# Fire Tutorial

**Note**: Requires both the Fire.js file and the flameparticle.png in the assets folder.

## Introduction

This particle system will allow to create and manipulate a permanent fire particle emitter that can be used to create the appearance of a fire. Here’s an example of one.



This fire is a basic fire that can be used in many scenarios. Now let’s go over how a fire such as this can be created

## Structure of Fire class

Fire(xPos, yPos, width, height, life, xVelocity, yVelocity, flicker, intensity, xAcceleration, size, yOffset)

This is the constructor header for the Fire class. Allow me to briefly explain what each parameter does to affect the fire.

xPos: The X Position that the object will be based on

yPos: The Y Position that the object will be based on

width(xOffset): The maximum Horizontal offset that fire particles can spawn from

yAcceleration: The vertical acceleration of the fire particles

life: The life of the fire particles that the emitters spawn

xVelocity: The maximum starting horizontal velocity for the fire particles

yVelocity: The maximum starting vertical velocity for the fire particles

flicker: How quickly the particles shrink

intensity: How many particles are created every update

xAcceleration: The horizontal acceleration of the fire particles

size: The starting size of the fire particles

yOffset: the maximum Vertical offset that fire particles can spawn from

A fire object creates a fire particle and modifies all these variables are used in creating the fire like effect that you desire.

To make a basic fire here are the parameter values that are needed.

|  |  |
| --- | --- |
| Width | 3 |
| yAcceleration | 2 |
| Life | 20 |
| xVelocity | 0 |
| yVelocity | 20 |
| Flicker | 4 |
| Intensity | 1 |
| xAcceleration | 0 |
| Size | 2.5 |
| yOffset | 0 |

The parameter values for a basic fire

Now, let’s get into a bit more detail with these parameters.

### Width (xOffset)

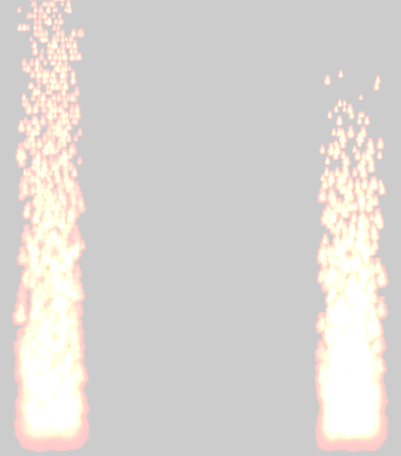
By increasing this parameter, you can increase the width of the fire. As you increase the width, you will also need to increase the intensity of the fire to compensate.

### Acceleration

yAcceleration increases the speed that the fire particles raise up into the air. Thus, it’s very useful in creating the appearance of very powerful flames. Controlling the height while increasing the speed will require you to lower the life of the particles. This however comes with the loss of the natural fading of the particles. This also applies for xAcceleration.

### Life

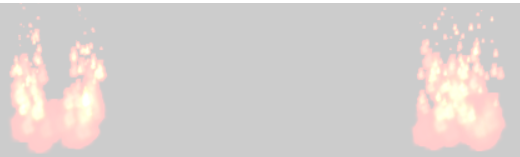
Useful for giving bigger fire particles more time to naturally fade, or for controlling the size of more intense fires.



The one on the right has 1/4th the life of the left one. The particles for the right fire don’t pass the screen due to that.

### Velocity

The velocity variables are useful for creating unusual fires. Take this image for example.



The xVelocity for the left fire is 10.

The flame particles will initially move away from base of the fire. If it spawns to the right of it, it will move right. The reverse applies if it spawns to the left. The higher you make the value, the farther the particles will be pushed from the base.

yVelocity will cause particles to shoot up faster initially. Since this is the maximum velocity that a particle can have, it creates the effect that particles in the foreground shoot up much faster than those in the background.

The acceleration will overpower the velocity if the difference between them is big enough.

### Flicker

By increasing this parameter, particles will shrink faster. which means that particles have less of that white center to them. The fire itself will also appear to be shorter due to the particles shrinking faster. The fire to the right is an example of this

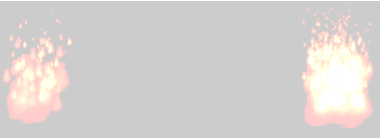


The effects that increasing the flicker by 8 will have

If the value is 0, then particles don’t shrink during their lifespan. If the flicker is negative, then particles grow in size rather than shrink. If the value is too high the entire screen goes white due to the eventual size of the particle.

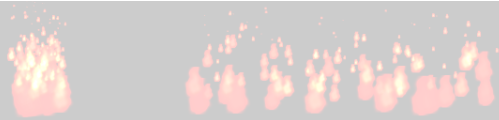
### Intensity

This is the number of fire particles that are created every update cycle. As you increase this parameter the fire becomes denser with particles and thus whiter.



The effects of increasing the intensity by 5

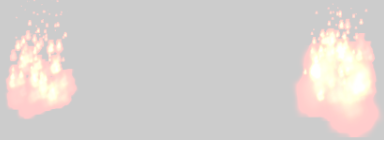
This particle is also very important to increasing the width and height of the fires. Increasing the offset merely increase the horizontal/vertical range where particles can spawn. Without an increase in intensity as well, you lose the fire effect and will instead just look like a bunch of particles spawning at random points.



An example of this effect

### Size

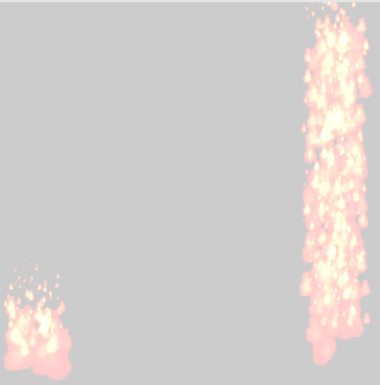
Increasing the particle’s size will also cause the base of the fire to be bigger and will make the red part of the fire’s base more apparent.



The right fire has its size increased by 8

### yOffset

This is similar to xOffset but instead is for vertical range. There is a difference in how the vertical offset is calculated per particle. Rather than using a base number, it just takes a random percentage of the yOffset and makes that the vertical offset of that particle. Because of that, the higher the value is, the less consistent the base of the fire becomes.



Example of this

To compensate for this, you can use an object to hide the base, or lower the yPos of the Fire Object for more desirable positioning.

Now let’s look at 3 other types of fires to gain a better understanding of how these parameters interact with each other.

## Torch Fire



|  |  |
| --- | --- |
| Width | 0 |
| yAcceleration | 0 |
| Life | 20 |
| xVelocity | 0 |
| yVelocity | 20 |
| Flicker | 32 |
| Intensity | 1 |
| xAcceleration | 0 |
| Size | 2.5 |
| yOffset | 0 |

By shrinking the width and yAcceleration parameters, you get a thin, slower burning fire. The increase of the flicker parameter is what causes it to shrink to such a degree. Its reduction reduces both the width and height of the fire by about half, and the fire also is far less intense due to how quickly it shrinks.

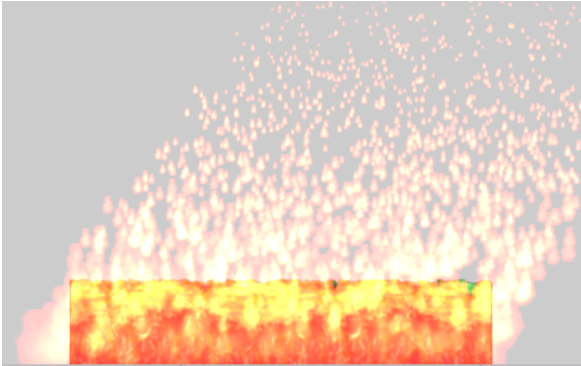
## Pillar Fire



|  |  |
| --- | --- |
| Width | 3 |
| yAcceleration | 36 |
| Life | 8 |
| xVelocity | 0 |
| yVelocity | 0 |
| Flicker | 2 |
| Intensity | 15 |
| xAcceleration | 0 |
| Size | 2.5 |
| yOffset | 1 |

The 2 most impactful parameters are the yAcceleration and the intensity. To achieve that blazing, solid white fire effect, you need the particles to rise very fast, and you need lots of particles to spawn at once. The flicker parameter is set to 2 to make it shrink very slowly, while still keeping the fire like effect where it is shrinking.

## Forest Fire



|  |  |
| --- | --- |
| Width | 23 |
| yAcceleration | 11 |
| Life | 20 |
| xVelocity | 0 |
| yVelocity | 12 |
| Flicker | 4 |
| Intensity | 23 |
| xAcceleration | 8 |
| Size | 3.5 |
| yOffset | 0 |

This requires raising both the width and intensity parameter to have a consistently wider fire. To have a “blowing in the wind” effect to the fire, you would raise both x and y acceleration until you reach the wind speed that you desire.

## Conclusion

This particle system will allow for the creation of fires for the usage in games. With the knowledge you gained from this tutorial, you’re now capable of finding further applications for this fire particle system.