Universitat Politècnica de Catalunya

Facultat d'Informàtica de Barcelona (FIB)

Trabajo de Fin de Grado

Jordi Gil González

Análisis del método de renderizado Path Tracing en CPU y GPU

Entrega 2: Planificación temporal

Departament de Ciències de la Computació

Director del Trabajo de Fin de Grado:	Chica Calaf, Antoni
Tutor de GEP:	Barrabes Naval, Fernando
Programa de estudio:	Computación
Especialización:	Computación Gráfica

Curso Académico 2019/2020

30 de septiembre de 2019

Índice general

*

CAPÍTULO 1

PLANIFICACIÓN TEMPORAL

1.1. Planificación y programación

Este capítulo trata la planificación del proyecto y describe las tareas realizadas, a través de un plan de acción, para cumplir los plazos relativos a la entrega del proyecto. Veremos también los recursos utilizados y se tendrán en cuenta los posibles obstáculos que puedan modificar la planificación.

El proyecto se inició a principios de julio de 2019 con vista a entregarlo el 13 de enero de 2020.

1.2. Descripción de tareas y recursos utilizados

1.2.1. Descripción de tareas

Estudio de conceptos

Antes de comenzar el proyecto, fue necesario familiarizarse con los conceptos básicos que rigen nuestro proyecto. Durante el semestre previo hubo un par de reuniones con el director del proyecto para definir el tema en el cual se centraría y poder así encaminarlo. También, el autor del presente proyecto se matriculó de la asignatura de Tarjetas Gráficas y Aceleradores (TGA por sus siglas) para introducirse en la programación en el entorno de `CUDA` y así agilizar el proceso de desarrollo del proyecto su etapa inicial. Dado que esta tarea se realiza fuera del proyecto sumado a la dificultad de determinar una estimación en horas, no se tendrá en cuenta en la planificación.

Configuración del sistema

Antes de entrar propiamente en el desarrollo en si del proyecto, debemos configurar las herramientas necesarias y garantizar su correcto funcionamiento.

Para poder desarrollar el proyecto será necesario tener instalado en nuestros sistemas la API de `CUDA` con tal de poder crear el programa principal de nuestra aplicación paralela en GPU. Las librerías de `OpenMP` vienen instaladas por defecto en los sistemas Linux, por lo que no será necesaria una instalación, pero sí comprobar que funcionan correctamente. Por último, haremos uso de una plataforma, *TexMaker*, que nos permita compilar código LaTeXcon tal de generar la documentación requerida.

Esta configuración debe hacerse tanto en el computador de sobremesa como en el portátil. La parte de *CUDA* en el cluster de docencia no será necesario de configurar debido a que ya está previamente configurado por el DAC (Departament d'Arquitectura de Computadors).

Planificación del proyecto

Esta es la tarea que se está desarrollando actualmente. Esencialmente trata sobre todo el contenido cubierto por el curso de GEP. La podemos dividir en tres subtareas:

1. Contextualización y alcance.
2. Planificación temporal.
3. Presupuesto y sostenibilidad.

Desarrollo

Esta tarea es la más importante de todo el proyecto. Cubre el desarrollo de las aplicaciones, experimentación y documentación de los resultados de cada una de las partes. Dividiremos el desarrollo en cuatro ciclos de 14 días cada uno, coincidiendo con las reuniones programadas con el director del presente proyecto. Cada uno de los ciclos los dividiremos en cuatro tareas:

- **Desarrollo versión secuencial:** Se trata de la versión más básica de todas. Sirve como base para desarrollar tanto la versión paralela en CPU (`OpenMP`), como la versión paralela en GPU (`CUDA`).
- **Desarrollo versión paralela CPU:** Una vez implementada la versión secuencial pasaremos a implementar una versión paralela de ésta haciendo uso de la librería de `OpenMP`.
- **Desarrollo versión paralela GPU:** Una vez implementada la versión secuencia pasaremos a implementar una versión paralela de ésta haciendo uso de `CUDA`.
- **Experimentación:** Crearemos la misma escena en cada una de las versiones comentadas en los puntos anteriores y analizaremos el rendimiento de ambas, teniendo en cuenta el tiempo de creación requerido para generar la imagen final.

Las dependencias entre las diferentes tareas son fáciles de describir. Hasta que no tengamos la versión secuencial no podremos empezar a programar las versiones paralelas de ésta. Tampoco podremos empezar un ciclo nuevo sin haber terminado el anterior.

Etapa final

En esta etapa estructuraremos toda la documentación de las etapas anteriores y preparemos la presentación final para la defensa del proyecto.

1.2.2. Tabla resumen y Estimación

Tarea	Tiempo empleado (horas)
Configuración del sistema	10
Planificación del proyecto	130
Desarrollo	330
Etapa final	120
Total	590

Cuadro 1.1: Resumen de horas

Con una estimación de 35 horas a la semana (3 horas de lunes a jueves, 10 horas sábados y domingos), y teniendo presentes un total de 18 semanas de trabajo, tenemos: $35h \cdot 18 = 630$ horas en total. Esto nos deja un margen de tiempo para posibles inconvenientes.

1.2.3. Recursos utilizados

Software

- Linux, usado en todas las tareas.
- LaTeX, para realizar la documentación.
- `C++`,`OpenMP` y `CUDA`, para el desarrollo de las aplicaciones.
- `git`, utilizado para el control de versiones y compartir el código entre diferentes entornos (computador de sobremesa y el portátil).

Hardware

- PC Sobremesa con las siguientes características: i7 7700 3.6 GHz, NVIDIA GeForce RTX 2080 SUPER, 24GB RAM, 250 SSD, 1TB SSD, 1TB HDD
- PC Portátil con las siguientes características: i7 7700HQ 2.8GHz, NVIDIA GeForce 1050 Mobile, 8GB RAM, 500GB SSD
- Cluster de docencia con las siguientes características: Intel Xeon E5-2620 v2 2.10GHz x2, NVIDIA Tesla K40c x4, 64GB RAM, 1TB x2

1.3. Diagrama de Gannt

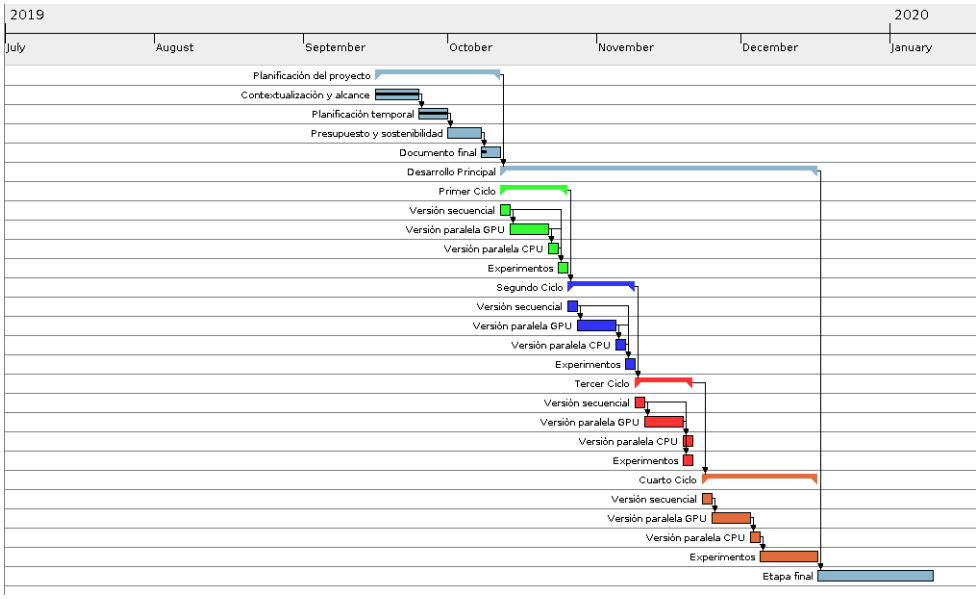

Figura 1.1: Diagrama de Gannt completo. Fuente: Propia

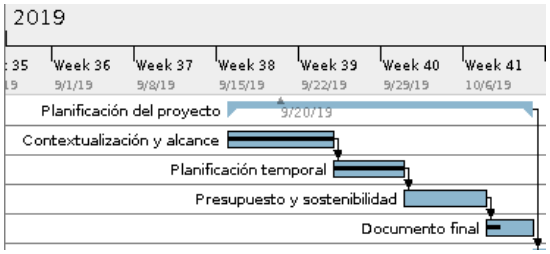

Figura 1.2: Etapa de gestión.
Fuente: Propia

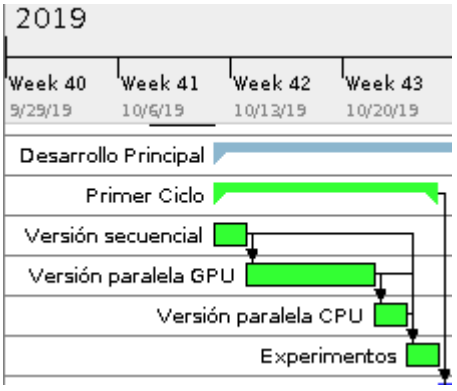

Figura 1.3: Primer ciclo.
Fuente: Propia

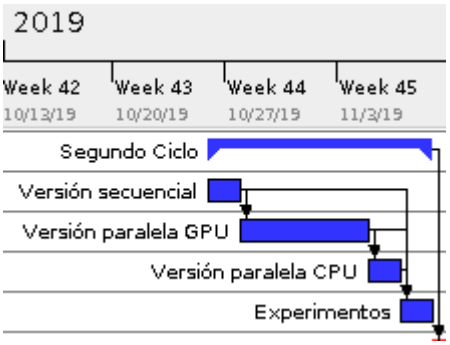

Figura 1.4: Segundo Ciclo. Fuente: Propia

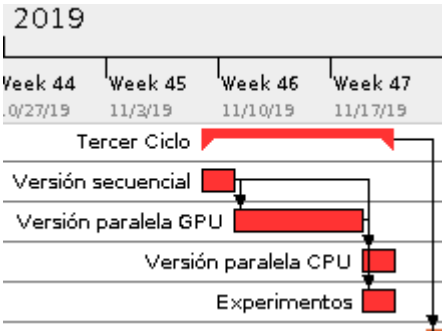

Figura 1.5: Tercer ciclo.
Fuente: Propia

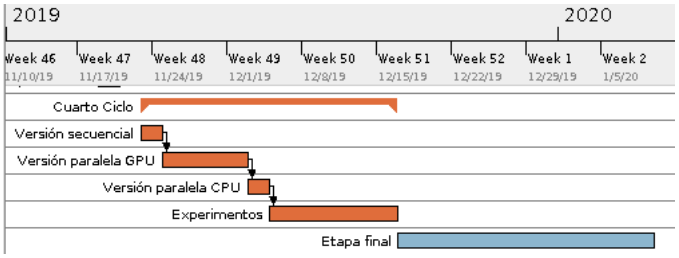

Figura 1.6: Cuarto ciclo y etapa final.
Fuente: Propia

1.4. Gestión de riesgos: Planes alternativos y obstáculos

Como podemos observar en la Figura 1.6, en el cuarto ciclo de desarrollo la tarea de experimentos ocupa más tiempo en comparación a ciclos anteriores. En este caso, hemos sobrestimado la duración de ésta para tener tiempo suficiente de resolver problemas que puedan aparecer, como los definidos en el capítulo anterior (Contextualización y alcance), en el curso del desarrollo del proyecto o experimentos que puedan surgir a última hora.