ESAT ESCUELA SUPERIOR DE ARTE Y TECNOLOGÍA

Programación gráfica

Memoria

> Alumno: Carlos García Roig
> Profesor: Iván Sancho

> Asignatura: Programación gráfica
> Curso: 2023-2024

Index

1 Porstmortem

1.1 Fortalezas del proceso creativo

Entre las fortalezas del proceso creativo se encuentran las siguientes:

- **Perseverancia**: Cuando una tarea se atasca por diversos motivos, ya sea la dificultad o por algún error que se es capaz de detectar, la perseverancia juega un papel crucial para no desistir y, con calma, buscar la forma de solucionar la situación.
- **Iteración y Feedback**: Contar con el apoyo del profesorado para que te vayan validando el trabajo conforme se va haciendo es algo que ayuda mucho en el proceso de desarrollo. Así como un Feedback continuo que te ayude a detectar mejor posibles errores que no se estaban detectando.

1.2 Dificultades

La principal dificultad que he tenido en este proyecto ha sido la de trabajar con sobre un software desarrollado por terceros, en este caso la escuela. Por motivos académicos no se nos brinda acceso al código fuente. Tampoco contamos con documentación por parte de la escuela. Lo cual impide saber con exactitud lo que hacen las funciones por debajo, dejando como único método posible, en algunos casos, la prueba y error para saber cómo avanzar con algunas cosas. Pese a todo esto, el profesor ha explicado todas las dudas relacionadas con el motor de la escuela cuando se le planteaban en clase.

1.3 ¿Que se haría distinto?

No estoy del todo contento con el esquema interno que he diseñado respecto a las entidades y las geometrías, sobre todo, respecto a las geometrías. El escalado de los drawables no queda bien, provoca que la geometría empiece a tener manchas negras, lo cual se puede solucionar recalculando los puntos de la geometría asociada a cada entidad. El problema de hacer eso ahora mismo es que se modificarían todas las entidades asociadas a esa geometría, por lo tanto, generar instancias de geometrías en el momento de crear una entidad sería algo a considerar. Por otro lado, considero que no hago un buen uso de los materiales porque en muchas ocasiones podría reutilizar la misma settings para diferentes usos, simplemente actualizando la textura que se muestra en cada momento. La jerarquía de ventanas de ImGui no ha quedado como a mi me gustaría y la cambiaría totalmente, así como las funciones que se encargan de gestionar las ventanas, lo cual viene heredado del proyecto de la asignatura de programación avanzada.

1.4 ¿Qué problemas han surgido y como se han solucionado?

Se han enfrentado multitud de problemas durante el desarrollo del proyecto. El principal de ellos ha sido el lidiar con **OpenGL** dentro de un entorno que no permite hacer debug con facilidad. El motor de la escuela actúa, en muchas ocasiones, como un **wrapper** de **OpenGL** e intentar detectar los errores puede ser muy tedioso. La solución que se ha seguido para esto ha sido la de implementar las nuevas funcionalidades poco a poco, sin incluir muchos cambios de

golpe. También se ha hecho uso de la función **assert** para comprobar si OpenGL había tenido algún error.

También se han tenido muchos problemas a la hora de implementar el **custom gpu manager**. Concretamente a la hora de programar la parte de las texturas. En este caso, tras muchos intentos, se necesitó la ayuda del profesor para poder lidiar con los errores que se tenían.

1.5 ¿Qué ha salido bien y que ha salido mal?

Creo que el proyecto en general ha tenido un resultado muy satisfactorio. Casi todos los retos planteados se han conseguido implementar correctamente. El uso de un **heightmap** a la hora de generar el terreno ha permitido tener una geometría mas complejo y mas detallada que, junto con el uso de la función de ruido, generaba una geometría de una calidad mas que aceptable. Quiero destacar también la creación de varias superficies de revolución propias para generar islas o árboles. En el caso de las islas, se ha combinado junto con la función de ruido para añadir aún mas variedad. Sin embargo, la parte que mas me ha gustado implementar ha sido la de las partículas. Por falta de tiempo no he podido incluir mas mejoras pero el resultado creo que ha sido muy bueno.

Considero que nada en el proyecto ha salido “mal” porque aquellos aspectos que no se podían abordar o se complicaron mucho se enfocaron de otra forma y se encontró una solución valida yendo por otro camino.

1.6 ¿Qué mejoras se implementarían con más tiempo?

Siguiendo un poco la línea de los anteriores puntos. La principal mejora que me hubiese gustado implementar es la de poder generar una instancia de la geometría seleccionada para cada entidad. Esto hubiese permitido escalar la geometría de cada entidad sin afectar al resto. Por otro lado, me hubiese gustado poder implementar un sistema de partículas mas complejo, tomando como referencia el sistema de partículas de Unity, incluyendo diferentes formas de emisión, adición de diversas formas de renderizado de las partículas, tamaño durante el tiempo de vida, etc… Por último, me hubiese gustado poder incluir un método para guardar y cargar escenas.

2 Análisis de las APIs

La APIs de este proyecto, sobre todo, la parte de la gestión de entidades, se diseñaron con el objetivo de ser bastante escalables y versátiles de cara a posibles futuros cambios. Creo que se ha logrado en gran medida este objetivo ya que las entidades actúan únicamente como contenedores de información para los elementos que se van a dibujar en la escena y no resulta engorroso ni excesivamente complicado el añadir nuevas funcionalidades. Quiero destacar también la integración con la API de animaciones procedurales, la cual se implementó sin problemas.

El uso de una librería matemática también ha sido de mucha ayuda ya que no solo incorpora estructuras útiles como los vectores o las matrices, sino que también añade toda una serie de métodos como el producto escalar o el producto cruz, importantes a la hora de realizar ciertos cálculos como, por ejemplo, la orientación del **Billboarding** del sistema de partículas. Siguiendo esta línea, considero que el sistema de partículas implementado es bastante eficiente ya que solo se ejecuta un **DrawCall** por ciclo, independientemente de que haya 1,10,100 o 500 partículas en pantalla. Evitando así que un exceso de partículas en pantalla lastre el rendimiento de la escena. Además, está preparado para incorporar nuevas funcionalidades gracias a uso de una estructura de configuración que se aplica a cada sistema de partículas.