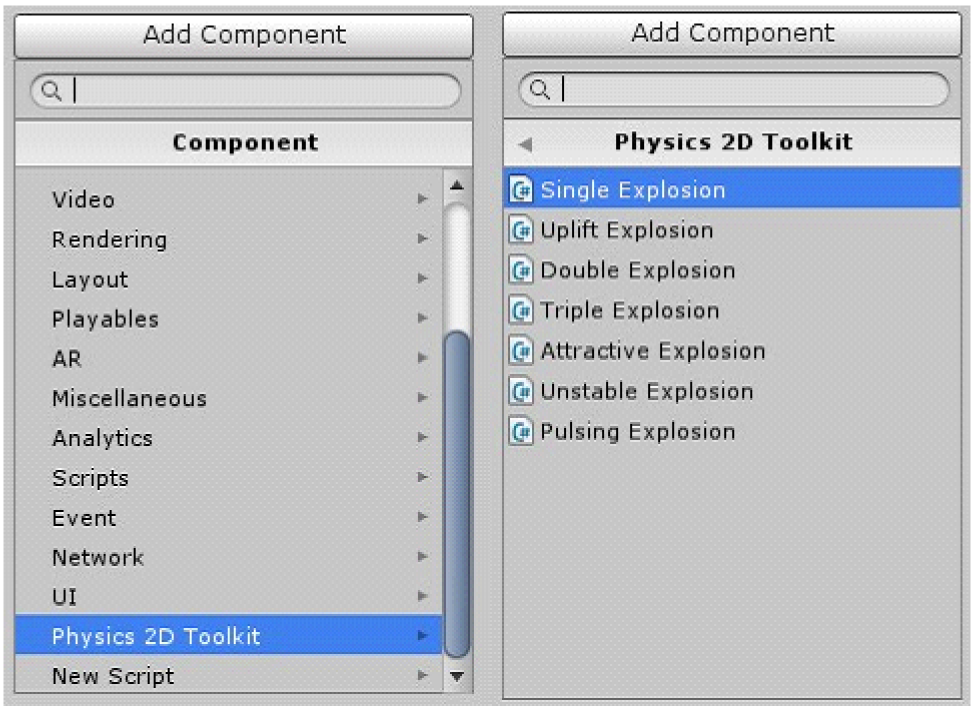
How To Use

Step 1:
Create some objects that will be affected by the explosion.
Those objects should have:

- -Rigidbody2D (Dynamic)
- -Collider2D
- -and SpriteRenderer

Step 2:
Make a simple BoxCollider2D at the bottom of the screen so that objects don’t fall down.

Step 3:
Create an empty Gameobject, and add the explosion component to that object.
For example: Add Component/Physics 2D Toolkit/ Single Explosion



Step 4:
Check the toolbox On Click. That means the explosion will be activated by the left mouse click.
You can also make reference to a SingleExplosion script form another script and call a method named: Activate();

Step 5:
Finally, start the game and press the left mouse click.
This is it. Enjoy! :)

Explosions

Single Explosion

Add Component/Physics 2D Toolkit/Single Explosion

This is a simple explosion that applies force to a rigidbody that simulates explosion effect.

Uplift Explosion

Add Component/Physics 2D Toolkit/Uplift Explosion

This is the same as simple explosion, but it contains one additional feature - 'Uplift Modifier'. Using this parameter, you can easily make the explosion appear to throw objects up, which often gives a more dramatic effect than a simple outward force.

Double Explosion

Add Component/Physics 2D Toolkit/Double Explosion

These are the two simple explosions. The first explosion is activated instantly, and the second explosion can be activated with delay by modifying the 'Second Explosion Delay' parameter.

Triple Explosion

Add Component/Physics 2D Toolkit/Triple Explosion

These are the three simple explosions. The first explosion is activated instantly, then the second explosion can be activated with delay by modifying the 'Second Explosion Delay' parameter, and the third explosion is delayed with the 'Third Explosion Delay' parameter.
NOTE! The 'Third Explosion Delay' parameter starts counting when the first explosion is activated (not the second)! - which means that the third explosion can be activated before the second!

Attractive Explosion

Add Component/Physics 2D Toolkit/Attractive Explosion

In order to achieve the effect of attractiveness, this is the best solution for you.
This explosion has two states:
1. Attraction: when activated, it will constantly attract rigidbodies towards the center of attraction. This attraction is used inside FixedUpdate().
2. Explosion: when the 'Explosion Delay' timer runs out, the attraction will stop, and the explosion will occur.

Unstable Explosion

Add Component/Physics 2D Toolkit/Unstable Explosion

This explosion has two states:
1. Shaking: when activated, based on the frequency settings, this will shake rigidbodies randomly.
2. Explosion: when the 'Explosion Delay' timer runs out, the shaking will stop, and the explosion will occur.

Pulsing Explosion

Add Component/Physics 2D Toolkit/Pulsing Explosion

This is a simple explosion that applies force to a rigidbody that simulates explosion effect. It can be useful for objects that constantly fall into the specific area, and need to be pushed away.

Settings

Filter Settings

Layer Filter: Chose layers that will be affected by the explosion.
Tag Filter: Chose tags that will be affected by the explosion.
Min Depth: Only include objects with a Z coordinate (depth), greater then or equal to this value.
Max Depth: Only include objects with a Z coordinate (depth), less then or equal to this value.
Use Gameobject Depth: Only include objects with a Z coordinate(depth) that are equal to this object Z coordinate(depth).
Use Trigger Colliders: Use trigger type of Collider2D.
Use Solid Colliders: Use solid type of Collider2D.

Finish Actions

Destroy Script: When the explosion is finished. Should this script automaticaly destroy itself?
Destroy Gameobject: When the explosion is finished. Should this Gameobject automaticaly destroy itself?
Deactivate Gameobject: When the explosion is finished. Should this Gameobject automaticaly deactivate itself?

Activate

On Click: Activate the explosion by the left mouse click ? This is useful for testing. But note that this works only in Editor!
On Enable: Activate the explosion when this Gameobject becomes enabled?
On Disable: Activate the explosion when this Gameobject becomes disabled?

Explosion Settings

Explosion Radius: Radius of the circle within witch the explosion has its effect.
Explosion Offset: The offset of the explosion position (World Space).
Explosion Force: The force of the explosion.
Modifie Force By Distance: if set to true, Force will be modified by distance, but Note that bodies with the center of mass outside the radius will not be affected by the explosion.
ForceMode2D: Use this to apply a certain type of force to a 2D RigidBody. There are two types of forces to apply: Force mode and Impulse Mode.

Methods

namespace: using ExplosionForce2D;

Rigidbody2D.AddExplosionForce2D

Parameters

force	The force of the explosion.
explosionPosition	The center of the circle witch the explosion has it's effect.
radius	Radius of the circle within witch the explosion has it's effect.
modifyForceByDistance	Force will be modified by distance.
lookAtMovingDirection	Instantly rotate Rigidbody2D towards it's moving direction.
lookAtAngleModifier	Modifies the angle of rotation
ForceMode2D	The method used to apply the force to it's target.

Description

- Applied a force to a Rigidbody2D that simulates explosion effects.
- The explosion is modelled as a circle with a certian center position and radius in world space.
- modifyForceByDistance if set to true, Force will be modified by distance, but Note that bodies with the center of mass outside the radius will not be affected by the explosion.
- lookAtMovingDirection if set to true, Rigidbody2D will be instantly rotated towards it's moving direction, not force direction! This is useful when using every frame. It is recommended to apply this feature to round objects.
- lookAtAngleModifier Modify the angle of rotation based on your initial sprites rotation.
- Force can be applied only to an active Rigidbody2D. If a GameObject is inactive, AddExplosionForce2D has no effect.

Rigidbody2D.AddUpliftedExplosionForce2D

Parameters

force	The force of the explosion.
explosionPosition	The center of the circle witch the explosion has it's effect.
radius	Radius of the circle within witch the explosion has it's effect.
upliftModifier	Adjustment to the apparent position of the explosion.
modifyForceByDistance	Force will be modified by distance.
lookAtMovingDirection	Instantly rotate Rigidbody2D towards it's moving direction.
lookAtAngleModifier	Modifies the angle of rotation
ForceMode2D	The method used to apply the force to it's target.

Description

- Applied a force to a Rigidbody2D that simulates explosion effects.
- The explosion is modelled as a circle with a certian center position and radius in world space.
- upliftModifier : Using this parameter, you can easily make the explosion appear to throw objects up, which often gives a more dramatic effect than a simple outward force.
- modifyForceByDistance if set to true, Force will be modified by distance, but Note that bodies with the center of mass outside the radius will not be affected by the explosion.
- lookAtMovingDirection if set to true, Rigidbody2D will be instantly rotated towards it's moving direction, not force direction! This is useful when using every frame. It is recommended to apply this feature to round objects.
- lookAtAngleModifier Modify the angle of rotation based on your initial sprites rotation.
- Force can be applied only to an active Rigidbody2D. If a GameObject is inactive, AddUpliftedExplosionForce2D has no effect.

Rigidbody2D.AddAttractionForce2D

Parameters

force	The force of the explosion.
attractionPosition	The center of the circle witch the attraction has it's effect.
radius	Radius of the circle within witch the attraction has it's effect.
modifyForceByDistance	Force will be modified by distance.
lookAtMovingDirection	Instantly rotate Rigidbody2D towards it's moving direction.
lookAtAngleModifier	Modifies the angle of rotation
ForceMode2D	The method used to apply the attraction force to it's target.

Description

- Applied a attraction force to a Rigidbody2D that simulates explosion effects.
- In order to achieve the effect of attractiveness, this is the best solution for you.
- The attraction is modeled as a circle with a certain center position and radius in world space.
- This Attraction Force should be used inside FixedUpdate().
- modifyForceByDistance if set to true, Force will be modified by distance, but Note that bodies with the center of mass outside the radius will not be affected by the explosion.
- lookAtMovingDirection if set to true, Rigidbody2D will be instantly rotated towards it's moving direction, not force direction! This is useful when using every frame. It is recommended to apply this feature to round objects.

- lookAtAngleModifier Modify the angle of rotation based on your initial sprites rotation.
- Attraction force can be applied only to an active Rigidbody2D. If a GameObject is inactive, AddAttractionForce2D has no effect.

Rigidbody2D.AddRandomizedExplosionForce2D

Parameters

force	The force of the explosion.
explosionPosition	The center of the circle witch the attraction has it's effect.
radius	Radius of the circle within witch the attraction has it's effect.
modifyForceByDistance	Force will be modified by distance.
randomizeDirection	Multiply force direction randomly with 1 or -1.
randomizeForce	Multiply force with random value between 0 and 1.
ForceMode2D	The method used to apply the attraction force to it's target.

Description

- Applied a random force to a Rigidbody2D that simulates explosion effects.
- The explosion is modeled as a circle with a certain center position and radius in world space.
- randomizeDirection if set to true, the direction of the force will be multiplied with random sign (-1 or 1).
- randomizeForce if set to true, Force will be multiplied with random value between 0 and 1.
- modifyForceByDistance if set to true, Force will be modified by distance, but Note that bodies with the center of mass outside the radius will not be affected by the explosion.
- explosion force can be applied only to an active Rigidbody2D. If a GameObject is inactive, AddAttractionForce2D has no effect.

Send Explosion Damage

You can send custom explosion damage. This feature uses the Unity 'SendMessage()' method.

- Explosion Damage: is the custom damage and has nothing to do with explosion force. You can specify this, for example, based on enemy type.
- Modify Damage By Distance: if set to true, 'Explosion Damage' will be modified based on the distance between rigidbodies and the explosion center.
- Method To Call: is the name of the method to call on a targeted Gameobject that holds the Rigidbody2D component affected with explosion.
- Options : 'Require Reciver' or 'Dont Require Reciver' . Should an error be raised if the method doesn't exist on the target Gameobject?

Example

```
using UnityEngine;

public class Enemy : MonoBehaviour {

        public float health = 100f;

        public void TakeExplosionDamage (float damage) {
                health -= damage;
        }
}
```

Description

When an enemy, that has Rigidbody2D and this class attached, is affected by the explosion, the 'TakeExplosionDamage' method will be called, and the 'Explosion Damage' that we specified will be passed as 'damage' parameter. Let's say our enemy has 100 healths, and we setup the explosion like this:

So after the explosion occure the enemy will have 60 healths left.

Radius Color

You can change the color of the explosion radius by opening the script **'ExplosionForce2DPreferences'** which is located in:
Assets/ExplosionForce2D/Scripts/Main

Physics2D.OverlapCircleNonAlloc

By default, all explosions use Physics2D.OverlapCircleAll, and if you want to use Physics2D.OverlapCircleNonAlloc, there is a comment example in every explosion at the bottom of the script, under the #region OverlapCircleNonAlloc.

The disadvantage of using OverlapCircleNonAlloc is that the explosion is limited to a certain number of rigidbodies that can be affected. Basically, you must specify the max number of rigidbodies.
The advantage of using OverlapCircleNonAlloc is that no memory is allocated for the result, so garbage collection performance is improved when the check is performed frequently.