Fire Simulation System

Game Systems Course Project

Ruhil Nitin
Student ID: V01016220

1 System Overview

This project is a fire simulation system made in Unity using the Universal Render Pipeline (URP). The goal of this project is not only to show fire visually, but to model how fire behaves as a system.

The system controls how fire starts, burns over time, reacts to wind, runs out of fuel, and spreads to nearby objects. Unity's particle system and shaders are used only to display the fire. All fire behavior such as fuel usage, intensity, wind, and spreading is handled by custom scripts.

Because the logic and visuals are separated, the system can be reused for different situations such as a torch, a campfire, or a spreading environmental fire.

2 Features

The Fire Simulation System includes the following features:

- Fire that can ignite and extinguish correctly
- Fuel-based burning where fire dies when fuel runs out
- Fire intensity based on remaining fuel
- Wind that affects flame direction
- Fire spreading to nearby objects
- Modular prefab-based design
- Easy customization using Inspector values
- Visual reaction when objects catch fire
- Three demo scenes showing different use cases

3 System Design

The system is divided into two main parts: logic and visuals.

3.1 Main Scripts

- **FireSystem**
  Controls the fire state, fuel amount, intensity, and public interaction functions.
- **FireVisualController**
  Updates the particle system based on fire intensity and wind values.
- **FireSpread**
  Handles how fire spreads to nearby flammable objects.
- **Flammable**
  Marks objects that can catch fire and controls their visual reaction.

3.2 Design Principle

Fire behavior is fully controlled by custom logic. Visual effects only respond to values from the system. This makes the system clean, reusable, and easy to extend.

4 Code Structure

4.1 FireSystem.cs

Handles fuel usage, fire intensity calculation, ignition, and extinguishing.

4.2 FireVisualController.cs

Controls particle emission rate and particle speed based on fire intensity and wind.

4.3 FireSpread.cs

Detects nearby flammable objects and spreads fire using a chance-based method.

4.4 Flammable.cs

Allows objects to catch fire and change appearance when ignited.

5 Models and Calculations

5.1 Fuel Usage

Fuel decreases over time while the fire is burning:

$$fuel = fuel - (burnRate \times time)$$

5.2 Fire Intensity

Fire intensity is based on the remaining fuel:

$$intensity = \frac{fuel}{maxFuel}$$

5.3 Fire Spread

Nearby objects have a chance to catch fire each second:

$$spread\_chance = spreadRate \times time$$

This approach keeps the simulation simple and efficient.

6 How to Use the System

6.1 Public Functions

- `Ignite()` – starts the fire
- `Extinguish()` – stops the fire
- `AddFuel()` – adds fuel during runtime
- `SetWind()` – updates wind direction and strength

6.2 Inspector Controls

- Fuel amount and burn rate
- Wind direction and strength
- Particle intensity settings
- Fire spread radius and chance

These controls allow one fire prefab to be reused in many different ways.

7 Demo Scenes

7.1 TorchFire

This scene shows a single fire instance with fire spread disabled. It demonstrates customization and reuse of the fire system.

7.2 Campfire

This scene shows fire burning out naturally, reacting to wind, and being refueled during runtime.

7.3 SpreadingFire

This scene demonstrates fire spreading between nearby flammable objects with clear visual feedback.

8 Performance Considerations

Performance is maintained by:

- Limiting particle counts
- Using chance-based fire spread
- Restricting fire spread distance
- Separating logic from visual effects

This allows multiple fire instances to exist without major performance issues.

9 References

- Game Developer. *Fire, Blood, Explosions: Prototype 2 Effects Technology.* Available at: `https://www.gamedeveloper.com/programming/fire-blood-explosions-i-prototype-2-i-s-over-the-top-effects-tech`.
- GDC Vault. *Visual Effects Bootcamp: Rapid Talks.* Available at: `https://www.gdcvault.com/play/1024299/Visual-Effects-Bootcamp-Rapid`.
- GDC Vault. *The Dao of VFX Animation.* Available at: `https://www.gdcvault.com/play/1026048/The-Dao-of-VFX`.
- Unity Asset Store. *Camping Low Poly Pack Lite.* Available at: `https://assetstore.unity.com/packages/3d/environments/camping-low-poly-pack-lite-322826`.

10 Final Summary

This project implements a complete fire simulation system rather than a simple visual effect. Fire behavior is controlled by custom logic, while visuals only display the system state. The system is modular, reusable, and suitable for different game environments.