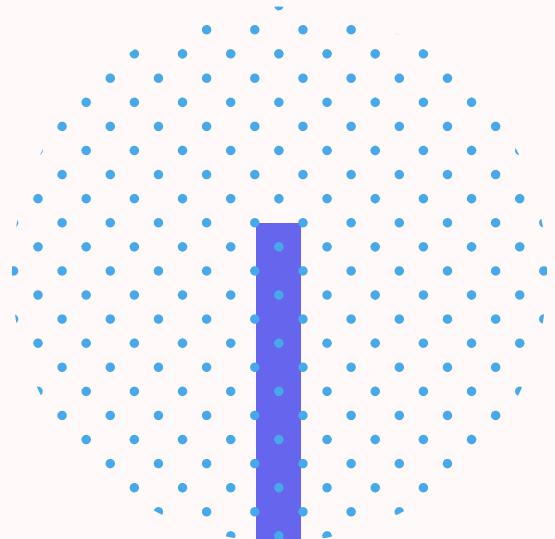


PHASE 3

ENGINE

GODBLESS ENGINE



Spring Class

- This class contains the necessary information in order to make a spring. It contains A vector and float as a variable. It also have its own update force, which varies depending on what spring you want to use.

Anchored Spring

-This class makes an anchored spring to simulate

Bungee Spring

-This class makes a spring similar to a bungee.

Particle Spring

-This class makes a particle into a spring

The three of them uses the same type of variables and function however; the logic on how they perform varies from one spring to another. The spring class has Update Force as a public function. Update Force adds a force to the particle. The Force that is being added to the particle depends on the characteristic of the spring. All of the other springs has an update force so the simulation of a certain spring can be realistically done with the help of some mathematics.

Particle Link

-this class connects two particle and get the distance of the connected particle. Particle Link has a public function get distance, which gets the distance between two particles and return the magnitude of the particle.

Cable

-This class contains two-float value as a variable. In this class, has a function Get Contact that checks whether the distance of two particle is greater than the length of the cable and returns the particle contact.

Rod

-similar to a cable this contains two float values and has a function that returns a particle contact. Similar to the cable class it checks the distance between two particle and checks whether the length of the rod is greater than or less than. This returns a particle contact.

Contact Resolver

-this resolves if objects collide to each other and simulate it properly in our world. The contact resolver has a function resolve contacts that checks a particle contact and rearrange the contact base on the lowest separating speed. After rearranging the particle contact, resolve the first one in the list until the last.

Force Generator

-This class contains the necessary information to generate force in our world. It contains a virtual function, which adds force to our particle class.

Drag Force Generator

-This class generate the drag force an object has and applies it to the object. It contains two float which are the drag constant of two objects.

Gravity Force Generator

-In this class, you can experiment with the gravity that is being applied in your physics world. This class adds gravity to your particle class.

After putting some drag and gravity values, it will update the force that the particle have and add the Drag and Gravity force to that particle.

Force Registry

-This class contains a structure, which have a particle and a force generator as its component. In this class, the user can add, remove and clear forces. This class also update the forces whenever you do something with it.

Utils

-is used to add offset value to the position of our object. Since the values of SFML is quite different from real life simulation Utils is used to make the position of the object more realistic. Utils have two public function which are Rotate point and converting physics to sfml point. Converting to physics to sfml point is used to make the simulation base on what it would look like in the real world. The coordinate system that sfml uses is different that is why this function is use to convert it similar to the Cartesian coordinate system. Rotate Point is used in order to rotate the point properly using the Cartesian coordinate system.

Render Particle

-this class is used to render the shape of the all particles. This class contains the draw function for our particles.

Physics World

-This class handles most of the physics and contains the information in order to apply physics in our world. This class is the brain that connects everything to make the object move within our world.

Physics World contains the following public function:

- Add particle – this adds a particle to the physics world
- Update – this update what happening in the physics world
- Add Contact – This add contact to two particle
- Get Contact – This gets contact between a particle and a rectangle ,a rectangle and rectangle or , a particle and a particle
- Generate Particle Contact – if two particle will collide this function will add it on the list of add contact.
- Generate Rigid Body Contact – this function checks whether a particle and a rectangle ,a rectangle and rectangle or , a particle and a particle will collide and add it to the list of contact

Particle Contact

-This class checks whether two objects will collide or not and fix the simulation to properly visualize what should really happened.

Particle contact has multiple public function:

- Get Separating Speed – this returns a velocity different multiplied by collision normal
- Resolve – this resolves both velocity and interpenetration
- Resolve Velocity – this resolve velocity of two particle
- Resolve Interpenetration – this resolves two object that will collide

My Vector

-This class handles the mathematical computations that are needed in the engine. This class has operator-overloading functions that lets the engine perform Vector classes operations.

My Vector contains a lot of public function:

- Get Magnitude – given two vector this returns the magnitude of it.
- Normalize – this normalize the vector
- Get Direction – returns the direction of a vector.
- Addition – this is an override function that allows to add 2 vector
- Subtraction – this is an override function that allows to subtract 2 vector
- Multiply – this is an override function that allows to multiply two vector
- Square Magnitude – returns the square magnitude of a vector
- Invert – this makes the a vector negative
- Component Product – this multiplies two vector which returns a vector ($a.x*b.x, a.y*b.y$)

My Particle

-This class has all the necessary things for our object. All information about our object can be find in this class.

Circle Rb

-This class tells my particle if it is a circle rigid body. Circle Rb containsa public function which allows it to set what particle type it is

RectPrism Rb

-This class tells my particle if it is a rectangle rigid body. Similar to circle Rb this contains a public function, which allows it to set its particle type. It also set the edges of the rectangle.

The public function that my particle contains:

- Add force – this add force to an object
- Reset force – this reset the force that the object has
- Get Moment of Inertia - this returns the moment of inertia of an object
- Add Force on Point – this adds force on a certain point of an object.
- Stationary Setting – this sets the object velocity to zero
- Get Type – this gets the type of object whether a particle , circle or rectangle
- Is Destroyed – this check whether a particle is destroyed or not it returns a bool
- Check life Span – this checks whether a life span of an object and if it runs out it destroy the object.
- Destroy – this returns true to destroy a particle
- Update – this updates the world and applies all of the thing in the simulation