.

- Maximum: The applied bounciness is the maximum bounciness of either collider.

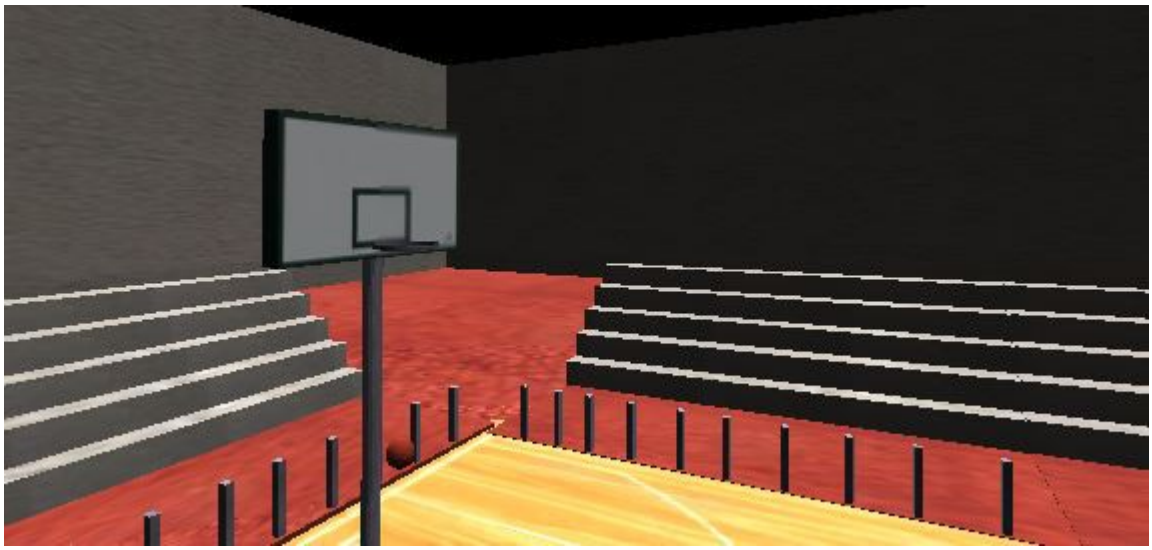
- Friction Direction 2: The Friction Direction 2 is a vector in object local space that can be used to simulate surfaces that can slide easier in one direction than in the other. For example, a corrugated surface that can slide easily in the direction of the corrugated grooves, but tend to resist motion opposite to the corrugated groves.

•Dynamic Friction 2: The amount of dynamic friction to apply in the direction of Friction Direction 2 while the object is in motion.

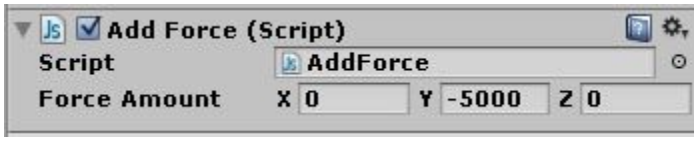
•Static Friction 2: The ammount of static friction to apply in the direction ofFriction Direction 2 while the object is at rest.

Unity provides a few default Physic Materials in the standard packages for Ice, Metal, Rubber, and Wood.

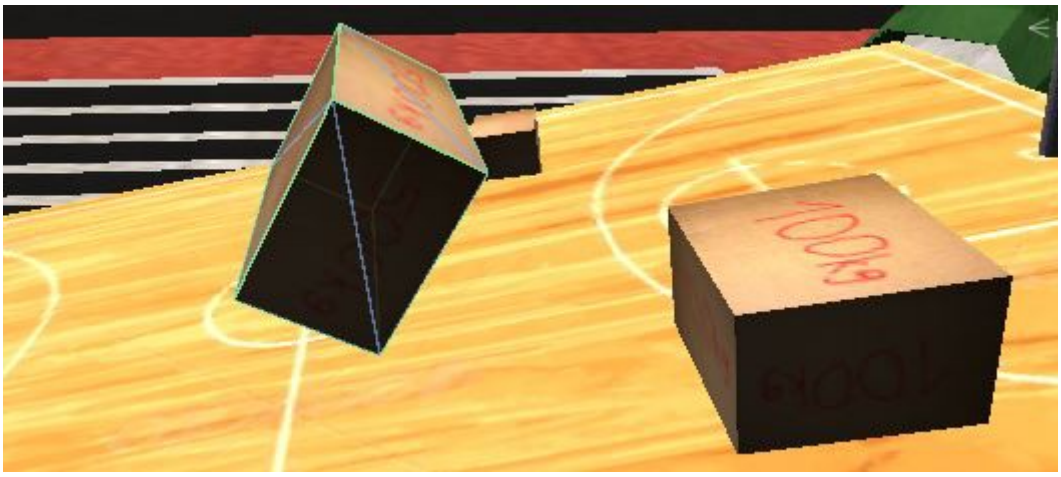
CHAPTER 4: FORCE

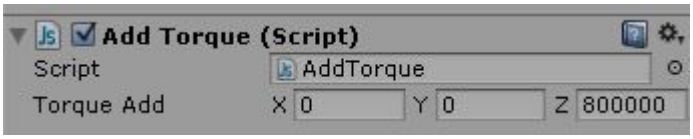


You can apply forces and torques to the Rigidbody component from scripts or you can directly manipulate the linear and angular velocity of the Rigidbody if you want more control over it's behavior.



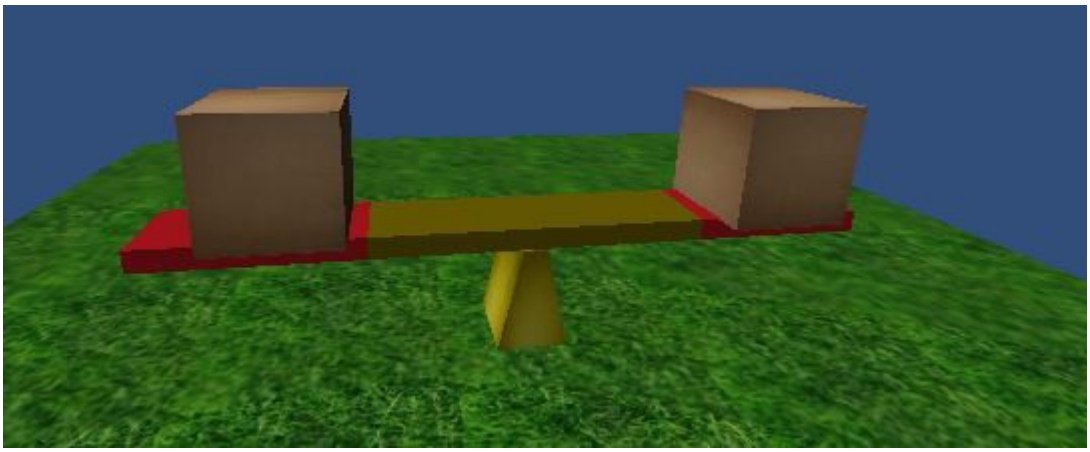
CHAPTER 5: TORQUE

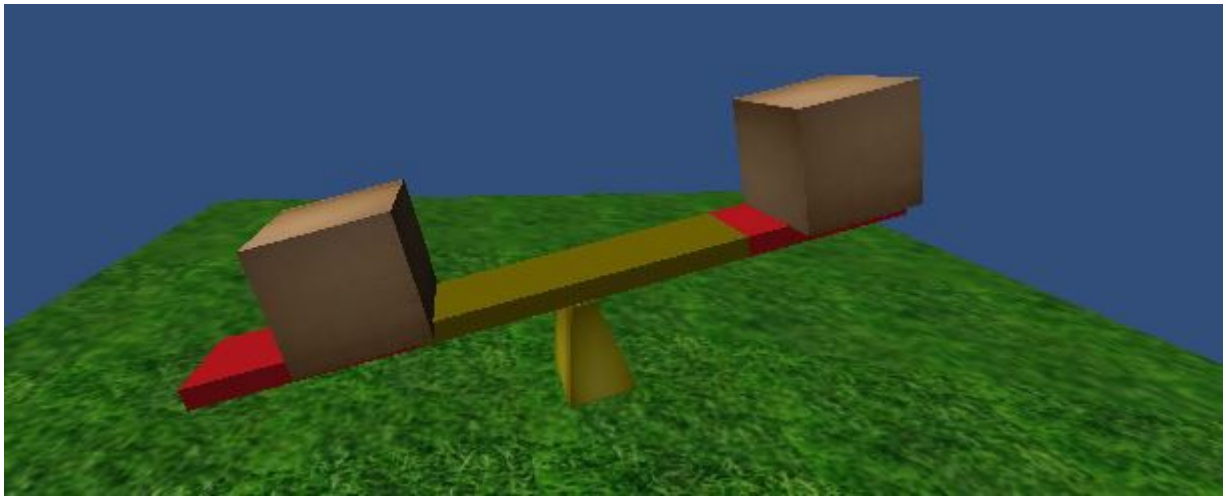




We can use Torque on objects, we need script for it.

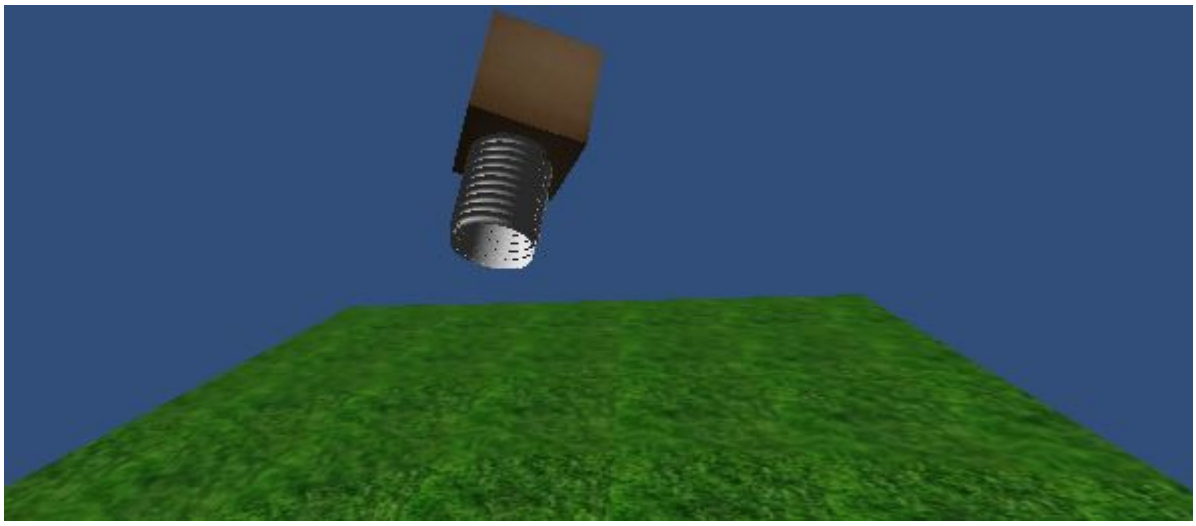
CHAPTER 6: EQUIBILIRIUM

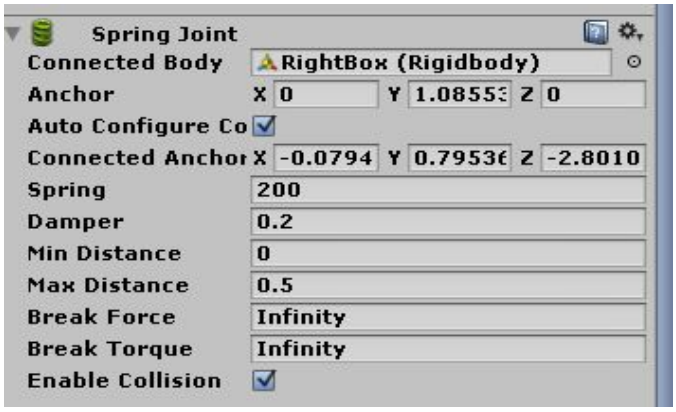




When we changed the mass of the left box, the change effects just like In real life in Unity Physics.

CHAPTER 7: SPRING





A Spring Joint can be used to create an invisible spring that tries to keep two rigid bodies together

Spring Joints allows a Rigidbodied GameObject to be pulled toward a particular “target” position. This position will either be another Rigidbodied GameObject or the world. As the GameObject travels further away from this “target” position, the Spring Joint applies forces that will pull it back to its original “target” position. This creates an effect very similar to a rubber band or a slingshot.

The “target” position of the Spring is determined by the relative position from the Anchor to the Connected Body (or the world) when the Spring Joint is created, or when Play mode is entered. This makes the Spring Joint very effective at setting up Jointed characters or objects in the Editor, but is harder to create push/pull spring behaviors in runtime through

scripting. If you want to primarily control a GameObject's position using a Spring Joint, it is best to create an empty GameObject with a Rigidbody, and set that to be the Connected Rigidbody of the Jointed object. Then in scripting you can change the position of theConnected Rigidbody and see your Spring move in the ways you expect.

Connected Rigidbody

You do not need to use a Connected Rigidbody for your joint to work. Generally, you should only use one if your object's position and/or rotation is dependent on it. If there is no Connected Rigidbody, your Spring will connect to the world.

Spring & Damper

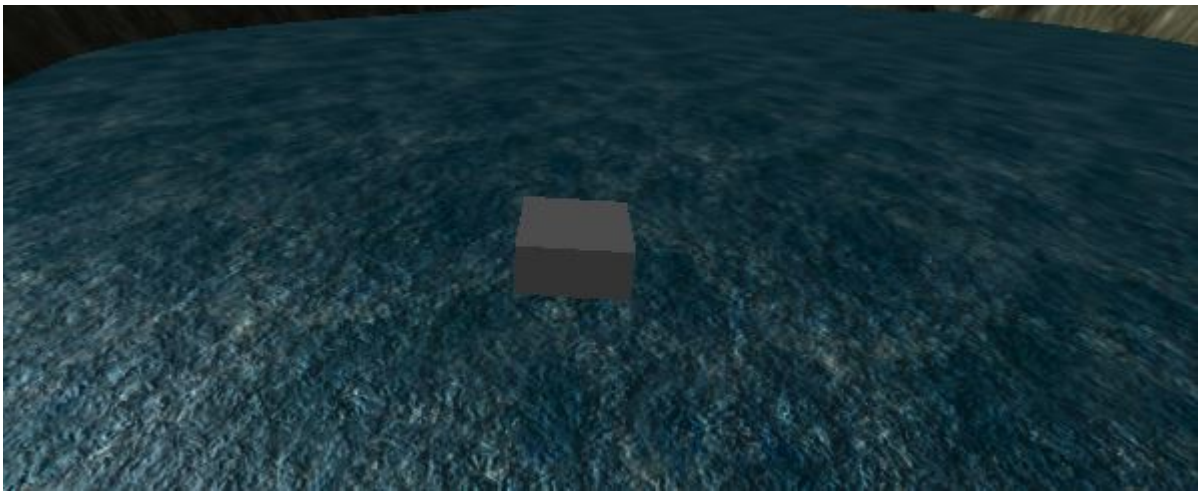
Spring is the strength of the force that draws the object back toward its "target" position. If this is 0, then there is no force that will pull on the object, and it will behave as if no Spring Joint is attached at all.

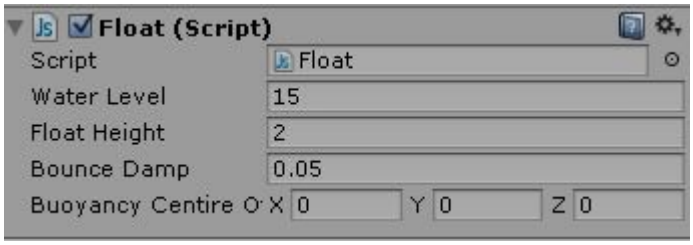
Damper is the resistance encountered by the Spring force. The lower this is, the springier the object will be. As the Damper is increased, the amount of bounciness caused by the Joint will be reduced.

Min & Max Distance

If the position of your object falls in-between the Min & Max Distances, then the Joint will not be applied to your object. The position must be moved outside of these values for the Joint to activate.

CHAPTER 8: FLOATING OBJECT





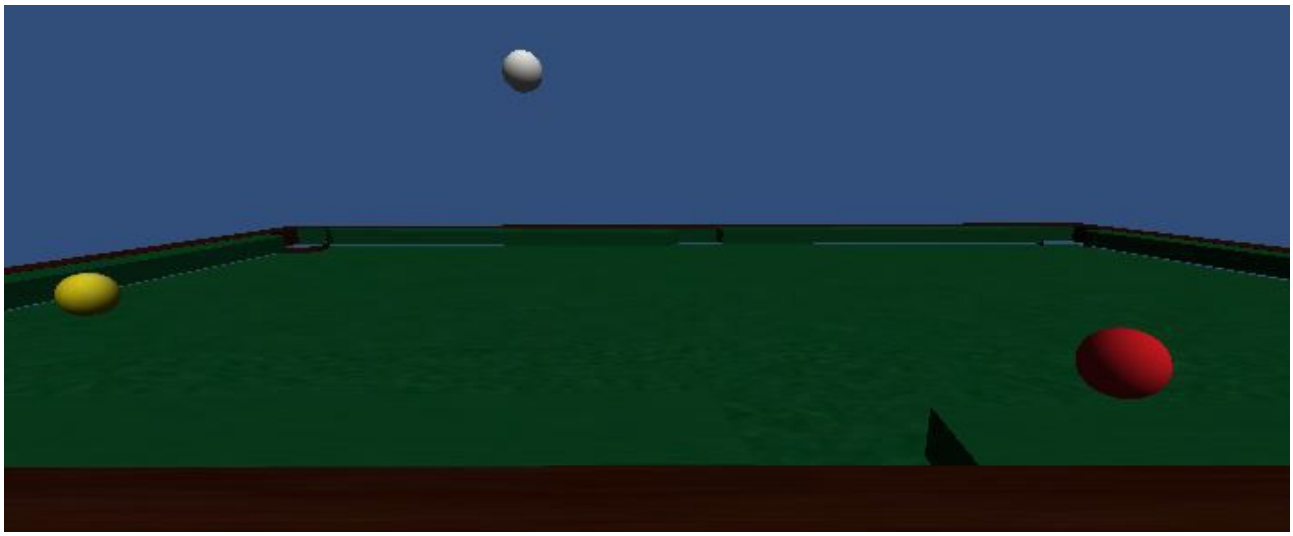
It's possible with a script. In this kind of projects, I always use, public variables that, easy to change parameters without coding again.

CHAPTER 9: BILLIARD

Adding Rigidbodies all the balls, and white ball is heavier than others. Pool itself also has a friction.

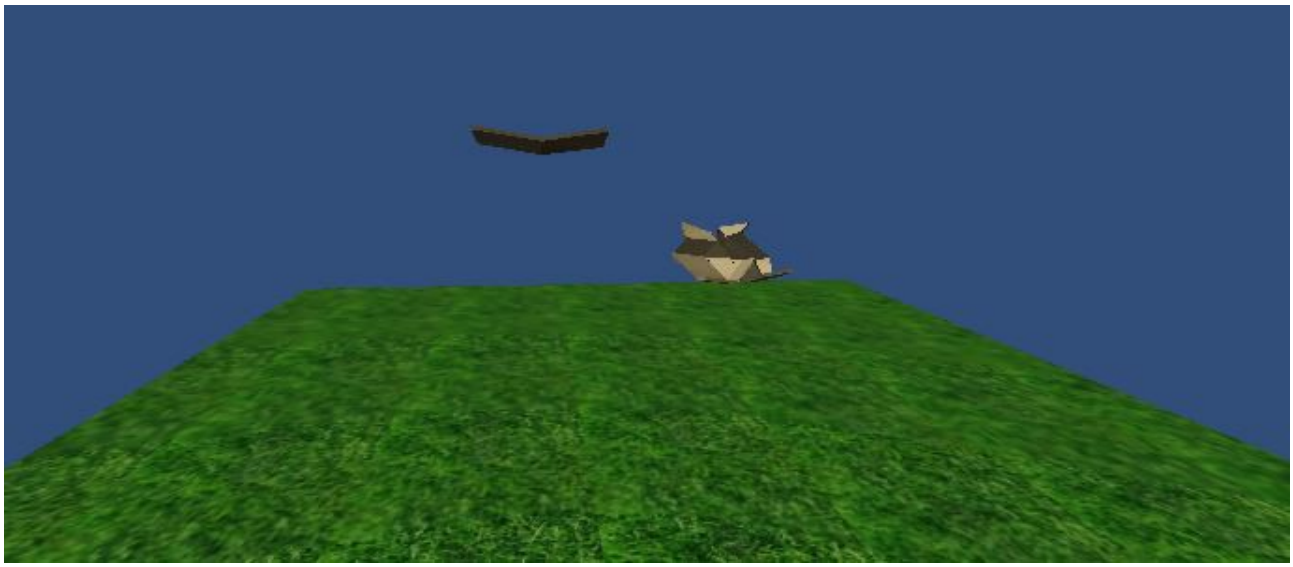
Using force on white ball: X, Y, Z axes by script.

CHAPTER 10: BILLIARD – PROJECTILE MOTION



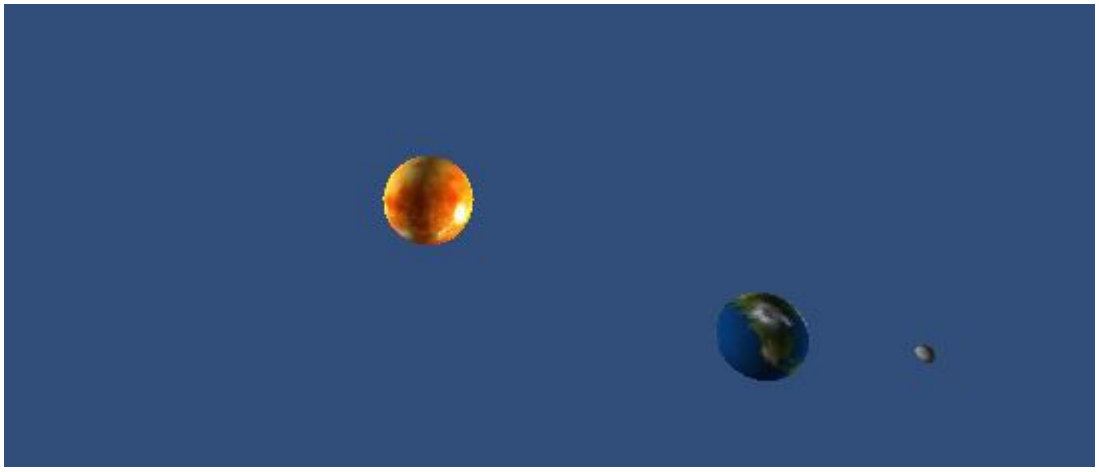
Using force on the whiteball, X and Y axes. So it will make a projectile motion, by using script.

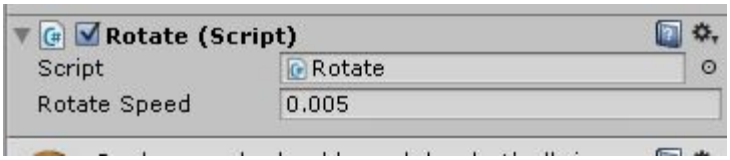
CHAPTER 11: AIR FRICTION



We observe here that air friction effects on the surface of the object, and paper ball lands to ground in a less time than bended paper. We define this properties by script again.

CHAPTER 12: SOLAR SYSTEM MODEL : SUN, EARTH, MOON





By wrting a script, we rotate the objects. In “ Parental System” of Unity, When parent Rotates, child rotates too.

Sun>Earth>Moon , my Rotate script effects on each object “rotating” but circling around by using Unity's parental system.