Documentacion de Acreditacion

Comunicación Cliente-Servidor

Aprendizaje: Se adquirieron conocimientos relacionados con la implementación de la comunicación entre el cliente y el servidor en juegos multijugador. Esto implicó comprender en profundidad cómo utilizar sockets para establecer una conexión bidireccional que permitiera la transmisión de datos en ambas direcciones.

Desafío: El desafío principal en este aspecto fue lograr una transmisión de datos fluida. No se trata únicamente de establecer una conexión; sino garantizar que los datos enviados y recibidos sean exactos y se procesen de manera eficiente. La complejidad se enfocó en la gestión de flujos de entrada y salida, que involucra la lectura y escritura de datos, así como asegurar que esta comunicación funcione en tiempo real.

Para superar estos desafíos, se implementó una comunicación cliente-servidor mediante el uso de sockets. El cliente establece una conexión con el servidor a través de la dirección IP y el puerto del servidor. Se emplearon flujos de entrada y salida, como BufferedReader y PrintWriter, para enviar y recibir datos entre el cliente y el servidor. Para simplificar la serialización y deserialización de objetos Java en formato JSON, se recurrió a la biblioteca Gson. Esto facilitó en gran medida la transferencia de datos entre ambas partes y garantizó que la comunicación fuera eficiente.

Representación de Datos del Juego

Aprendizaje: Se destacó la importancia de seleccionar las estructuras de datos adecuadas para representar los elementos del juego de manera organizada y coherente. En este proyecto, la clase GameData se eligió como un parte fundamental para esta tarea.

Desafío: El reto principal consistió en diseñar la clase GameData de tal manera que fuera lo suficientemente adaptable para representar varios tipos de datos del juego, como puntos, líneas, colores y puntuaciones. Esto requería una planificación de las variables y métodos de la clase, asegurando que cada tipo de dato pudiera ser representado bien y que fuera fácilmente convertible a JSON para la comunicación con el servidor.

Para superar este desafío, se implementaron métodos de fábrica estáticos en la clase GameData. Estos métodos permitieron crear instancias de GameData que representaban diferentes tipos de datos del juego, como puntos, líneas, colores y puntuaciones. Cada método configuraba las variables de la instancia GameData creada y establecía su tipo. La variable 'type' se empleó para etiquetar cada instancia con su tipo de datos correspondiente. Esto permitió la adaptabilidad de la clase GameData, permitiendo la representación de múltiples tipos de datos sin requerir una clase diferente para cada uno. Además, se utilizó la biblioteca Gson para la serialización y deserialización de objetos GameData a formato JSON, facilitando así la transferencia de datos.

Interfaz Gráfica y JavaFX

Aprendizaje: El desarrollo de interfaces gráficas interactivas con JavaFX implica la creación de componentes visuales, la gestión de eventos y la actualización en tiempo real de la interfaz.

Desafío: Uno de los desafíos clave fue implementar una cuadrícula de puntos en un Pane de JavaFX y gestionar eventos de teclado para permitir el movimiento del jugador. También fue de suma importancia dibujar líneas en la interfaz gráfica utilizando la clase Line, garantizando que se mostraran correctamente y en el orden adecuado.

La implementación se basó en el uso de JavaFX para crear una interfaz gráfica interactiva. Se generó una cuadrícula de puntos representados por objetos Circle y se incorporaron a la escena. Los eventos de teclado se manejaron para permitir el movimiento del jugador dentro de la cuadrícula. La actualización en tiempo real de la interfaz se logró mediante Platform.runLater() para garantizar una modificación segura de la interfaz gráfica.

Control de Jugador con Teclado y Puerto Serial

Aprendizaje: Integrar el control del jugador desde el teclado y desde un puerto serial, como un Arduino, requiere la gestión de eventos y la sincronización de acciones.

Desafío: El principal desafío en este aspecto fue en garantizar que el control del jugador fuera preciso y sincronizado, independientemente de si se realizaba mediante eventos de teclado o mediante datos del puerto serial. Configurar y abrir adecuadamente el puerto serial, así como leer datos de manera eficiente, también representó dificultades.

Para abordar estos desafíos, se configuró la escucha de eventos de teclado para el control del jugador. Además, se empleó la biblioteca jSerialComm para la comunicación con el puerto serial, asegurándose de configurar y abrir el puerto de manera apropiada. Se creó un hilo separado, denominado serialReaderThread, para leer datos del puerto serial de manera eficiente.

Dibujo de Líneas y Actualización de la Interfaz Gráfica

Aprendizaje: Dibujar líneas entre puntos seleccionados en la cuadrícula y actualizar la interfaz gráfica en tiempo real representó un desafío importante.

Desafío: Uno de los desafíos clave en esta área se centró en implementar el método drawLineFromReceivedData, encargado de tomar las coordenadas de puntos recibidas del servidor y dibujar líneas correspondientes en la interfaz gráfica. También resultó importante asegurar que las líneas se dibujaran en el orden correcto y se ajustaran adecuadamente a las coordenadas de los puntos en la cuadrícula.

Para solucionar estos desafíos, se implementó el método drawLineFromReceivedData, que transformaba las coordenadas de puntos en coordenadas de píxeles en la interfaz gráfica mediante una simple fórmula de escala. Las líneas se agregaron al Pane de fondo en el orden correcto, asegurando que se dibujaran adecuadamente y que representaran los datos recibidos del servidor.

Manejo de Puntajes y Colores de Jugadores

Aprendizaje: Llevar un seguimiento de los puntajes de los jugadores y asignar colores a los clientes se reveló esencial para la experiencia de juego.

Desafío: Uno de los desafíos clave en esta área fue garantizar que los puntajes se mantuvieran precisos y se actualizarán correctamente en la interfaz gráfica a medida que se recibieran datos del servidor. Además, era necesario mostrar el color del jugador en la interfaz para que los usuarios pudieran identificarse fácilmente.

Para resolver estos desafíos, se asignaron colores a los clientes mediante el envío de datos de color desde el servidor. Este último rastreó los puntajes de los jugadores y los envió al cliente correspondiente cuando fue necesario. En la interfaz gráfica, se mostró el color del jugador en el punto en el que se encontraba, permitiendo así una experiencia de juego completa.

Manejo de Errores y Excepciones

Aprendizaje: Reconocer y gestionar posibles errores y excepciones en una aplicación es fundamental para garantizar su robustez.

Desafío: Uno de los desafíos principales abordados consistió en la gestión de excepciones al establecer conexiones de socket y al leer datos del puerto serial. La impresión de errores resultó esencial para la depuración y el mantenimiento de la aplicación.

La solución a estos desafíos se basó en la implementación de bloques try-catch para capturar excepciones al establecer conexiones de socket y al leer datos del puerto serial. Los errores se imprimieron en la consola, lo que facilitó la depuración y el seguimiento de problemas potenciales. Esto contribuyó a evitar bloqueos inesperados de la aplicación.