



Lenguajes, Compiladores e Intérpretes CE-3104

Tarea 3 Lenguajes

PaCE-Man

Estudiantes

Juan Peña Rostrán 2018080231

Ignacio Morales Chang 2020096499

Anthony Montero Román 2019275097

Profesor

Marco Rivera Meneses

18 de noviembre 2021

Manual de usuario

Para el desarrollo del programa se utiliza el lenguaje de programación C con el compilador gcc para la creación del servidor y el lenguaje java en su versión openJDK 11 para la creación del cliente e interfaz, además se utiliza el sistema operativo Ubuntu 20.04, por lo que se recomienda tener estos lenguaje y S.O y preferiblemente usar las versiones mencionadas de cada uno para la ejecución del mismo.

Para el funcionamiento del programa, se debe ejecutar primeramente el archivo server.c desde la terminal de linux con el siguiente comando: "gcc server.c -ljson-c -o server" y seguidamente el comando "./server". Por haber ejecutado el servidor, puede seguidamente ejecutar el archivo main del repositorio de java el cual ejecutará la interfaz y a su vez el cliente. Por haber ejecutado estos dos programas, podrá hacer uso del juego paceMan el cual su personaje principal se manipula con el uso de las teclas "w", "a", "s", "d".

Descripción del uso de las estructuras de datos desarrolladas

Listas:

Las listas se utilizan para guardar diversos elementos que se encuentran en pantalla. Por ejemplo:

```
LinkedList<Ghost> ghostList = new LinkedList<Ghost>();  
LinkedList<Dot> dotList = new LinkedList<Dot>();  
LinkedList<Fruit> fruitList = new LinkedList<Fruit>();  
LinkedList<Energizer> energList = new LinkedList<Energizer>();
```

Se utilizan listas para elementos que podrían desaparecer de la ventana. Veamos el caso de un punto cuando es comido por el jugador:

```

for (int l = 0; l < dotList.size(); l++) {

    if (dotList.get(l).intersects(ball)) {

        //SUMAR PUNTOS
        score.points += dotList.get(l).value;
        dotList.remove(l);
    }
}

```

Podemos observar cómo se recorre la lista dotList y se revisa si alguno está colisionando con el jugador “ball” y si esto es cierto, se agregan puntos y se elimina el punto contra el que se está colisionando de la lista dotList.

Array:

Las listas se utilizan para guardar diversos elementos que se encuentran en pantalla los cuales no desaparecen nunca. Por ejemplo las paredes:

```
Wall[] wallArr = new Wall[4];
```

A la hora de instanciar paredes, se deben guardar en este array como se hace en el método newWalls():

```

public void newWalls() {
    wallArr[0] = new Wall(cons.SQUARE_SIDE, GAME_HEIGHT - (6 *
        cons.SQUARE_SIDE), cons.SQUARE_SIDE, 5 * cons.SQUARE_SIDE);
    wallArr[1] = new Wall(GAME_WIDTH - 2 * cons.SQUARE_SIDE, GAME_HEIGHT - (6 *
        cons.SQUARE_SIDE), cons.SQUARE_SIDE, 5 * cons.SQUARE_SIDE);
    wallArr[2] = new Wall(2 * cons.SQUARE_SIDE, 2 * cons.SQUARE_SIDE, 4 *
        cons.SQUARE_SIDE, 4 * cons.SQUARE_SIDE);
    wallArr[3] = new Wall(GAME_WIDTH - 6 * cons.SQUARE_SIDE, 2 *
        cons.SQUARE_SIDE, 4 * cons.SQUARE_SIDE, 4 * cons.SQUARE_SIDE);
}

```

Descripción detallada de los algoritmos desarrollados

Problemas sin solución:

1. Serialización y deserialización de los datos:

La conexión entre el server y el cliente está disponible pero no se logró implementar el envío de la misma en formato JSON, el problema que se presentó fue que al crear un string de la biblioteca JSON-C no fue compatible con el tipo de dato char del lenguaje C, solamente lo aceptaba si el puntero al objeto json de la biblioteca se creaba en la función main.

2. Administración de vidas, puntos y velocidad:

En el server no fue posible manejar estos detalles por el mismo problema de serialización.

3. Existe un bug en la interfaz en el que los puntos se dibujan encima de las paredes del mapa.

4. El jugador a pesar de tener 3 vidas pierde en el instante en que choca con un fantasma, no se pudo reparar el bug que provoca esta situación.

Plan de Actividades realizadas por estudiante

El estudiante Anthony Montero investigará el funcionamiento e implementación del servidor y cliente tcp con la respectiva conexión y envío de información, además del manejo de json para la parte de java y la conexión entre la interfaz y el cliente para el envío y recepción de información.

El estudiante Juan Peña realizó la conexión de múltiples clientes en el servidor nuevo pues el que había no podía manejar más de un cliente, además implementó las estructuras para el juego PaCe-Man y el procesamiento de mensajes desde el cliente, también se investigó las bibliotecas para el manejo y creación de archivos JSON.

El estudiante Ignacio Morales se encargará de la investigación y desarrollo de las distintas partes de la interfaz, como lo son la creación de objetos, movimiento y las respectivas colisiones entre objetos y el producto de estas colisiones.

Problemas encontrados

1. Paso de datos entre los bytes del server y el cliente debió leerse en el cliente mediante un método distinto al disponible en Java, pues la codificación UTF-8 de java no fue compatible con la de C.
2. Cuando el primer server implementado recibía la conexión del segundo se terminaba el programa, pues la solicitud de otro socket interrumpía la que estaba ya establecida.
- 3.

Recomendaciones

Con respecto al tráfico de bytes por medio de sockets es necesario establecer un protocolo de comunicación que permita recuperar la información tanto en el cliente como en el servidor. Se recomienda experimentar a profundidad la conversión de los objetos de las bibliotecas JSON con respecto a los tipos de datos que posee el lenguaje de programación en donde se codifican el server y el cliente.

Conclusiones

En la participación de este proyecto se permite comprender la mecánica que hay detrás del uso de punteros y comunicación entre un cliente y un servidor de protocolo TCP, donde pudimos descubrir que el paso de información entre un lenguaje de alto nivel como java y otro de más bajo nivel como c es mas que solo pasar una cadena de caracteres logrando entender la importancia del uso de punteros en lenguajes de este tipo para el buen desarrollo de los programas.

Bibliografía

Chuidiang. (2021, 10 21). Socket entre C y java. Retrieved from Socket entre C y java: http://www.chuidiang.org/java/sockets/cpp_java/cpp_java.php

Pac-Man. (2021, 11 02). Pac Man. Retrieved from webpacman: <https://www.webpacman.com/retrogames.cz>. (2020, 03 18). retrogames. Recuperado de retrogames:

http://www.retrogames.cz/play_064-Atari2600.phpWikipedia.

(2021, 11 02). Pac-Man. Retrieved from Wikipedia: <https://es.wikipedia.org/wiki/Pac-Man>

Refactoring Guru. (s.f.) Singleton. Retrieved from: <https://refactoring.guru/es/design-patterns/singleton>

GeeksforGeeks. 2021. Socket Programming in C/C++ - GeeksforGeeks. [online] Recuperado de: <<https://www.geeksforgeeks.org/socket-programming-cc/>>.