

Trabajo Práctico Final: Dune 2000
Informe General

Iglesias, Matias Sportelli Castro, Luciano Alvarez Juliá, Santiago

4 de diciembre de 2018

Índice

1. División de tareas	2
2. Inconvenientes encontrados	2
2.1. Editor	2
3. Análisis de puntos pendientes	2
3.1. Editor	2
4. Herramientas	2

1. División de tareas

Basándonos en el enunciado del trabajo práctico nos dividimos las tareas de la siguiente manera:

Tarea	Integrante
Servidor	Matias Iglesias
Cliente	Luciano Sportelli Castro
Editor	Santiago Alvarez Juliá

2. Inconvenientes encontrados

2.1. Cliente

El principal inconveniente encontrado en el cliente fue la dependencia de las texturas con el contexto de renderizado, que terminaron provocando dificultades a la hora de realizar un diseño adaptable. Este inconveniente está explicado con mayor detalle en el informe técnico.

2.2. Editor

Un inconveniente encontrado a la hora de programar el Editor fue la manera en que tenía que ser almacenado un mapa en memoria para que el Editor mismo lo pueda cargar y también lo puedan leer el Cliente y el Servidor. La dificultad en este caso era conseguir almacenar el mapa de manera que ocupe la menor cantidad de espacio en memoria posible, lo que aumentaría la velocidad de lectura de éste, y que además sea fácil leerlo. La solución que encontramos fue almacenarlos con el formato JSON al ser un formato ampliamente conocido y sencillo de usar. Dentro del JSON se almacenan 2 arrays: uno representa la posición de cada jugador en el mapa y el otro representa al mapa en si mismo. El array de jugadores es de largo n siendo n la cantidad de jugadores. Cada elemento del array de jugadores a su vez es un array de largo 2 que representa la posición en el mapa del jugador, $[X, Y]$. El array que representa al mapa también es una array de arrays. En este caso los arrays que se encuentran dentro del array principal son de largo k siendo k la cantidad de columnas del mapa y el array principal tiene un largo de h siendo h la cantidad de filas del mapa. Dentro de los arrays secundarios se ubican strings que representan los diferentes terrenos ubicables en el mapa.

3. Análisis de puntos pendientes

3.1. Cliente

Entre los puntos pendientes del cliente quedaron:

- Reestructurar el sistema de renderizado de modo de poder tener un almacenamiento en caché mucho más fuerte.
- Mejorar el sistema que evita que se reproduzcan sonidos simultáneamente de modo de hacerlo más inteligente.
- Reestructurar el Área de Juego.
- Ofrecer una pantalla de carga más informativa, que indique el estado de los otros jugadores.

- Mejorar el renderizado del terreno, reutilizando los fragmentos de texturas anteriores.

3.2. Editor

Para mejorar la experiencia del usuario se podría agregar un feature para cuando hay que seleccionar un terreno de la pestaña Terrenos para poder ubicarlo en el mapa que facilite la ubicación de un mismo terreno en una zona grande del mapa. La idea sería que al mantener presionado el click izquierdo del mouse y moverlo sobre el mapa se ubicaría el terreno seleccionado de la pestaña en esa zona del mapa.

Otro feature interesante que haría al Editor más inteligente, que a su vez lo implementan editores de mapas reconocidos como el del videojuego Age of Empire, sería que en vez de mostrar distintas variaciones de un mismo tipo de terreno en la pestaña Terrenos, mostrar un único sprite por tipo de terreno y que un algoritmo se encargue de decidir cual sprite de cada tipo es el adecuado de ubicar en esa zona del mapa específica.

4. Herramientas

Para programar las distintas aplicaciones requeridas por el TP se utilizaron las siguientes herramientas:

- Sistema de compilación: CMake.
- Control de versiones: GitHub.
- Editor de interfaz gráfica: Qt Designer.
- Herramientas para debug: memcheck, callgrind, gdb.
- Librerías gráficas: Qt, SDL.
- Librerías generales: JSON para C++ modernos.