



UNIVERSIDAD DE BUENOS AIRES

FACULTAD DE INGENIERÍA

75.42 - TALLER DE PROGRAMACIÓN

LENGUAJE: C++

Primer cuatrimestre de 2022

Manual de Proyecto

Burgos, Juan Sebastian	100113	jsburgos@fi.uba.ar
Chávez Cabanillas, José E.	96467	jchavez@fi.uba.ar
Del Pup, Tomás	102174	tdelpup@fi.uba.ar
Huzan, Hugo	67910	hhuzan@fi.uba.ar

Índice

1. Enunciado	2
2. División de tareas	2
3. Desarrollo	2
3.1. Cliente	2
3.2. Server	2
3.3. Editor	3
4. Features	3
4.1. Features actuales	3
4.2. Features a agregar	4

1. Enunciado

El presente trabajo realizado se basa en un remake del juego de estrategia Dune 2000, desarrollado en el año de 1998, este juego se basa en la conquista de un planeta llamado Dune, por parte de tres casa, "Harkonnen, Ordos y Atreides", el juego original contaba con un campaña de un solo jugador y un multijugador online, en este proyecto solo se recrear el multijugador. Se usa un tipo de socket TCP bloqueante.

2. División de tareas

Los primeros días fueron de instalación de utilidades y primeras pruebas de todo lo que se iba a utilizar para el proyecto. Después de esos días, la repartición de la tarea fue más por lo que cada uno tenía intenciones de realizar, algunas de las tareas realizadas por los integrantes fue:

- Hugo: se encargo de realizar tanto el diseño del server con el cliente, junto con el modelo del juego.
- Juan: al igual que Hugo se encargo también del diseño del server, cliente y modelo del juego. Así como también el parser de un archivo tipo YAML
- Tomás: se encargo de la parte gráfica del juego, motor gráfico en SDL, realizando todos los arreglos necesarios, para que el cliente interactue con el servidor.
- José: se encargo de la interfaz de login para establecer la conexión entre el servidor y el cliente, además del editor de mapas del juego. Ambos elaborados en Qt5

Cabe resaltar que algunos de los integrantes, estuvieron colaborando entre si, ya sea buscando assets de las diferentes unidades, ordenando las diferentes imágenes, etc

3. Desarrollo

En esta sección hablaremos un poco de como se fue resolviendo cada una de las partes involucradas en el proyecto. Desde la primera semana tuvimos reuniones, para ir comentando algunas ideas, así como también tener todo preparado para ir empezando el proyecto.

3.1. Cliente

Lo primero de todo fue establecer una interfaz basica de conexión entre el servidor y el cliente, para ello se uso Qt5, para crear dicha interfaz, mientras a la par se empezaba a trabajar con SDL, y poder como primer paso mostrar un mapa, el cual era enviado desde el servidor, para después poder renderizar los edificios en distintos puntos del mapa, así como también las unidades, para finalizar con la interfaz del juego. Para lograr todo esto se iban agregando diferentes métodos al protocolo del cliente para que reciba dicha información a renderizar.

3.2. Server

Como primer paso se realizó la lectura de un archivo tipo YAML, donde contenia información del mapa, que para las primeras pruebas se agrego tambien información de los edificios donde comienzan los jugadores, la ubicación de las especias y las unidades.

A la vez se fue creando el modelo del juego y se empezo a definir movimiento de unidades a través del algoritmo de A-Star, el cual determina el mejor camino entre dos puntos del mapa. Y se agrego un ID, para determinar una unidad enemiga y poder calcular el daño que se realiza.

3.3. Editor

El primer problema que se presento fue el decidir cual tile de terreno se usaria, para el editor, después fue el como armar la tabla de tiles de 32x32px y poder dibujar correctamente el mapa, el primer paso del editor fue el crear un mapa y traducir cada trozo del mapa en un valor numerico el cual se usaria para construir el archivo YAML. Así como definir el punto de inicio para los jugadores. El segundo paso del editor, que sería la modificación del archivo YAML, fue un tanto más sencilla, dado que al tener ya la forma de cortar los trozos del tile de terreno y poder renderizar el mapa, dado que cada trozo se guardaba en una matriz con el código del tile correspondiente, lo complicado fue crear una ventana para cargar el archivo y que las coordenadas leidas se posicionarán adecuadamente en la pantalla y con eso poder editar el mapa.

4. Features

Hablaremos de los features que tenemos hasta ahora asi como también de los que nos gustaría agregar.

4.1. Features actuales

- Correcta conexión al servidor.
- Operaciones de Creación, Unión y listado de partidas.
- Modelado de mapa, unidades y edificios.
- Modelado de interfaz del jugador, para la creación de unidades y edificios
- Carga de mapa, configuraciones varias desde los archivos YAML.
- Editor con funcionalidad de crear y editar mapas.
- Posibilidad de configurar los parámetros del juego desde un archivo de texto, sin necesidad de recompilar el código.
- Movimiento de unidades utilizando algoritmo de búsqueda A-star
- Unidades:
 - Movimiento(cliente y server)
 - Ataque(server)
- Edificios:
 - Estado (cliente y server)
- Selecccion de unidades y edificios.
- Selecccion multiple de unidades y edificios arrastrando mouse.
- Creación y funcionalidad del gusano de arena del lado del server.
- Mejorar flujo en las ventanas de Qt5 del cliente.
- Una sala de espera, antes de poder iniciar la partida de juego.
- Renderizar el gusano de arena en la partida.

- Manejo de energía.
- Sonido fondo
- Edificios:
 - Animaciones & Sonido:
 - Crear
 - Destruir
 - Contruccion segun terreno
- Unidades:
 - Sonido & animaciones:
 - Crear
 - Ataque
 - Animación Morir
- Tiempo de creacion segun cada uno pedido al server
- Vender edificio

4.2. Features a agregar

- Unidades:
 - Sonido:
 - Disparos
 - Animación Morir