What is included

At a minimum, all that is required are the scripts located in the "Scripts" directory. Prefabs of all the common GameObject's required, if you don't want to make your own, to quickly get started are located in the "Prefabs" directory.

A demo scene has been created and included in the "Scenes" directory, which includes multiple flammable and non-flammable terrain textures and objects. The demo shows all of the fire propagation's systems being used. All of the demo assets are in the "Demo" directory.

Get started

The following is based on starting a new Unity project if you already have a project which you would like to add fire propagation to some of the requirements may already have been met.

Requirements:

- 1 Terrain GameObject (with two terrain textures and grass details).
- 1 Fire Manager, child of the Terrain object.
- A GameObject with a FireIgniter script attached to create fires.
- 1 Prefab with Particle Systems, to be used to create fires.

Optional:

- Windzone.
- 'Fire' tag set in Editor, can still work without the Fire tag but this could affect performance.

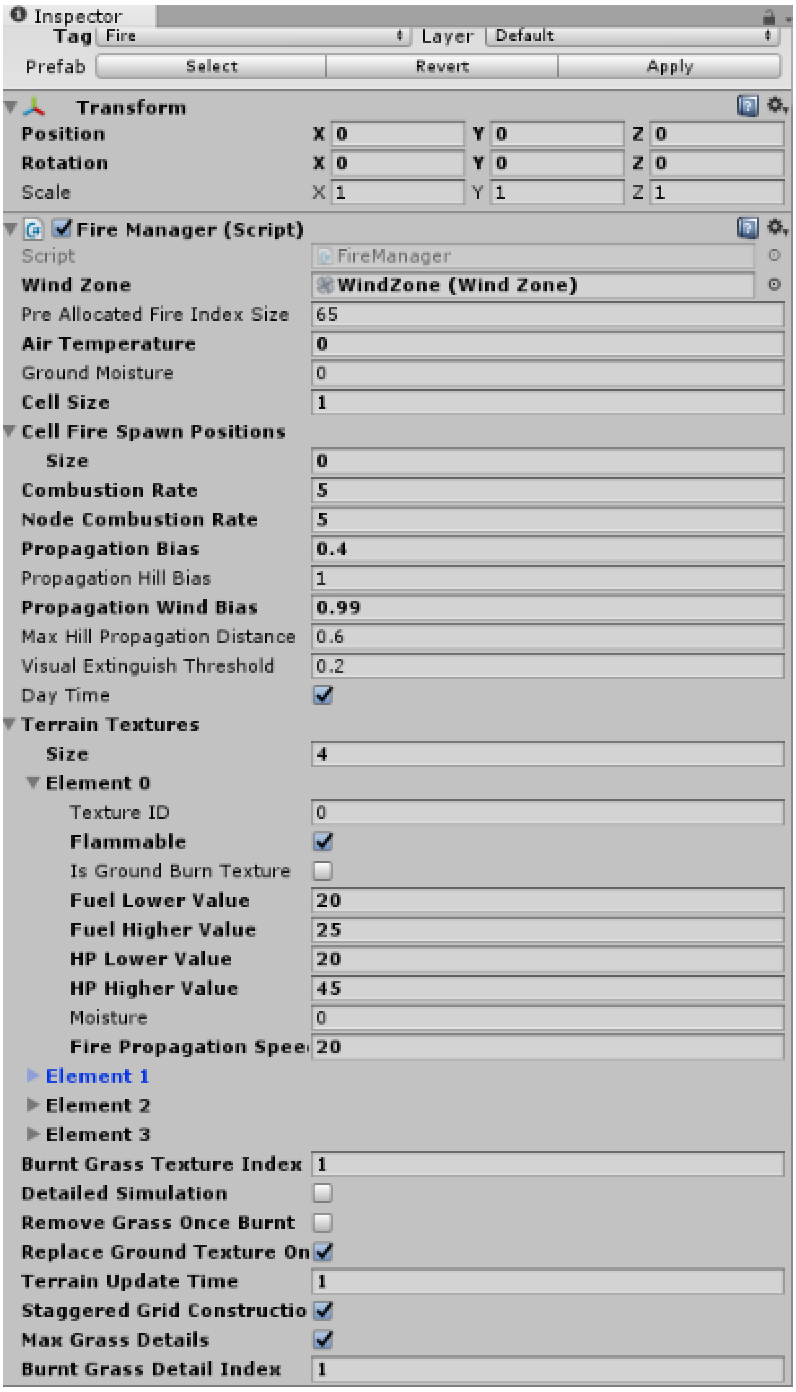
Make a Fire Manager GameObject a child of the terrain in the scene, if your terrain is a mesh the fire propagation will not work it has to be a Unity terrain. Set the fuel and simulation values along with the terrain textures in the Fire Manager, these map to the textures applied to the terrain.

Each of the variables has tooltips in the editor for more details about what each one does. The variables descriptions are also included in the Fire Manager Section of this document. For a video of getting started, go to YouTube.

Setting types of fuel:

Terrain textures are the primary fuel, for each terrain texture you will need to set the fuel values in the Fire Manager. If a terrain texture is not flammable the only variable that needs to be set is the "Texture ID" which is the array index of the texture in the terrain texture order; 0 -> n. There should only be a single scorch mark texture for the terrain, to set this tick the "Is Ground Burn Texture".

A grid is created when a Fire Igniter collides with a terrain, however, the fire will only propagate over the terrain. In order to allow the fire to interact, ignite and propagate over objects in the world, ignitable GameObjects need to have a Fire Node Chain script attached to them and Fire Node's placed inside or on the object.



Inspector

Tag	Fire	Layer	Default			
Prefab	Select	Revert	Apply			
Transform						
Position	X	0	Y	0	Z	0
Rotation	X	0	Y	0	Z	0
Scale	X	1	Y	1	Z	1

☑ Fire Manager (Script)	
Script	FireManager
Wind Zone	WindZone (Wind Zone)
Pre Allocated Fire Index Size	65
Air Temperature	0
Ground Moisture	0
Cell Size	1
Cell Fire Spawn Positions	
Size	0
Combustion Rate	5
Node Combustion Rate	5
Propagation Bias	0.4
Propagation Hill Bias	1
Propagation Wind Bias	0.99
Max Hill Propagation Distance	0.6
Visual Extinguish Threshold	0.2
Day Time	☑
Terrain Textures	
Size	4
Element 0	
Texture ID	0
Flammable	☑
Is Ground Burn Texture	☐
Fuel Lower Value	20
Fuel Higher Value	25
HP Lower Value	20
HP Higher Value	45
Moisture	0
Fire Propagation Spee	20
Element 1	
Element 2	
Element 3	
Burnt Grass Texture Index	1
Detailed Simulation	☐
Remove Grass Once Burnt	☐
Replace Ground Texture On	☑
Terrain Update Time	1
Staggered Grid Constructio	☑
Max Grass Details	☑
Burnt Grass Detail Index	1

Fire Manager

The Fire Manager settings are global across all fires in the scene, some individual settings for fires can be set on the Fire Igniter. To fully understand how to use the Fire Manager to achieve the outcome you want, some understanding of the propagation simulation is required. The propagation is simulated in a grid which each cell having different fuel values based on the terrain textures that have been set, taking into account variables such as air temperature and ground moisture. Each cell is heated up to ignition temperature, then burns until there is no more fuel left in the cell then goes out. The **Wind Zone** is any Windzone in the scene that should be used by the simulation.

The higher the **Air Temperature** the quicker fuels heat up. The higher the **Ground Moisture** the slower fuels heat up. Terrain textures have a **Fuel** and **HP** value, the higher the HP value the longer it takes for fuels to reach ignition temperature. The fuel is the amount of fuel that the fire can burn and once the fire is out of fuel the fire dies out.

> *For example*, a hot dry day with very dry grass can be modelled by using a high air temperature, low ground moisture and low HP.

Higher the **Combustion Rate** the faster fires consume fuel, if you want a thin fire front this value could be high. For a large pool of fire, this value should be lower. **Node Combustion** Rate is the same but applied to Fire Nodes.

To provide more direct control over the simulation there are a number of bias variables that can be set:

- **Propagation Bias**: Affect how uniformed the propagation is, smaller values result in more random looking fires.
- **Propagation Hill Bias**: How much slopes in the terrain affect fire, larger values the faster fire propagates on slopes.
- **Propagation Wind Bias**: How much the wind affects the fire, larger the value the faster fire propagates with and into the wind.

To stop the fire from "jumping" up steep slopes or cliffs the maximum distance possible to be travelled between cells can be set with the **Max Hill Propagation Distance**. The **Visual Extinguish Threshold** is a value between 0 and 1, which defines the point that fires internal state should be set to visual extinguish state; more details in the *How to change the appearance of a fire* section.

The time of day is only taken into account by the “Detailed Simulation” which takes into account the air flows on slopes depending if it is **Day Time** or not. For night, untick the Day Time box.

Terrain Textures have additional local variable just for that type of fuel, such as **Moisture** and **Fire Propagation Speed**. The terrain texture moisture is added to the Ground Moisture. The fire propagation speed provides greater control over how fast the fire should spread, the higher the value the faster the fire propagates.

Burnt Grass Texture Index is the index of the grass texture that should be used when fire replaces a grass texture once it has been burnt by fire.

Detailed Simulation performs more physical factors that affect fires, such as, taking into account the time of day. This generally has very little impact on performance over the normal simulation but if your game only has one time of day then it recommended not to enable this. **Remove grass once burnt,** by default grass is replaced with a burnt grass texture, this removes it rather than replacing it. **Replace terrain texture once burnt** will affect performance, this will replace the current terrain texture with a burnt/scorch mark terrain texture. This setting is only recommended if targeting powerful hardware.

Terrain Update Time is the amount of time required before the terrain will be updated (i.e. scorch marks or removing grass).

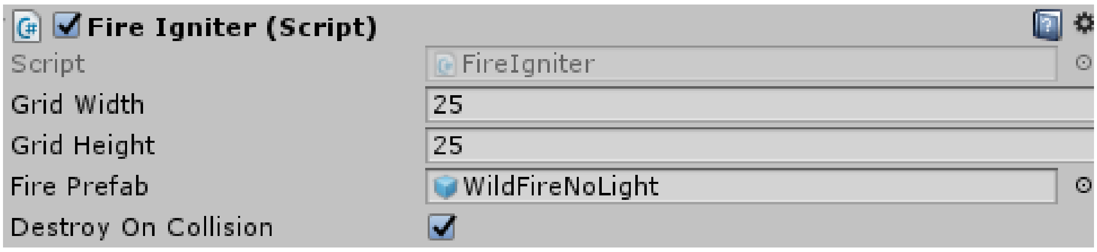
Staggered Grid Construction allows grids to be initialised over multiple frames rather than in a single frame. Enabling this is useful when using larger sized grids.

Max Grass Details allows for more than two grass texture details to be used by the simulation, performance and memory use will be impacted when more grass details are used on the terrain. **Burnt Grass Detail Index** is the index of the grass detail that will be used as the burnt grass to replace any grass detail once the grass has been burnt.

Many of the Fire Manager settings can be **modified by other scripts** in order to update fire behaviour during gameplay, such as changing the time of day or air temperature. These changes will affect all new fire's created after the changes. The Fire Manager presents many settings that allow for a wide range of different behaviour, you should experiment with the different settings to find the behaviour you are looking for.

Fire Igniter

If an object has a Fire Igniter script attached to it, whenever the object collides with the terrain it will spawn a fire. Individual fire settings can be set using this script, such as the size of the grid to create and what **Prefab** should be used for the fire, the Prefab needs to contain at least one **Particle System** component.



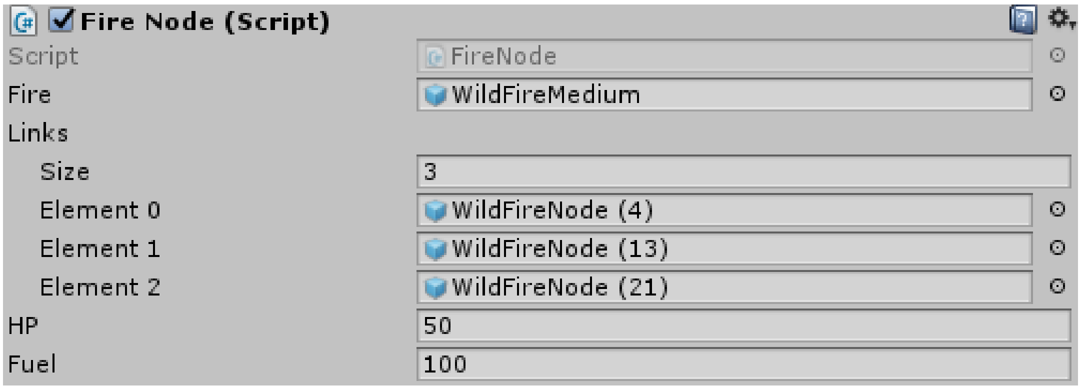
☑ Fire Igniter (Script)

Script	FireIgniter
Grid Width	25
Grid Height	25
Fire Prefab	WildFireNoLight
Destroy On Collision	☑

Fire Nodes and Chains

When fire is propagating on the terrain it will detect any fire chain within heating distance and will start to heat up the ignitable object the fire chain belongs to. The ignitable object can either be destroyed or replaced with another object once all of the fire nodes have been consumed by the fire. A Fire Node Chain is mostly a container for all for the Fire Nodes an object has and to set shared fuel values between all contained nodes.

Fire Node's allow for each node to have its own fuel value, which is useful for an object like a tree where the fuel from the base of the trunk to the top of the tree could change. They also have an array of “linked” nodes, fire will propagate an ignitable object using these links.

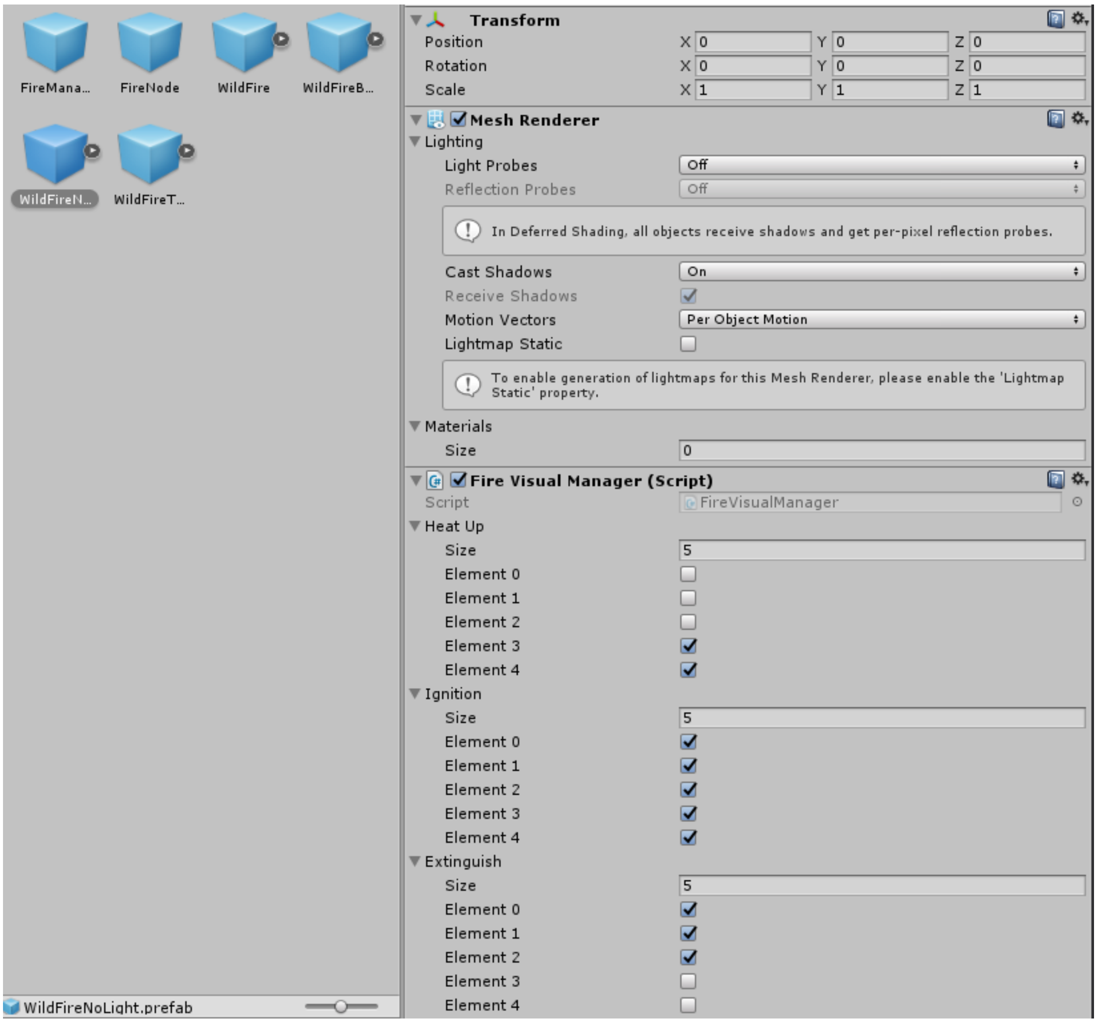


☑ Fire Node (Script)

Script	FireNode
Fire	WildFireMedium
Links	
Size	3
Element 0	WildFireNode (4)
Element 1	WildFireNode (13)
Element 2	WildFireNode (21)
HP	50
Fuel	100

How to change the appearance of a fire

The appearance if any fire is under your direct control, the only thing the fire propagation simulation does is provide the ability to toggle when particle systems contained within the Fire Prefab should be active or inactive. There are three internal states, **Heat Up**, **Ignition** and **Extinguish**. Set these arrays to match the number of particle systems in the Prefab, for each state tick the particle systems you want to be active. When a cell is being heated up the Heat Up state it active, when the ignition temperature has been reached the internal state is Ignition. The internal Extinguish state is set by the **Visual Extinguish Threshold** in the Fire Manager, the smaller the threshold value to longer the Ignition state will last. The threshold is percentage based on the about of fuel available to the fire.



Transform

Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

☑ **Mesh Renderer**

Lighting
- Light Probes: Off
- Reflection Probes: Off
- In Deferred Shading, all objects receive shadows and get per-pixel reflection probes.
- Cast Shadows: On
- Receive Shadows: ☑
- Motion Vectors: Per Object Motion
- Lightmap Static: □
- To enable generation of lightmaps for this Mesh Renderer, please enable the 'Lightmap Static' property.

Materials
- Size: 0

☑ **Fire Visual Manager (Script)**

Script: FireVisualManager

Heat Up
- Size: 5
- Element 0: □
- Element 1: □
- Element 2: □
- Element 3: ☑
- Element 4: ☑

Ignition
- Size: 5
- Element 0: ☑
- Element 1: ☑
- Element 2: ☑
- Element 3: ☑
- Element 4: ☑

Extinguish
- Size: 5
- Element 0: ☑
- Element 1: ☑
- Element 2: ☑
- Element 3: □
- Element 4: □

Advanced settings

Multiple fires can be spawned in each cell, using **Cell Fire Spawn Positions** in the Fire Manager. The positions are relative to the cell position based on the **Cell Size**. If a cell size is 1, to ensure fire is not spawned out of place make sure all relative spawn positions are between 0 and 1. This is very useful when using larger sized grids, as you can still have the same number of flames in the fire but using fewer cells which cover more ground. Using a larger Cell Size can improve performance but can affect the overall visual quality of the propagation.

Pre Allocated Fire Index Size allows for arrays that hold active cell information (per grid basis, Fire Nodes are excluded from these arrays) to be allocated to the defined size to prevent resizing of arrays. The total number of active cells will change depending on fuel values and grid size settings.