

7542 Taller de programacion I

Manual de proyecto

Integrantes:

Alumno	padron
Pinto, Santiago Augusto	96850
Blanco, Sebastian Ezequiel	98539

Fecha de Entrega: 27/06/2017

Índice

1. Division de tareas	1
2. Evolucion del proyecto	2
3. Inconvenientes Encontrados	4
4. Analisis de puntos pendientes	5
5. Herramientas	6
6. Conclusiones	7

1. Division de tareas

Para la realización del trabajo las tareas que cada uno realizó fueron las siguientes:

Sebastian Blanco:

- Lógica del servidor-modelo
- Implementación del algoritmo A-star para el movimiento de las unidades
- Serialización/ deserialización de la información enviada entre el cliente y el servidor
- Comunicación de sockets
- Multithreading en server

Santiago Pinto:

- Animaciones del juego
- Movimiento de cámara
- Selección de unidades a partir del mouse
- Interfaces de las fábricas y fuertes
- Selección de unidad a crear a través de las interfaces según el nivel de tecnología
- Multithreading en el cliente

2. Evolucion del proyecto

Semana 1 a 2: Durante las primeras 3 semanas cada uno trabajó por separado, dado que aún no había nada para ensamblar. Del lado del servidor estas semanas se utilizaron para la implementación del algoritmo A-star y a realizar numerosas pruebas en el mismo para el posterior ensamble. Mientras que del lado del cliente estas semanas se utilizaron para decidir que librería gráfica se iba a utilizar (decidimos utilizar SDL), a la configuración de la misma y a la creación de las primeras animaciones de movimiento con algunas unidades. A esta altura del proyecto se habían cumplido con los tiempos del cronograma recomendado por la cátedra dado que el mismo pedía algunas animaciones y la implementación y correcto funcionamiento del algoritmo A-star.

Semana 2 a 4: En este periodo comenzamos a juntar las distintas partes del trabajo en un único módulo (cliente y servidor juntos) para ir probando las funcionalidades de movimiento. Al hacerlo nos encontramos con las primeras dificultades, dado que había que sincronizar correctamente las acciones del cliente y el server para que la animación se viera bien.

Dada la baja de uno de los integrantes de nuestro equipo, del lado del cliente hubo que reestructurar el cronograma para que ciertas partes que eran necesarias por el servidor estuvieran presentes. Por esto se decidió retrasar la sección del movimiento de la cámara a la semana 5 y se priorizó la implementación de las interfaces de las fábricas para la creación de unidades, así como la selección de las mismas vía mouse y las animaciones y acciones de disparo y movimiento. Por el lado del servidor se implementaron la lógica de los disparos, monitoreo, creación de unidades y conquista de territorios. A esta altura comenzamos a retrasarnos un poco del lado del cliente, dado que hubo que realizar la implementación de interfaces que habían quedado pendientes del tercer integrante en las 2 primeras semanas. El cronograma estipulaba que en este momento debíamos tener:

Servidor: Lógica del modelo sobre la creación de unidades, capturas de vehículos, ataque entre unidades.

Cliente: movimiento de cámara, selección de unidades, selección de qué unidad crear, acciones de atacar.

Dado que se hicieron las interfaces de las fábricas, la parte del movimiento de cámara aun no se había completado en esta etapa.

Semana 4 a 6: Comenzamos a juntarnos con mayor frecuencia, dado que a medida que las funcionalidades estaban terminadas de ambos lados del proyecto, era necesario probarlas para ver la correcta coordinación del cliente y el servidor. A es-

ta altura no teníamos un cliente y un servidor por separados, sino que probábamos las funcionalidades en un solo módulo en el que ensamblamos ambas partes.

Del lado del cliente se implementaron el scroll del mapa y la cámara que habían quedado pendientes en semanas anteriores y la extensión de todo lo hecho anteriormente a todas las unidades, tipos de tile y edificios (dado que por practicidad las pruebas las hacíamos con 2 tipos de unidades, dado que la extensión, si bien tomaba tiempo, era simplemente hacer cambios chicos).

Del lado del servidor se terminaron de implementar la lógica de victoria y derrota.

A esta altura del cronograma ambas partes debían tener terminada toda la lógica. Sin embargo del lado del cliente quedaban pendientes las explosiones y la implementación de la lógica para moderar el acceso de las interfaces de las fábricas a los distintos jugadores.

Semana 1 a 2: En este periodo separamos el cliente el servidor para que la comunicación de los mismos fuera a través de sockets. Previo a esto fue necesario definir un protocolo de comunicación y de qué forma haríamos para darle un formato fácil de parsear a la información que iría desde el servidor al cliente. Una vez hecho esto se procedió a la inclusión de sockets.

Luego de testeada la comunicación a través de sockets se comenzó con la parte de concurrencia de ambos lados, agregamos los hilos en aquellas partes consideramos que mejorarían la eficiencia de la aplicación y colocamos los locks en las zonas críticas.

En paralelo a la corrección de bugs de race conditions provocados por la concurrencia redactamos los informes para la presentación final.

3. Inconvenientes Encontrados

- El primer inconveniente encontrado fue a la segunda semana de proyecto, cuando ya se habían implementado algunas de las animaciones del trabajo, nos encontramos con que la librería SDL1.2 tiene de por sí pérdidas de memoria de las cuales valgrind nos advierte en cada ejecución. Si bien los leaks no aumentan conforme el juego transcurre, si están presentes y son inevitables.
- En segundo lugar, el trabajar con un integrante menos durante todo el desarrollo hizo que tuviéramos que trabajar un poco más en algunas partes del proyecto que no podían faltar en la entrega y, en consecuencia, nos atrasamos con el cronograma establecido.
- Por otro lado, al no haber trabajado nunca con una librería gráfica, todas las semanas era necesario aprender algo nuevo y, lamentablemente, no hay mucha documentación de la librería `SDL1.2` que sea útil, aunque si la hay de `SDL2.0`.
- Nos costó mucho sincronizar al cliente y al servidor para que las animaciones cobraran una fluidez aceptable para un jugador. A tal punto que algunas cosas se pueden mejorar todavía en ese aspecto.
- A la hora de serializar la información nos tomó mucho tiempo definir el formato en la que se enviaría la información por sockets y aun habiéndolo hecho tuvimos que realizar cambios.

4. Analisis de puntos pendientes

- Dada la baja de un integrante quedaron pendientes la inclusión de sonidos e interfaces con las caras de robots y sus barras de vida.
- La pantalla de login puede mejorarse agregando una interfaz amigable en lugar de tener que hacerlo por consola, pero no llegamos por priorizar el funcionamiento del juego.
- Se puede jugar una única partida a la vez, faltaría ampliar para que se puedan jugar N partidas.
- Faltaría agregar una interfaz de bienvenida, dado que al terminar el login, la partida comienza automáticamente.
- Al finalizar la partida podrían mostrarse pantallas más explicativas, incluyendo los puntajes, unidades matadas, etc.
- La sincronización entre el cliente y servidor podría mejorarse mucho mas, evitando tener que forzar posiciones como hacemos actualmente.

5. Herramientas

Se utilizaron las siguientes herramientas para ayudarnos en el desarrollo del proyecto:

- Edición de imágenes: Photoscape
- Generación de documentación: Latex

6. Conclusiones

En conclusión este trabajo fue el proyecto mas grande en el que hemos trabajado hasta el día de hoy. Nos vimos en la necesidad de incorporar nuevas herramientas para el desarrollo del mismo como SDL, así como de aprender el funcionamiento de el algoritmo A-star. Además nos vimos en la necesidad de mantener la comunicación constante entre los miembros del grupo para que los distintos módulos pudieran ensamblarse sin inconvenientes. Sin mencionar que es la primera vez que trabajamos en una modalidad en la que cada semana había que tener una determinada parte completa para su entrega y prueba, lo cual creemos, nos ayudo a organizarnos mejor. Aprendimos que a veces se deben priorizar ciertas tareas del proyecto sobre otras para que la entrega final tenga las mínimas funcionalidades para su uso, dado que ciertas tareas, si bien necesarias, no agregan tanto valor al producto final.