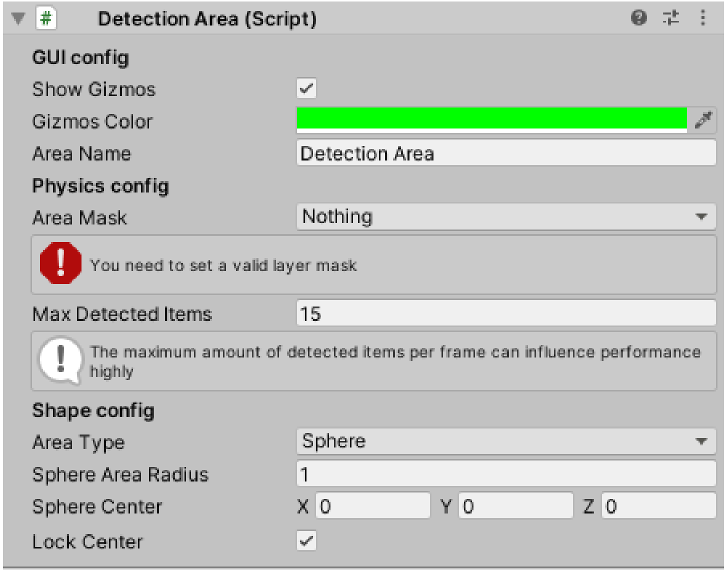
Simple Physics Tools

DOCUMENTATION

1. COMPONENTS

A) DetectionArea



Detection Area (Script)

GUI config

Show Gizmos ☑

Gizmos Color 🟩

Area Name: Detection Area

Physics config

Area Mask: Nothing

You need to set a valid layer mask

Max Detected Items: 15

The maximum amount of detected items per frame can influence performance highly

Shape config

Area Type: Sphere

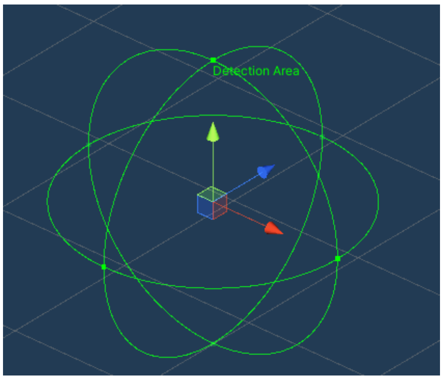
Sphere Area Radius: 1

Sphere Center: X 0 Y 0 Z 0

Lock Center ☑

This component allows you to customize a zone able to detect colliders in it.

1- GUI Config



The **Show Gizmos** option toggles the visibility of the zone shape and name on scene GUI

The **Gizmos color** option defines the shape and name color

The **Area Name** is the id text that will show on the shape, this name is purely for GUI purposes, it can be left blank

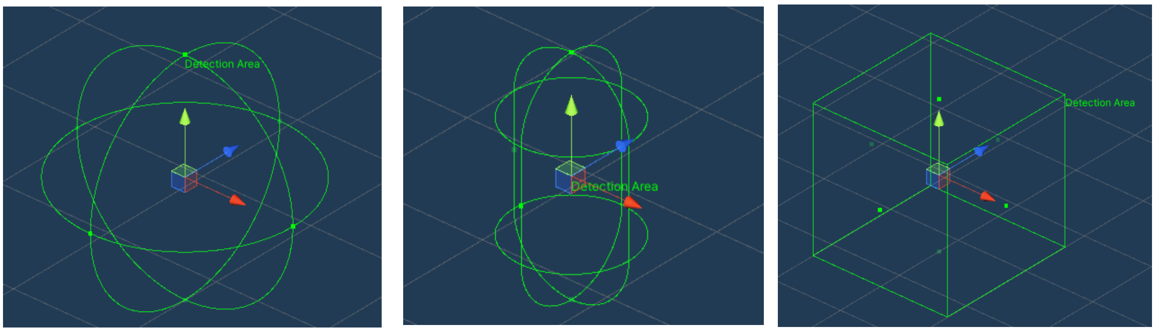
2- Physics Config

The **Area Mask** is a layer mask to filter the targeted collider layers, it is blank by default and should be filled for the component to work, for optimization purposes the layer mask should be as precise as possible, to avoid useless detections

The **Max Detected Items** number defines how many colliders can be detected on one frame, this value is used by the component to initialize a detection cache to avoid frame by frame instantiation. For optimization purposes this should be as small as possible.

3- Shape Config

The **AreaType** defines the shape type of the area, with three available values: **Sphere, Box, Capsule.**



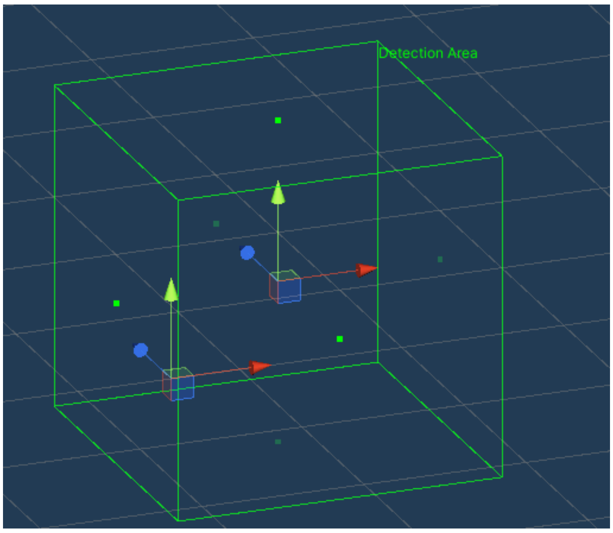
A box, Capsule and Box area

Each **AreaType** unlocks specific configuration for each shape:

- Sphere: a **Radius** is available.
- Capsule: a **Radius** is available as long as a **height** and an **Axis** defining the height orientation
- Box: a **Bounds** value (see unity documentation)

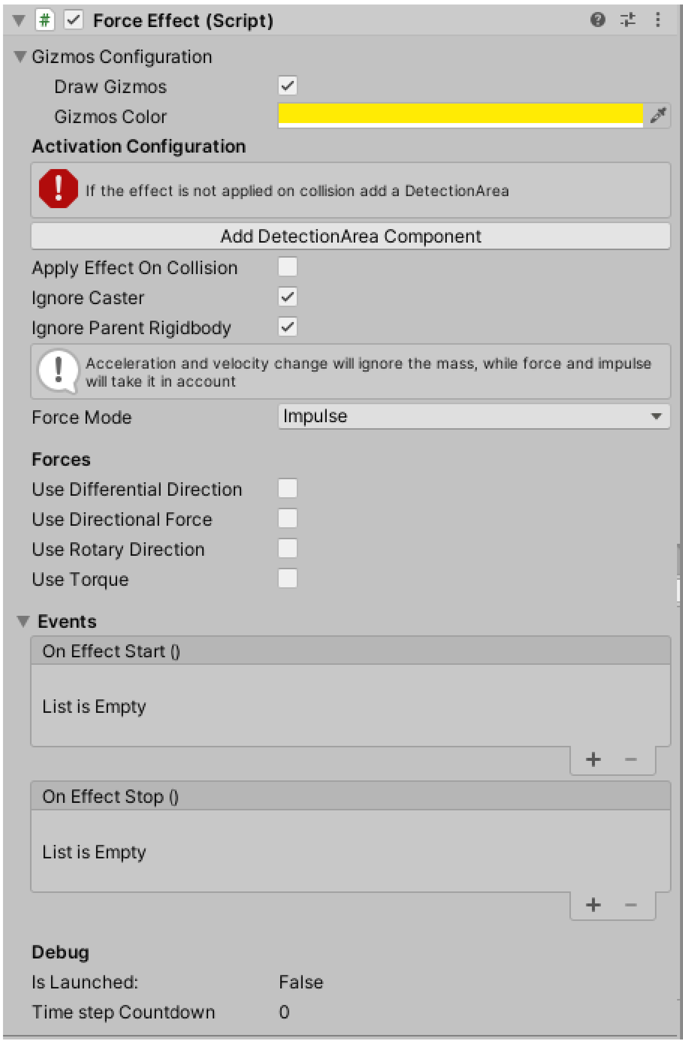
All those shapes are editable in the scene GUI using the handles, specifying the values in the inspector is not necessary

All those shapes also provide a **center** option allowing to move the shape around the gameObject, if the **Lock Center** option is disabled, the center can be moved from the Scene GUI.



The center of the zone and of the object are separated

B) ForceEffect



Force Effect (Script)

Gizmos Configuration

Draw Gizmos ☑

Gizmos Color

Activation Configuration

If the effect is not applied on collision add a DetectionArea

Add DetectionArea Component

Apply Effect On Collision □

Ignore Caster ☑

Ignore Parent Rigidbody ☑

Acceleration and velocity change will ignore the mass, while force and impulse will take it in account

Force Mode: Impulse

Forces

Use Differential Direction □

Use Directional Force □

Use Rotary Direction □

Use Torque □

Events

On Effect Start ()

List is Empty

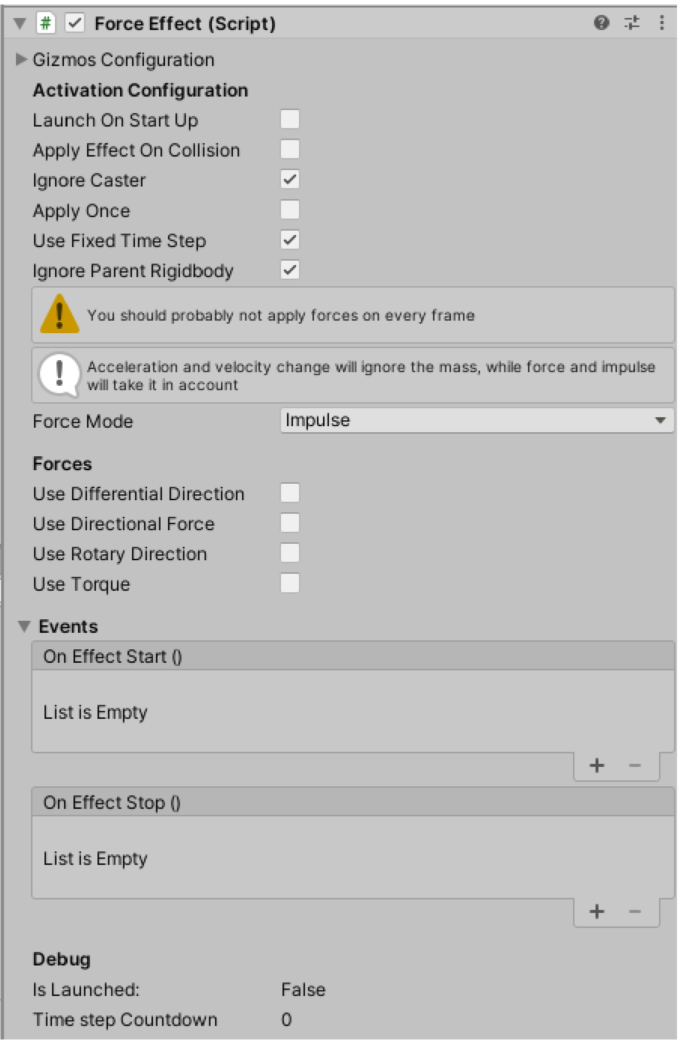
On Effect Stop ()

List is Empty

Debug

Is Launched: False

Time step Countdown 0



Force Effect (Script)

Gizmos Configuration

Activation Configuration

Launch On Start Up □

Apply Effect On Collision □

Ignore Caster ☑

Apply Once □

Use Fixed Time Step ☑

Ignore Parent Rigidbody ☑

You should probably not apply forces on every frame

Acceleration and velocity change will ignore the mass, while force and impulse will take it in account

Force Mode: Impulse

Forces

Use Differential Direction □

Use Directional Force □

Use Rotary Direction □

Use Torque □

Events

On Effect Start ()

List is Empty

On Effect Stop ()

List is Empty

Debug

Is Launched: False

Time step Countdown 0

This component allows you to apply forces to rigidbodies in a **DetectionArea** or on **Trigger** or **Collision** with a collider

1- Activation Configuration

The **launchOnStartup** option means that the component will begin to apply forces on **Start**. If set to false, the component can be started from events or script.

If the **ApplyEffectOnCollision** is set to true, the component requires a collider (trigger or not) otherwise it requires a DetectionArea. It can also use both but it is not recommended.

The **overrideIgnoreColliders** option is in case you launch this component through scripting and provide a colliders to ignore, for example if you use a **ForceEffect** as an activable AOE skill you'll want to ignore the caster and this option is here to override it. Tweaking the physics layer or the associated **DetectionArea AreaMask** can be useful too.

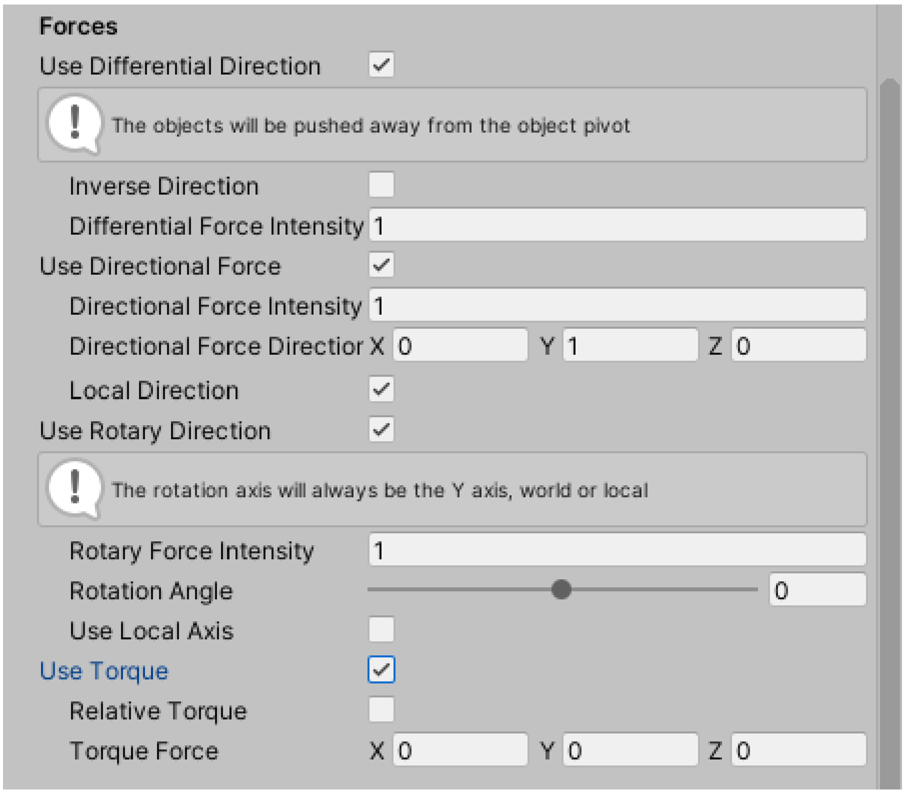
The **ApplyOnce** option will prevent the effect from looping, launched from startup or not, it allows more control from script.

The **UseFixedTimeStep** option defines if the force should be applied every physical frame, it set on false a custom timestep can be provided.

The **ignoreParentRigidbody** allows to filter out a parent rigidbody from the forces.

The **ForceMode** is a very important parameter, it defines what mode will be used the rigidbodies, see unity documentation for more information on the available force modes.

2- Forces



Forces

Use Differential Direction ☑

The objects will be pushed away from the object pivot

Inverse Direction □

Differential Force Intensity 1

Use Directional Force ☑

Directional Force Intensity 1

Directional Force Directior X 0 Y 1 Z 0

Local Direction ☑

Use Rotary Direction ☑

The rotation axis will always be the Y axis, world or local

Rotary Force Intensity 1

Rotation Angle 0

Use Local Axis □

Use Torque ☑

Relative Torque □

Torque Force X 0 Y 0 Z 0

DIFFERENTIAL DIRECTION

This force will take the object and detected collider positions in account, using the resulting direction. It allows to create a “repulsion” effect, if the **inverse direction** option is checked, the effect will be “attraction”.

The **differential force intensity** defines the strength of the applied force, relative to the selected forceMode.

DIRECTIONAL FORCE

The force direction can be simply specified as a **Vector3** but can be set to **localDirection** meaning that the direction will follow the object rotation, useful for dynamic force effect in skills.

The **Directional force intensity** defines the strength of the applied force, relative to the selected forceMode.

The force direction and strength will be shown on the scene GUI as an ArrowCap.

ROTARY DIRECTION

This force will make the target rigidbodies move around the Y axis (this axis can be the local one through **Use local axis** option) this effect creates a “tornado” like effect

The **rotary force intensity** defines the strength of the applied force, relative to the selected forceMode.

The **rotation angle** (shown as an arc on the scene GUI) can be set from -90° to 90° and defines how the force direction will be calculated (the target force point will be a (**rotation angle**) rotation around the Y axis relative to the target collider position.

3- Events

Two events are available to listen on the start of the effect and its stopping, these events will be triggered if the effect is started and stopped from script too.