Project Firefly

Haoyang Zhang, Yinfeng Cong

May 4, 2021

Group members and division of workload

Name NetID	Workload
Haoyang Zhang hz333	Implemented firefly movement, glowing, camera, and menu. Write the report in LaTeX.
Yinfeng Cong yc957	Modeled fireflies. Tested the application.

1 Description

Fireflies float in the midair, lighting up the moonless night. In this project, we are going to use `WebGL` to simulate fireflies glowing on the riverbank, and their self-synchronizing effect.

This project provides (Fig. 1):

- A 3D scene that contains fireflies floating in the midair.
- Glowing of fireflies that begins with chaos but will synchronize after a certain time.
- An omniscient third-person view to appreciate the beauty of self-synchronizing.
- A first-person view to enjoy fireflies surrounding yourself.
- A menu to configure the simulator, including the number of fireflies, glowing settings, self-synchronizing settings, and camera settings.

2 Implementation

The implementation can be considered as three parts: initialization, interaction, and animation. The animation part contains two independent cycles: the rendering cycle and the updating cycle (Fig. 2).

In initialization, we initialize the environment, and construct 3D models of a single firefly and a single glowing sphere, sending them to GPU. In animation, we use updating cycle to update uniform variables, including the transposes of fireflies and the glowing clocks of glowing. The other cycle, the rendering cycle, sends these variables to GPU and draws multiple fireflies with glowing. The interaction part takes care of the user's inputs and processes them for the animation part. We use two independent cycles in the animation part to make sure this simulation runs at a constant speed no matter how large the screen refresh rate is. Or this could be a nightmare where some people see fireflies run over-fast like rockets, while others find them almost steady.

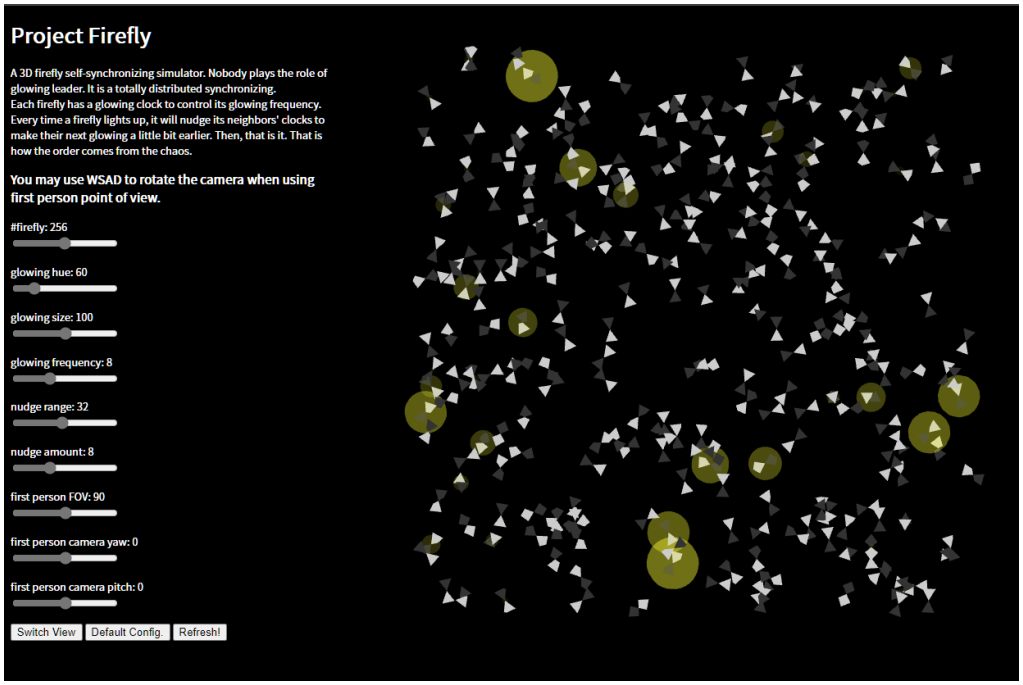

Figure 1: application interface

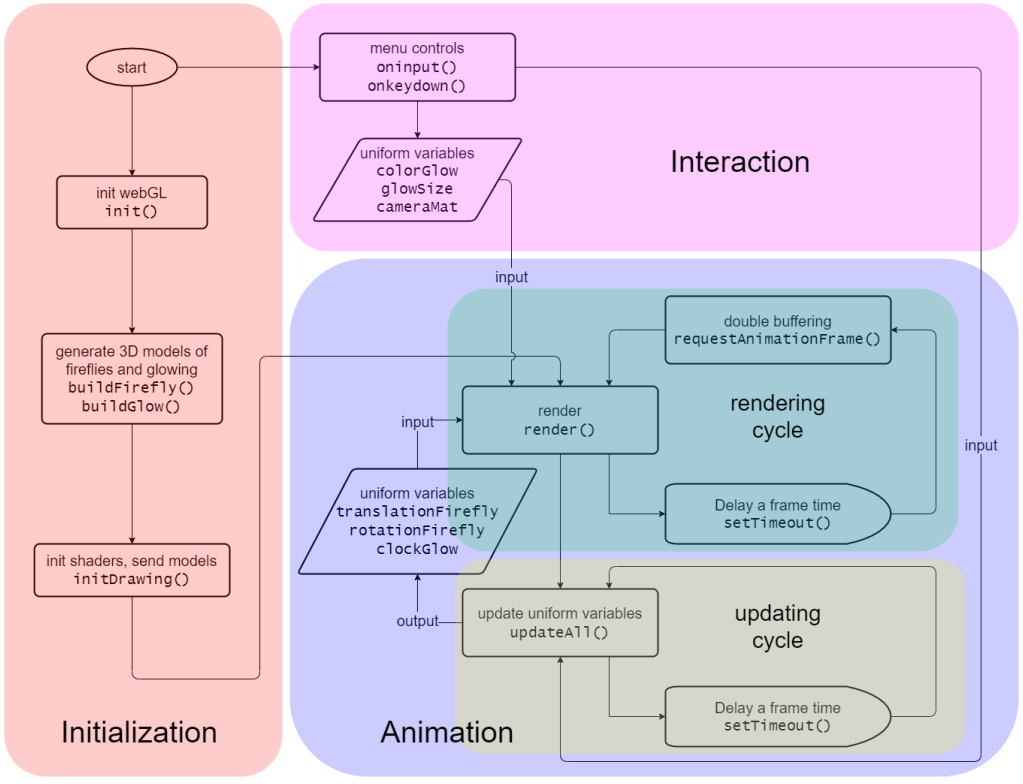

Figure 2: Implementation conceptual flow chart

2.1 Modeling fireflies and their movements

A firefly is described by two tetrahedrons sharing a common vertex (Fig. 3), where the light grey tetrahedron is its head, and the deep grey one is the tail.

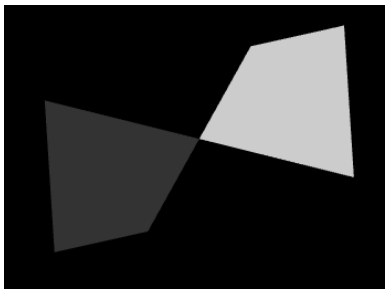

Figure 3: firefly model

The movement of fireflies can be described by their direction and length. Given the initial position and heading direction of each firefly, at each updating step, we randomly adjust its yaw, pitch, and row angles slightly, and then move it by a unit distance. If this movement causes flying out of boundary or colliding with others, we will discard this movement. Instead, we turn this firefly a lot more to expect the next updating will not lead to an invalid movement.

To initialize positions and directions of all fireflies, we first randomly sample points in the space satisfying the minimum pair-wise distances to avoid collisions, which are used as positions. The collision checking here is stricter to leave enough space for fireflies to fly. Then we assign a random direction for each firefly, by randomly picking yaw, pitch, roll angles.

When users change the number of fireflies, we pop or append new randomly initialized fireflies accordingly. Therefore users may use this feature to reset the simulation to the chaos state.

2.2 Modeling glowing

The glowing is described by a single sphere. Each firefly has two related clocks: `clockGlow` $\in [0, 1]$ for the glowing phase, and `clockFirefly` $\in [0, 1]$ for the non-glowing phase, which connect to each other end to end. The glowing sphere size and alpha value are proportional to $(1 - \texttt{clockGlow})^2$. The color is controlled by users through the menu.

The secret of the self-synchronizing described in Sarfati et al. (2020) is: once a firefly lights up, its neighbors (within a certain distance) that in `clockFirefly` phase will nudge the clock a little bit to spark earlier. We provide detailed control of this nudging effect for users to explore its result.

2.3 Modeling Camera

The omniscient third-person view can be easily defined by the orthographic projection of $[-1, +1]^3$. For the first-person view, we fixed the camera at $(0, 0, 0)$, and use the perspective projection of `near = 0.1, far = 1`. The field of view is controlled by users through the menu. We also enable the user to rotate the camera like turning their head by changing the yaw and pitch angles of the camera.

3 Timeline

The Gantt chart is Fig. 4. For the time being, we have not complete one of the ultimate goals: using mouse clicking to interrupt the synchronizing. However, as described above, users may use the slider for the number of fireflies to reset parts of the fireflies' clocks.

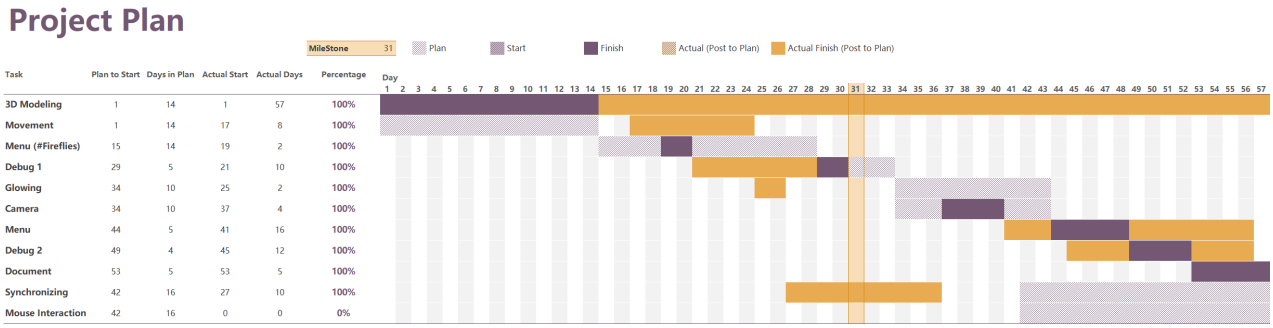

Figure 4: Gantt chart

4 Conclusion and future work

This project aims to show simulate the scene of fireflies and their self-synchronizing. Users can use the first-person view to dive into the scene and configure the parameters of glowing to explore the effect of self-synchronizing. To get a better user experience, we are expected for better modeling of fireflies, and an environment background like rivers under the night sky. An array of cameras may make it easier to understand the scene, which may lead to a VR version of this application. It will take about two minutes to synchronize using the current default configuration, which is usually good for presentation. However, the current configuration is complex for non-expert users. They may find it difficult to find a good set of parameters so that the synchronization will happen in their expected time. A set of preset configurations may help to address this issue.

References

Raphaël Sarfati, Julie C Hayes, Élie Sarfati, and Orit Peleg. 2020. Spatio-temporal reconstruction of emergent flash synchronization in firefly swarms via stereoscopic 360-degree cameras. *Journal of the Royal Society Interface*, 17(170):20200179.