

# ShockWave Effect

---

[Asset Store Link](#)

© 2017 Justin Garza

PLEASE LEAVE A REVIEW OR RATE THE PACKAGE IF YOU FIND IT USEFUL! Enjoy! :)

## Contact

---

Questions, suggestions, help needed?

Contact me at:

Email: [jgarza9788@gmail.com](mailto:jgarza9788@gmail.com)

Cell: 1-818-251-0647

Contact Info: [justingarza.info/contact](http://justingarza.info/contact)

## Description/Features

---

Awesome ShockWave Effect!

- Add it to any script with just one line of code!
- Create ShockWaves (and Reverse ShockWaves)
- Customize your ShockWave style with AnimationCurves
- Easily Pause/UnPause Shockwaves
- Unity Free friendly.
- Fully commented C# code.
- Awesome demos!

## Terms of Use

---

You are free to add this asset to any game you'd like However:

please put my name in the credits, or in the special thanks section. :)

please do not re-distribute.

## Table of Contents

---

1. ShockWave.cs

- Speed,Radius,Amplitude,WaveSize
- Methods
- ShockWave\_WorldSpace.cs

2. Demo1
3. Demo2
4. Demo3
5. Demo4
6. Demo5

## ShockWave.cs

---

ShockWave.cs is the main script that creates and manages the shockwaves.

### Speed,Radius,Amplitude,WaveSize

these are the 4 values that we can adjust to change the look, and style of the shockwave.

#### **Speed:**

this is just the speed at which the animation will play.

#### **Radius:**

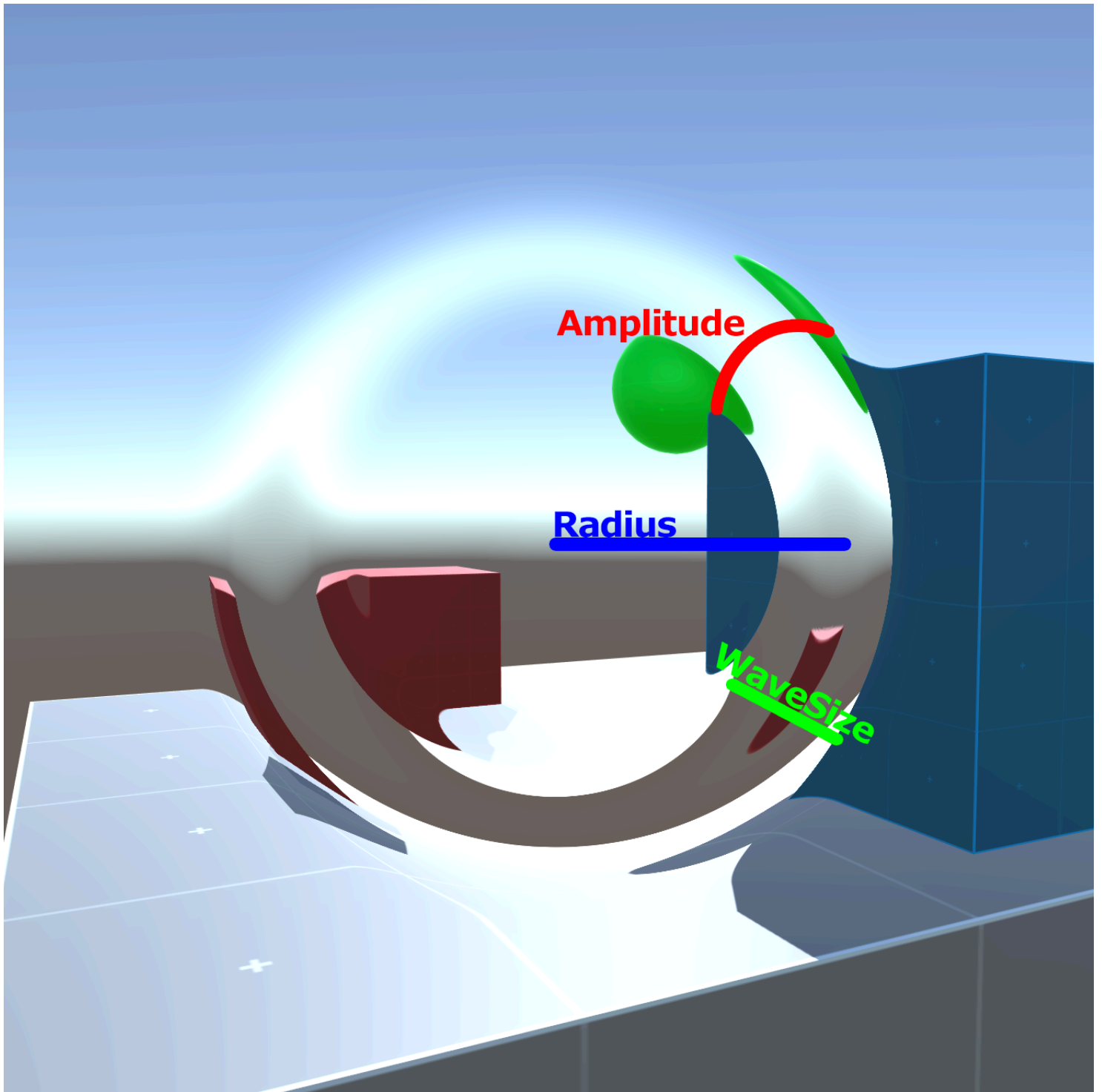
the distance between the center of the circle to it's edge.

#### **Amplitude:**

the distortion amount along the edge of the shockwave.

#### **WaveSize:**

this is the thickness of the shockwave.



## Methods

There are 12 methods that can be used to display a shockwave.

**This first set will allow you to pass in a Position (or a GameObject), and floats for Speed, MaxRadius, Amplitude, and WaveSize**

```
public void StartIt(Vector2 Position, float Speed = 1f, float MaxRadius = 1f, float Amplitude = 1f, float WaveSize = 1f) {
```

```
public void StartIt(Vector3 Position, float Speed = 1f, float MaxRadius = 1f, float Amplitude = 1f, float WaveSize = 1f) {
```

```
public void StartIt(Vector3 Position, bool IsScreenPosition, float Speed = 1f, float MaxRadius = 1f, float Amplitude = 1f, float WaveSize = 1f) {
```

```
public void StartIt(GameObject Target, float Speed = 1f, float MaxRadius = 1f, float Amplitude = 1f, float WaveSize = 1f) {
```

Similar to the first set, this Second set will allow you to pass in a Position (or a GameObject), and floats for Speed, MaxRadius, Amplitude, and WaveSize. However the animation will play in reverse

```
public void ReverseIt(Vector2 Position, float Speed = 1f, float MaxRadius = 1f, float Amplitude = 1f, float WaveSize = 1f) {
```

```
public void ReverseIt(Vector3 Position, float Speed = 1f, float MaxRadius = 1f, float Amplitude = 1f, float WaveSize = 1f) {
```

```
public void ReverseIt(Vector3 Position, bool IsScreenPosition, float Speed = 1f, float MaxRadius = 1f, float Amplitude = 1f, float WaveSize = 1f) {
```

```
public void ReverseIt(GameObject Target, float Speed = 1f, float MaxRadius = 1f, float Amplitude = 1f, float WaveSize = 1f) {
```

If you want more control over how the shockwave looks, the last set will allow you to pass in animationCurves

```
public void StartIt(Vector2 Position, float Speed = 1f, AnimationCurve radiusOverTime = null, float Amplitude = 1f, float WaveSize = 1f) {
```

```
public void StartIt(Vector3 Position, float Speed = 1f, AnimationCurve radiusOverTime = null, float Amplitude = 1f, float WaveSize = 1f) {
```

```
public void StartIt(Vector3 Position, bool IsScreenPosition, float Speed = 1f, AnimationCurve radiusOverTime = null, float Amplitude = 1f, float WaveSize = 1f) {
```

```
public void StartIt(GameObject Target, float Speed = 1f, AnimationCurve radiusOverTime = null, float Amplitude = 1f, float WaveSize = 1f) {
```

## ShockWave\_WorldSpace.cs

Is a script much like ShockWave.cs however it is used in the ShockWave(WorldSpace).prefab.

Please view Demo-Scene4 and read ShockWave\_WorldSpace.cs for more info.

## Demo1

---

In demo1 we are creating shockwaves based on the value of the sliders, and the value of the reverse toggle.

```
...
void Update ()
{
    if (Input.GetMouseButtonDown(0))
    {
        if (RevSW)
        {
            ShockWave.Get().ReverseIt(Input.mousePosition,true,Speed,MaxRadius, Amp ,WS);
        }
        else
        {
            ShockWave.Get().StartIt(Input.mousePosition,true,Speed,MaxRadius, Amp, WS);
        }
    }
}
...
```

C#

## Demo2

---

In demo2 we are creating shockwaves not calling StartIt (or Reverselt), and then setting the radius, amplitude, and waveSize to a random float.

This technique can be done if you want a lot of control over the shockwave, for example a game that uses 3D Touch.

C#

```

...
void Update()
{
    if (Input.GetMouseButtonDown(0))
    {
        SW = ShockWave.Get();
        SW.SetPosition(Input.mousePosition, true);
        SW.radius = Random.Range(0.05f, 0.2f);
        SW.amplitude = Random.Range(0.05f, 0.2f);
        SW.waveSize = Random.Range(0.05f, 0.2f);
    }
}
...

```

## Demo3

Demo3 is a testing scene for the ShockWaves using AnimationCurves. Edit the values in "ShockWaveMaker" GameObject, Play the scene, then click around.

C#

```

...
public float speed = 1f;
public AnimationCurve radiusOverTime;
public AnimationCurve amplitudeOverTime;
public AnimationCurve waveSizeOverTime;

void Update()
{
    if (Input.GetMouseButtonDown(0))
    {
        ShockWave.Get().StartIt(Input.mousePosition, true, speed, radiusOverTime, amplitudeOverTime, waveSizeOverTime);
    }
}
...

```

## Demo4

In demo4 the projectiles are creating shockwaves on collision. These shockwaves are rendered on quads within the workspace (not on the screen). These shockwaves can be customized by changing the values

ShockWave(WorldSpace).prefab.

The code below shows how these shockwaves can be created.

```

/// <summary>
/// Raises the collision enter event.
/// </summary>
/// <param name="collision">Collision.</param>
void OnCollisionEnter(Collision collision)
{
    //get the direction the projectile is moving
    Vector3 dir = collision.contacts[0].point - gameObject.transform.position;

    //create a Ray
    Ray ray = new Ray(gameObject.transform.position, dir);

    //the output value
    RaycastHit hit;

    //use the raycast
    if (Physics.Raycast (ray, out hit, 100, mask))
    {
        //create a shockwave by creating an object from a prefab
        GameObject obj = GameObject.Instantiate(prefab);

        //move the shockwave to the correct location and rotation
        obj.transform.rotation = Quaternion.FromToRotation (Vector3.forward, hit.normal);
        obj.transform.position = hit.point;
        obj.transform.position += obj.transform.forward * 0.1f;

        //if you need an example of how to use delegates...
        /*
        obj.GetComponent<ShockWave_WorldSpace>().OnAnimationComplete += delegateExample.pr
        obj.GetComponent<ShockWave_WorldSpace>().OnAnimationComplete += delegateExample.pr
        obj.GetComponent<ShockWave_WorldSpace>().OnAnimationComplete += delegateExample.re
        */
    }

    //make the particleSystem
    GameObject.Instantiate(particleSystem, gameObject.transform.position, gameObject.transfo

    //destory the projectile
    Destroy(gameObject);
}
...

```

## Demo5



In demo5 the Jump Demo! Spacebar makes the cube jump and you see the shockwave as an effect.

The code below shows how the cube jumps and generates a shockwaves.

```
// Update is called once per frame
void Update ()
{
    if (Input.GetKeyDown(KeyCode.Space))
    {
        gameObject.GetComponent<Rigidbody>().AddForce(0f, jumpForce, 0f);

        //create a shockwave by creating an object from a prefab
        GameObject obj = GameObject.Instantiate(prefab);

        //move the shockwave to the correct location and rotation
        //obj.transform.rotation = Quaternion.FromToRotation(Vector3.forward, hit.normal);
        //obj.transform.position = hit.point;
        obj.transform.rotation = Quaternion.Euler(90f, 0f, 0f);
        obj.transform.position = transform.position + swPositionOffset;
    }
}
```

C#