

Trabajo Práctico Final

Manual de proyecto

[7542/9508] Taller de programación
Curso Veiga
Segundo cuatrimestre de 2020

Alumno	Padrón
Vázquez Lareu, román	100815
Reberendo, Manuel	100930

Índice

1. Introduccion	2
2. Enunciado	2
3. División de tareas	2
4. Evolución del proyecto	2
5. Inconvenientes encontrados	3
5.1. POO y la implementación del ajedrez	3
5.2. Procesamiento de eventos del cliente	3
5.3. Estructura para resolucion de mediciones	3
5.4. Proin sodales leo dapibus sapien fermentum	3
6. Herramientias utilizadas	3
7. Conclusiones	3

1. Introducción

El presente manual de proyecto reúne la documentación referente al manual del proyecto del trabajo práctico final de la materia Taller de programación I, el cual consistía en una aplicación cliente-servidor.

2. Enunciado

La consigna consistía en desarrollar un ajedrez cuántico. A este juego básico había que sumarle la modalidad multijugador y permitir observar la partida. Tanto el servidor como el cliente debían ser implementados en C++, utilizando SDL para la interfaz gráfica del cliente.

3. División de tareas

Vázquez Lareu, Román:

- Logica de superposicion
- Mostrar las piezas clásicas y en estado de superposición
- Reproducior música, mostrar texto en pantalla
- Mostrar el tablero y las visualizaciones de superposicion
- Pantalla de login

Reberendo, Manuel:

- Soporte partida multijugador y observadores
- Protocolo cliente-servidor

Ambos:

- Logica de partida
- Logica de medicion
- Interacción por parte del usuario

4. Evolución del proyecto

Pasos seguidos para el desarrollo del proyecto:

- Cliente con capacidad de mover piezas sobre tablero
- Lógica de ajedrez mínima en el cliente
- Lógica de ajedrez movida al servidor
- Desarrollo de logica de superposicion (merge, split)
- Logica de ajedrez cuántico del lado del servidor
- Captura de eventos del cliente para logica de ajedrez cuántico

5. Inconvenientes encontrados

5.1. POO y la implementación del ajedrez

Al momento de definir los casilleros a los que podían moverse las piezas, surgió el problema del quien tendría que realizar estas tareas. Si lo hacían las piezas, estas debían a su vez conocer la posición de otras y por lo tanto el tablero. Si lo hacía el tablero, debía conocer de qué tipo era cada pieza y cómo se movían. Se optó por que las piezas supieran cómo moverse en un tablero vacío y con esas posiciones el Tablero las recortara de acuerdo al resto de las piezas.

5.2. Procesamiento de eventos del cliente

Surgieron varios inconvenientes a la hora de resolver que ocurría cuando un cliente no "seguía el camino feliz" a la hora de hacer los clicks. De esta manera se implementó una lógica de desecciones en el servidor, buscando que la experiencia del usuario fuera lo más intuitiva y menos restrictiva posible.

5.3. Estructura para resolución de mediciones

Se definió una estructura de árbol binario donde únicamente las hojas fueran las piezas en el tablero, y el resto del árbol ayudara a seguir rastro de la historia de la misma. De esta manera fue sencillo resolver las redistribuciones de probabilidades, no así la liberación de las piezas.

5.4. Proin sodales leo dapibus sapien fermentum

Quisque tempus, tortor et convallis interdum, ipsum leo tempus ipsum, in molestie tortor arcu sit amet tellus. Praesent fermentum hendrerit nulla. In maximus ornare maximus. Nullam consectetur placerat enim sit amet lacinia. Etiam pellentesque tellus consectetur hendrerit iaculis. Sed non laoreet felis.

6. Herramientas utilizadas

Git/Github

GCC

CMake

Make

Valgrind

VScode

Discord

Material de la cátedra

Jamboard

7. Conclusiones

El trabajo ponía a prueba lo aprendido en la cursada, pero no como desafío sino como herramienta. De no haber adquirido fundamentos sólidos en sockets y threads, el presente trabajo hubiese estado muy por fuera del alcance de los integrantes. A su vez, la dimensión del mismo obligó a fijar fechas, horarios y objetivos de antemano para lograr llegar con una aplicación funcional.

a la entrega final. Un claro ejemplo de esto fue la dedicación de varias reuniones a la resolución de la estructura a utilizar para implementar la lógica de superposición.

Si bien hubiese sido conveniente avanzar con el proyecto de manera lenta y segura, por los tiempos la dinámica fue más parecida a avanzar, testear, debuggear y así sucesivamente. A pesar de esto, no hay duda de que la realización del trabajo deja un saldo positivo, llevando a los integrantes a aprender el uso de herramientas nuevas, poniéndolas en práctica de manera correcta y a intercambiar constantemente (ya sea peer programming como por otros canales) con el objetivo de mantenerse al tanto del proyecto y desarrollar la aplicación consensuando las acciones a llevar a cabo.