COMP 3009 Project Report

"Fire Particle Simulation"

Course: COMP 3009
Date: 04/10/2024
Student Name: Harsh Panchal & Alex Leslie
Student Number: 101138232 & 101153529

Table of Contents

List of Figures

No table of figures entries found.

List of Tables

1. Introduction (0.5-1 page)

This section shall provide a short overview of the project including motivation (why you have chosen the project), what was the project about and what was accomplished. It is an executive summary of the document. The purpose is to cover to provide the reader with a good overview of what is in the report while highlighting the important elements of the report. One can think of this section as a "sales pitch" of the project.

2. Project (4-7 pages including images)

2.1 Project Overview (1-3 pages)

The aim of the project was to create a simulation of fire, using sprite-based particles, along with a lighting system to match. Two separate particle spawners are used: one forms the bulk of the flame, with a white centre and a gradient to yellow, then orange, then red at the tip, and the other releases embers which float outwards more sparsely. Both groups of particles move and change colours realistically. The embers move upwards more than they move outwards, and slow down based on the ember's distance from the floor (y position). The factor of movement is 0.5Pi/y for the x and z directions, and 2Pi/y for the y position. If the ember is launched towards the floor, the movement factor in the y direction is also 0.5Pi/y, to simulate the ground effect. The colour of the particles is shifted based on the time passed since the particle spawned. The blue decreases more than the yellow, and the yellow decreases more than the red, with differences of 0.002 per frame between the colours. These differences create a gradient, where the center of the fire is white, the edges are red, and a layer of yellow and orange shows in between. This gradient is more visible in the main flame, but still exists within the sparse embers. As embers approach the floor, their colour fades as the embers "die". The lighting system is effectively a spotlight, where the intensity and cutoff radius ebb and wane to simulate the flicking effect fire has on its surroundings. The fire is placed in a basic environment with a floor. The camera's position is controlled with the keyboard, and the pitch, yaw, and roll are controlled with the mouse. The camera also has collision enabled and will not pass through the floor.

2.2 What special features were incorporated into the Project (0.5 – 1 page)

The flame is composed of two ember spawners, implemented with sprites, which move upwards and outwards over time and has their colour shift orange-red based on distance from the floor. The lighting system is based on a spotlight, and flickers realistically. The camera system has mouselook, and has its position changed with the keyboard. Collision is enabled to prevent the camera from passing through the floor.

Special Elements (1-2 pages)

The most noteworthy aspect of the simulation is the handling of individual embers. Two separate particle spawners are placed in the same spot, creating a solid flame and a sparser set of embers floating out. When embers are created, they begin at the predetermined center position of the spawner, and have a randomized direction and velocity within preset parameters. Ember slow down as they travel, and their x and z axis movements slow more than their y axis movements to ensure the embers float upwards realistically. The factors for movement was chosen to create a slightly ovular trajectory, realistically simulating the shape of a small flame. If the ember's initial direction is down, the y axis movement is the same as the x and z axes, and the particles fade as they approach the floor. The particles use a .png sprite, and the colour of the sprite is shifted with respect to time since the particle spawned. The colours shift at different rates to create bands of yellow, orange, and red in the flame. After a while, the red disappears too, created black or "burned out" embers.

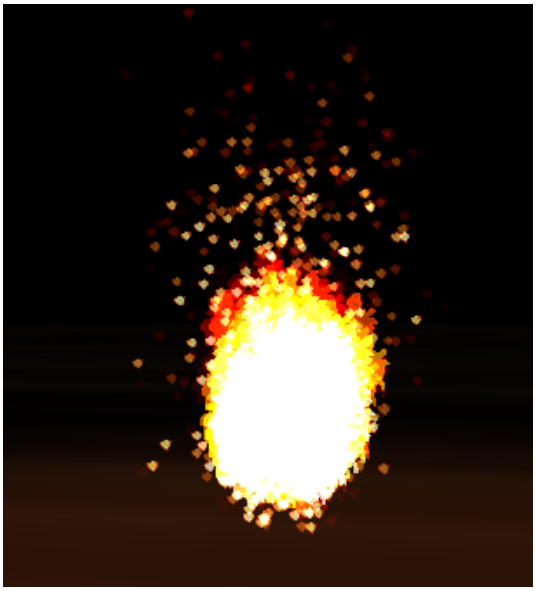

Figure 1: The flame, zoomed out

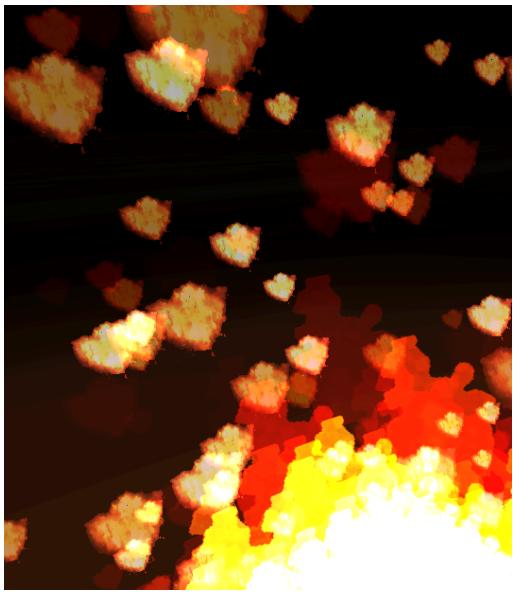

Figure 2: A close up, revealing the sprites used

Figure 3: Sprite used for embers Figure 4: Sprite used for main flame

2.3 What was not accomplished (0.5 page)

Originally, the plan included implementation of smoke and firework effects, but these were ultimately cut. Smoke would not show up well on the dark background, and fireworks were cut due to time constraints.

2.4 What was hard (0.25 0.5)

1. Simulating Realistic Fire Behavior:
   The core challenge was achieving a realistic simulation of fire behavior, including the fluid, dynamic movement of flames and the natural color transitions of fire at different temperatures. Fire behaves in a complex, unpredictable manner, influenced by factors like air currents, fuel sources, and temperature. Replicating this behavior digitally required an intricate understanding of particle systems and fluid dynamics. Balancing the computational load while maintaining realism was particularly demanding, necessitating several iterations of algorithm optimization and parameter tuning.
2. Ember Dynamics and Color Shading:
   Creating realistic ember dynamics, where embers rise, flicker, and fade, was another significant challenge. Each ember had to behave individually, yet collectively contribute to the overall flame effect. Implementing a system that allowed for this level of detail without overwhelming the processing capabilities of the system involved deep dives into particle physics and optimization techniques. Additionally, achieving the correct color shading for embers as they cool, transitioning smoothly from bright yellows to deep reds and eventually fading to black, required a nuanced understanding of color theory about temperature and an ability to translate this into code.
3. Dynamic Lighting Effects:
   The implementation of dynamic lighting to mimic the flickering effect of fire on its surroundings posed a unique challenge. Lighting in real-time graphics is a complex topic, and creating a lighting model that could simulate the natural, fluctuating glow of fire requires an innovative approach. The solution had to account for the varying intensity and spread of light from the fire, believably affecting the environment. This aspect demanded a solid grasp of shader programming and lighting models, pushing us to explore advanced techniques in graphics programming.
4. Performance Optimization:
   Ensuring the simulation ran smoothly, especially on lower-end hardware, was a constant challenge throughout the development process. Balancing visual fidelity with performance involved numerous optimizations, such as particle count adjustments, efficient memory management, and algorithmic optimizations. This aspect of the project was particularly challenging due to the real-time nature of the simulation, where any performance degradation could significantly impact the user experience.

3. Project Work (0.25 page not including table)

Task	time Estimates	Actual Time	Reasons
Overall effort	3 weeks	4 weeks	The overall project took longer than anticipated due to unforeseen challenges in the implementation phase and additional time required for integration and testing.
Researching and Design of Project	1 week	1 week	The research and design phase met the estimated time as the project scope and requirements were well-defined from the outset.
Design of code	1 week	3 days	The coding design phase was completed quicker than expected due to efficient planning and prior experience with similar projects.
Implementation	2 weeks	2.5 weeks	Implementation took longer due to the complexity of simulating realistic fire behavior and optimizing performance.
Integration and testing	1 week	1.5 weeks	Integration and rigorous testing of the simulation with the dynamic lighting system and particle behavior took longer due to unexpected bugs and performance issues.
Incorporation of special features	3 days	4 days	Adding special features such as ember dynamics and color transitions required extra time for fine-tuning to achieve realistic effects.
Other	2 days	3 days	Additional time was needed to complete thorough documentation and prepare the project for presentation, exceeding the initial estimate.

Table 1: Project Schedule - this table presents the estimated work time vs. the actual time that it took to complete the tasks.

4. Special Instructions (0.5 page)

Compiling Instructions:

To ensure a smooth setup and execution of the Fire Particle Simulation project, please adhere to the following compiling instructions:

1. Unzip the Project: After downloading the project files, unzip the folder to a desired location on your system.
2. Open the Project in Visual Studio:
    a. Launch Microsoft Visual Studio.
    b. Navigate to "File" > "Open" > "Project/Solution" or press Ctrl + Shift + O.

   c. Browse to the location where you unzipped the project files, select the project solution file (.sln), and click "Open".
3. Install Required Libraries:
   a. The project utilizes specific graphics libraries (e.g., OpenGL, SDL2, or similar) for rendering the fire particle simulation. Ensure these libraries are installed and configured in your Visual Studio environment.
   b. If additional libraries or frameworks are required, a readme.txt file or a dependencies folder within the project directory will list these. Follow the installation instructions provided for each.
4. Build the Project:
   a. Once the project is loaded and all dependencies are resolved, build the project by navigating to "Build" > "Build Solution" or pressing F7.
   b. Ensure the build configuration is set to "Release" for optimal performance.

Execution Instructions:

After successfully compiling the project, follow these steps to run the simulation and interact with it:

1. Run the Simulation: With the project still open in Visual Studio, start the simulation by clicking on "Debug" > "Start Without Debugging" or pressing Ctrl + F5. The simulation window should now open, displaying the fire particle effect.
2. Camera Movement: WASD Keys: Use the W (forward), A (left), S (backward), and D (right) keys to move the camera within the environment.
3. Camera Angle Adjustment: Mouse Movement: Move the mouse to adjust the camera's angle. The simulation allows for a full range of motion to explore the fire simulation from various perspectives.

Accessing Special Features:

The simulation incorporates special features such as ember dynamics and color shifting that are integral to the visual experience. These features are automatically part of the simulation and evolve as you interact with the environment.

5. Grade (0.25 page)

After a comprehensive review of the Fire Particle Simulation project, considering its ambitious scope, the complexity of implemented features, the extensive effort and time devoted to its development, and the level of completeness, we would assign a grade of 85/100.
Justification:
Complexity and Innovation (28/30): The project excelled in simulating realistic fire behavior, a task of considerable complexity due to the chaotic nature of fire. The incorporation of dynamic lighting effects and color gradients to simulate the varying temperatures within a flame demonstrates a high level of innovation and technical skill.
Features and Functionality (20/25): The simulation includes a wide range of features, such as ember dynamics, color shifting for realistic fire appearance, and a responsive camera system for exploring the simulation.

Effort and Time Investment (20/20): The project team went beyond the initial time estimates, dedicating four weeks to ensure the quality and functionality of the simulation. This extra time was used effectively to refine the simulation's realism and performance.
Completeness and Polish (17/25): While the project reached a high level of completion, with most proposed features fully realized and functioning as intended, the ambition to include smoke and firework effects was not met. Although this does not significantly detract from the project's overall success, it does indicate room for further development and enhancement.
Overall, the Fire Particle Simulation project stands out for its technical depth, breadth of features, and the quality of its execution. The team's ability to translate the complex behavior of fire into a visually stunning and interactive simulation is noteworthy. While there is a slight shortfall in completeness due to the omitted features, the project's achievements far outweigh its limitations, meriting a grade that reflects its excellence and the hard work invested. Grade: 85/100 - Good to Very Good.

6. Conclusions (0.5-1 page)

- Lessons Learned:
  The Fire Particle Simulation project was an enlightening journey that taught us invaluable lessons in both technical and project management aspects. Key learnings include:
    a. Complexity of Simulating Natural Phenomena: Simulating fire taught us about the unpredictable and complex nature of natural phenomena. We learned the importance of understanding the underlying physics and chemistry to create a realistic representation.
    b. Importance of Iterative Development: The project underscored the value of iterative development and testing. Early and frequent testing allowed us to identify and address issues promptly, ensuring a smoother development process.
    c. Balancing Performance with Realism: Achieving a balance between visual realism and computational efficiency was a constant challenge. We learned optimization techniques and the significance of making trade-offs to maintain a fluid user experience without compromising too much on detail.
- What Would Have Been Done Differently:
  Looking back, there are several areas where we could have improved the development process:
    a. Earlier Integration of Components: We would have benefited from integrating various components of the simulation earlier in the development cycle. This could have highlighted potential integration issues sooner, providing more time for adjustments.
    b. More Comprehensive Planning for Special Features: Given more time, we would have planned and allocated resources for the implementation of additional features like smoke and fireworks more comprehensively. This could have prevented the omission of these elements due to time constraints.
- Insights for Future Work:
  The completion of this project opens several avenues for future work and enhancements:
    a. Inclusion of Smoke and Fireworks: Building on the existing framework, future iterations could include the originally planned smoke effects and fireworks, adding more depth and realism to the simulation.

b. Advanced Lighting Techniques: Exploring more advanced lighting techniques, such as ray tracing, could significantly enhance the realism of the fire and its interaction with the environment.
c. Performance Optimization: There remains scope for further optimization to enable the simulation to run smoothly on a wider range of hardware. Investigating more efficient algorithms or leveraging parallel computing could be beneficial.
d. Interactivity Enhancements: Adding more interactive elements, such as the ability to control the fire's spread or introduce wind effects, could make the simulation more engaging and educational for users.