

Trabajo Práctico Final

Jazzjack Rabbit 2

Manual de Proyecto

[75.42] Taller De Programación I
Primer cuatrimestre de 2024

Apellido/s	Nombres	Padrón	Correo electrónico
Brizuela	Valentin	108071	vabrizuela@fi.uba.ar
Cabrera	Isaias A.	108885	isaiascabrera20@gmail.com
Mokorel	Pablo A.	103029	pmokorel@fi.uba.ar
Perez Esnaola	Lucas	107990	lpereze@fi.uba.ar

Índice

1. Introducción	2
2. Organización por semana	2
3. División de tareas	2
3.1. Brizuela Valentin	2
3.2. Cabrera Isaias A.	2
3.3. Mokorel Pablo A.	2
3.4. Perez Esnaola Lucas	3
4. Desarrollo	3
4.1. Cliente	3
4.2. Servidor	3
4.3. Editor	3
5. Features	4
5.1. Funcionalidades Actuales	4
5.2. Características Futuras	4
6. Informacion adicional	4
6.1. Puntos problemáticos	5
6.2. ¿Cumple con todo lo pedido?	5
6.3. ¿Hay algun error conocido?	5
6.4. ¿Que nos hubiese gustado cambiar o mejorar?	5

1. Introducción

En este documento se detallan las tareas realizadas por cada integrante del equipo, la organización y, las herramientas utilizadas.

2. Organización por semana

- **Semana 1:** Reunión inicial para definir el proyecto y asignar tareas. Investigación sobre el motor gráfico a utilizar. Definición de un protocolo inicial. Lógica básica del juego.
- **Semana 2-4:** Desarrollo del motor gráfico y de las primeras mecánicas del juego. Sonido. Desarrollo de la comunicación cliente-servidor.
- **Semana 5-7:** Implementación de interfaces cliente para la conexión y reproducción del juego. Varios jugadores por partida. Primera entrega.
- **Semana 8-10:** Pulido del juego, corrección de errores, refactorización y preparación de la entrega final (ver faltantes en README de la primera entrega).

3. División de tareas

3.1. Brizuela Valentin

- Desarrollo del motor gráfico.
- Desarrollo de la lógica del juego.
- Implementacion de camara.
- Desarrollo de fisicas y colisiones.
- Menu de pausa y pantalla final de partida.
- Implementacion de la comunicacion cliente-servidor durante la partida.

3.2. Cabrera Isaias A.

- Desarrollo de dibujado de juego
- Desarrollo de editor de mapas
- Desarrollo de menu de seleccion de mapas
- Creacion de diseño de mapas
- Persistencia y cargado de mapas
- Implementacion de efectos de sonido

3.3. Mokorel Pablo A.

- Diseño servidor y cliente.
- Diseño e implementació protocolo.
- Diseño e implementación de la comunicación cliente-servidor.
- Diseño e implementación aceptador/manejador de clientes (Servidor).
- Diseño e implementación de interfaces de SETUP (Cliente).
- Diseño e implementación manejador de juegos y partidas (Servidor).

3.4. Perez Esnaola Lucas

- Implementación de animaciones y sonidos.
- Desarrollo de enemigos.
- Desarrollo de drops aleatorios.
- Desarrollo de municiones.
- Desarrollo de personajes y sus acciones especiales.
- Diseño del estado del juego (Cliente).
- Implementación de cheats
- Archivo de configuracion YAML.
- Diseño de la interfaz de conexión.

4. Desarrollo

En esta sección hablaremos de cómo se fue resolviendo cada una de las partes involucradas en el proyecto.

4.1. Cliente

Como primeros pasos para la implementación del juego del lado del cliente, se decidió comenzar con una interfaz básica de conexión utilizando Qt5. Simultáneamente, se inició el trabajo con SDL para desarrollar las primeras físicas del juego, las cuales serían enviadas desde el servidor para que el cliente interpretara los comandos. Para lograr una comunicación efectiva entre el cliente y el servidor, se desarrolló un protocolo base que se fue ampliando con nuevos métodos y clases conforme avanzaba el proyecto. De esta manera, se aseguró una comunicación fluida y precisa entre ambos extremos.

4.2. Servidor

En cuanto al lado del servidor, se enfocó en desarrollar la lógica del juego y la gestión de las conexiones de los clientes. Se comenzó por implementar un servidor básico que pudiera manejar múltiples conexiones simultáneas en un solo juego y procesar los comandos enviados por los clientes.

El servidor se diseñó para interpretar los comandos recibidos, actualizar el estado del juego en consecuencia y enviar las actualizaciones necesarias a todos los clientes conectados. Se añadieron funcionalidades adicionales para gestionar partidas, mantener la sincronización del juego y manejar eventos específicos del juego.

A medida que avanzaba el desarrollo, se incorporaron mejoras como la creación de varias partidas y el envío de diversa información adicional.

4.3. Editor

Para la creación de mapas personalizados, se desarrolló un editor de mapas que permitiera a los usuarios diseñar y guardar sus propios niveles de juego. El editor se implementó utilizando una interfaz gráfica intuitiva que facilitara el proceso de diseño.

Se incluyeron funcionalidades para colocar elementos del juego, definir zonas específicas y establecer propiedades de los objetos del mapa. Los mapas diseñados se podían guardar en un formato compatible con el motor del juego (CSV), permitiendo su carga.

El desarrollo del editor de mapas involucró la creación de herramientas de edición, como selección, arrastre y eliminación de objetos, así como la capacidad de previsualizar el mapa antes de

guardarlo. Esto permitió una mayor flexibilidad y creatividad a los usuarios en el diseño de sus propios niveles.

5. Features

En esta sección se detallan las funcionalidades actuales del proyecto y las características que nos gustaría agregar en el futuro.

5.1. Funcionalidades Actuales

- **Multijugador en Línea:** Soporte para múltiples jugadores conectados simultáneamente en una o varias partidas.
- **Interfaz de Conexión:** Una interfaz de usuario básica para la conexión de los clientes utilizando Qt5.
- **Motor de Físicas:** Implementación de físicas del juego utilizando SDL.
- **Servidor Robustecido:** Un servidor capaz de manejar múltiples conexiones, interpretar comandos y sincronizar el estado del juego entre los clientes.
- **Gestión de Partidas:** Funcionalidades para crear, unirse y gestionar varias partidas de juego.
- **Editor de Mapas:** Una herramienta gráfica para la creación de mapas personalizados, con la capacidad de guardar y cargar mapas en formato CSV.
- **Sonido:** Implementación de efectos de sonido y música de fondo para mejorar la experiencia de juego.
- **Daño entre Jugadores:** Desarrollo de la funcionalidad que permite a los jugadores infligir daño a otros jugadores.
- **Sistema de Armas y Municiones:** Implementación de un sistema de armas y municiones para los personajes.
- **Pantalla Final con Scoreboard:** Implementación de una pantalla final que muestre el scoreboard.

5.2. Características Futuras

- **Chat en el Juego:** Integrar una funcionalidad de chat para que los jugadores puedan comunicarse durante las partidas.
- **Modos de Juego Adicionales:** Incluir nuevos modos de juego como captura la bandera, todos contra todos, y supervivencia.
- **Personalización de Personajes:** Permitir a los jugadores personalizar la apariencia y habilidades de sus personajes.
- **Sistema de Logros:** Implementar un sistema de logros y recompensas para incentivar el progreso y la competencia entre jugadores

6. Informacion adicional

En esta sección hablaremos de cuáles fueron los puntos mas problemáticos, si hay algun error conocido, y si cambiaríamos algo si empezáramos de vuelta.

6.1. Puntos problemáticos

- **Cierre de partidas**
- **Congruencia de mapas entre Server y Cliente**
- **Editor de mapas**

6.2. ¿Cumple con todo lo pedido?

- **Explosiones con radio:** Por falta de tiempo no pudimos realizar explosiones con funcionalidad (solo muestra animacion)
- **Animacion de muerte:** Nos hubiese gustado que los personajes hagan una animacion al morir.
- **Saltar hacia adelante:** El personaje no tiene una animacion de salto hacia adelante, solo verticalmente.

6.3. ¿Hay algun error conocido?

- **Spam habilidad:** Si el peronaje Jazz toca repetidamente la habilidad se queda levitando, no pudimos solucionarlo por falta de tiempo.
- **Cerrar en espera:** Si cerramos el cliente mientras esperamos por mas jugadores, este se bloquea.
- **Seleccion de mapa:** La seleccion del primer mapa del ComboBox al editar o iniciar partida puede que no se haga si no se selecciona primero otro mapa ya que por defecto el campo esta vacio por mas que a veces aparezca el nombre apenas iniciar.
- **Morir al cerrar:** Si cuando termina la partida, nuestro personaje esta muerto, el cliente se rompe. Encontramos este error al final y no pudimos solucionarlo a tiempo.

6.4. ¿Que nos hubiese gustado cambiar o mejorar?

- **Refactorizar:** Nos hubiese gustado tener mas tiempo para refactorizar ciertas clases o metodos que quedaron muy largos.
- **Sistema de drops:** El sistema de cómo los enemigos dropean items no es escalable, nos hubiese gustado cambiarlo.
- **Mayor testeo:** Nos hubiese gustado tener el tiempo de hacer una serie de tests mas extensos que sirvan como pruebas de regresión para futuras características.
- **Instalador:** Nos quedo pendiente hacer un instalador que facilite la instalación del juego.
- **Traducción:** Durante el desarrollo del trabajo, fuimos alternando codigo en inglés y en español. Pudimos cambiar los nombres de las clases a inglés para mayor profesionalidad, pero nos faltó cambiar el código.