Taller de Programación I (75.42)
Facultad de Ingeniería
Universidad de Buenos Aires

WORMS

TP Grupal Final

Manual de proyecto

2º Cuatrimestre 2023

Primera Entrega: 21/11/2023

Segunda Entrega: 5/12/2023

Integrantes	Padrón
Facundo Huttin	107854
Theo Lijs	109472
Ivan Erlich	105989

Distribución del Proyecto

Distribuimos el proyecto en 3 partes principales: Logical del juego (Servidor) , concurrencia y esquema de hilos (Protocolo y Threads) y la parte del cliente de renderización (Cliente).

Servidor:

La lógica del juego fue realizada por Theo, quien se encargó de todo lo relacionado con la física (Box2D) y las clases que iban a representar las diferentes partes del juego. Modeló el juego para ajustar Box2D a la forma en que se supone que debe ser utilizado.

Protocolo:

La concurrencia y la estructura del proyecto fueron realizadas por Iván. Su tarea fue crear el esqueleto que sostendrá el proyecto y gestionar la comunicación entre el servidor y el cliente. Se encargó de evitar cualquier race condition y otros problemas relacionados con los recursos utilizados en el proyecto.

Cliente:

La renderización del juego fue una tarea de la cual Facundo fue responsable. Logró utilizar tanto las bibliotecas QT como SDL (SDL2) para ofrecer una vista agradable y una sensación nítida del juego al renderizar adecuadamente los sprites según el framerate se lo permitía.

Aunque cada uno de nosotros tenía roles diferentes, todos contribuyeron en cada parte del proyecto. En más de una ocasión, nos ayudamos mutuamente para descubrir cómo implementar algunas partes del juego y/o para ayudar en la depuración del código. La cooperación fue clave cuando el tiempo estaba en nuestra contra.

Organización

Para organizar adecuadamente las tareas pendientes del proyecto, utilizamos una aplicación llamada "Trello" que permite crear tarjetas en una columna de "TODO" y moverlas a otras columnas como "Lista de Bugs", "Hecho", etc. Esto nos ayudó mucho a hacer un seguimiento de qué características quedaban, cuáles estaban con errores y cuáles estaban completadas.

Al principio, enfrentamos algunas dificultades y el tiempo pasó demasiado rápido, dejándonos con menos tiempo del que hubiéramos deseado. Esto, o dicho de otra manera, el tiempo fue nuestro principal obstáculo. Además del tiempo, otro factor que nos complicó muchísimo al principio del desarrollo del proyecto fue la comunicación en el equipo. No nos propusimos juntarnos la cantidad de horas necesarias al principio para charlar y ponernos de acuerdo en cómo se iba a desarrollar cada etapa y etc.

Por ende, no se tuvo un seguimiento de semana a semana tan preciso como se hubiera querido. Las primeras dos semanas no pudimos ni levantar una demo competente en que dos clientes se muevan en la pantalla del juego. Por lo que en las próximas semanas estuvimos trabajando muchísimas horas por día tratando de llegar con todas las features posibles. (Se puede notar muchisima la diferencia en los commits del repo)

El volumen del proyecto y el tiempo fueron las razones por las cuales el proyecto fue desafiante. Aunque tuvimos momentos difíciles, logramos completar todas las características que se nos asignaron a tiempo de la mejor manera que se nos ocurrió dado el tiempo que nos quedaba.

Recursos que utilizamos

Los tres usamos Visual Studio Code como nuestro principal Entorno de Desarrollo Integrado (IDE). Además, utilizamos clang format, cpplint y cppcheck para mantener el código lo más limpio posible. Theo y Facundo utilizaron la terminal para ejecutar comandos git, mientras que Iván utilizó GitHub Desktop en lugar de la terminal.

Para aprender todas las tecnologías requeridas para el proyecto:

- Theo hizo muchos de los tutoriales encontrados en https://www.iforce2d.net/b2dtut/ y leyo la documentacion official https://box2d.org/documentation/ . También consiguió un poco de iluminación en el discord oficial de box2D acerca de raycasting.
- Ivan usó más que nada la información de los handouts otorgados por la cátedra.
- Facundo leyó la documentación de QT https://doc.qt.io/, y SDL2 https://wiki.libsdl.org/SDL2/FrontPage. Además utilizó los repositorios auxiliares que proporcionó la cátedra más que nada para QT.
- Además, los tres recurrimos a la ayuda de Google (específicamente Stack Overflow) y otros sitios web, como el libro de Gehn, para obtener información y resolver problemas durante el desarrollo del proyecto.

Errores conocidos:

- El movimiento de los worms es un poco tosco, cumple con los requisitos de la consigna si se lo trata con amor y uno se ingenia para acomodar al gusano pero es un poco complicado de moverse debido a los AABBs y la hitbox que tienen los cuerpos en box2D. El movimiento fue una de las cosas que más costó en la parte del juego y en la que Theo perdió mucho tiempo innecesariamente.
- Los fragmentos de los proyectiles cuando explotan abajo de una plataforma en algunos casos pueden dañar a los worms que se encuentran arriba en la plataforma por como spawnean
- Cuando un worm apunta con un proyectil que no explota con impacto (una granada) en una dirección para abajo, puede hacer que la granada traspase la plataforma. Esto pasa debido al offset que tiene la creación del proyectil en el mundo de box2D pero que si se hacía más pequeña colisionaba previamente con el worm.
- Creemos que todas las features están implementadas correctamente pero no hicimos un chequeo riguroso de que toda feature funciona perfectamente bajo toda circunstancia.

Conclusión:

Creemos que este proyecto fue beneficioso para nosotros de diversas maneras, como aprender a escribir código limpio y organizado, ya que la cantidad de archivos y líneas por archivo era enorme, por lo que nos vimos obligados a modularizar y mantener todo ordenado. En nuestra opinión, lo desafiante del proyecto no fue la concurrencia ni las nuevas tecnologías (excepto Box2D), sino el volumen del mismo.

Si tuviéramos que hacerlo de nuevo, haríamos algunas cosas de manera diferente, como comenzar lo antes posible y apreciar cada segundo disponible para trabajar en el proyecto, ya que nuestra principal dificultad fue el tiempo limitado que teníamos debido a la mala organización inicial del trabajo.

A nivel de estructuras del código quizás por parte del protocolo se podría emprolijar muchisimo mas ya que hay bastante código repetido en algunos lugares. Por parte del servidor se cambiara como el worm tiene las armas ya que actualmente se tiene una instancia de cada tool en cada worm para llevar la cuenta de su munición actual (Cuando se podría guardar la munición actual en el worm).