



Instituto Tecnológico de Costa Rica

Módulo de Leguajes

Tarea Corta 3:

BreakOutTEC

Profesor:

Marco Rivera Meneses

Estudiantes:

Rachel Pereira González 2020186432

Bryan Gómez Matamoros 2020211778

Gregory Alpízar Céspedes 2020039796

15 de junio, 2022

Contenidos

Descripción de las estructuras de datos desarrolladas	3
Archivo blocks.json.....	3
Descripción detallada de los algoritmos desarrollados	3
Manejo de json	3
Problemas sin solución	4
Plan de Actividades.....	4
Problemas encontrados	5
Conclusiones del proyecto	5
Recomendaciones del proyecto.....	5
Bibliografía consultada	5
Bitácora de los estudiantes.....	6

Descripción de las estructuras de datos desarrolladas

Archivo blocks.json

Este archivo se encuentra en la carpeta del servidor y contiene toda la información relevante para el servidor acerca de cada uno de los bloques del juego. Su estructura está basada en una lista asignada como “blocks”, para la cual cada miembro corresponde a un bloque (80 en total). Cada bloque tiene un id, un color, una puntuación asignada y la especificación del poder que le brinda al usuario al romperse.

En este proyecto se utilizaron más estructura de datos, pero provenientes de librerías con los JSONArray, donde esta estructura se utiliza en la parte del Cliente en Java para poder recorrer correctamente el Archivo de Json que envía el Servidor, de la misma forma también se utilizan array para la creación para dibujar los bloques.

Descripción detallada de los algoritmos desarrollados

Manejo de json

Dado que el servidor es el encargado del manejo de archivos json, se tiene un algoritmo encargado de asignar valores a los bloques al iniciar una nueva partida, este algoritmo inicialmente toma los elementos del archivo y los separa según su necesidad en color, puntuación y estado. Se elige un número aleatorio entre 10 y 25 y se le asigna a todos los bloques de un mismo nivel, con respecto al estado, se trabaja con una probabilidad de 7% para cada uno de los poderes, contra un 58% de no tener poder asignado, posteriormente se repite lo mismo para cada uno de los niveles de bloques tanto en puntuación como en estados.

Respecto a la lectura del archivo, solamente se toma toda la información que este contenga y se convierte a cadena de texto para que sea utilizada por el servidor como un mensaje.

De la misma forma como se manejan el archivo Json en C se debe manejar en java donde se debe transformar el dato String recibido por el servidor a un Objeto Json para procesar por medio de ciclos y las Funciones de ObteinState y ObteinPoints, los valores que se necesitan del archivo, esto es todo un proceso de decodificación del mismo.

Lógica de Juego

Esta presenta múltiples funciones para manipular los bloques y las colisiones de la bola con estos, de forma que cuando colisionan deben eliminarse, otra parte del algoritmo de esta lógica es la obtención de los puntos y los poderes a la hora de que este bloque es destruido por la bola en juego. Para esta se debe obtener lo asignado en el clase bloque por medio de .get() para que luego este dato sea manipulado y asignado al jugador.

Problemas sin solución

Lectura de blocks.json

A pesar de que en la aplicación el archivo funciona de forma correcta, no es muy agradable para el desarrollador la lectura manual del documento, pues el editor de la librería coloca todo el texto sobre la primera línea, y ni mediante identaciones como normalmente se hace.

Implementación del Cliente Espectador

A pesar de que existe un estado de espectador y este obtiene los datos que le envía el servidor para que pueda replicar lo que el cliente esta realizando, no fue posible actualizar los datos ya que estos solo se manejaban en el cliente y no los trasmitía el servidor.

Plan de Actividades

El plan de Actividades se realiza en la primera reunión del grupo el día 29 de abril

Estudiante	Actividades	Etapas I (3 de junio)	Etapas II (7 de junio)	Etapas III (10 de junio)	Etapas Final (14 de junio)
Bryan Gómez	Lógica de Juego	Una interfaz básica en Java	Interfaz del Juego	Interfaz del Juego Finalizada, con todas las funciones requeridas	Finalización de todo lo faltante con respecto al código y la documentación
Rachel Pereira	Sockets y Cliente Espectador	Un conexión entre cliente y servidor	Envío de cadenas de datos pequeños	Implementación de Cliente Espectador en Sockets	
Gregory Alpizar	Manipulación de Datos en el Servidor	Investigación sobre el manejo de Datos en Json	Función de Lectura en Json en C	Creación del archivo y función de parseo para poder enviarlo en el socket	
Bryan – Rachel - Gregory	Documentación	Actualización de las bitácoras	Actualización de las bitácoras	Comenzar a detallar los puntos de la documentación requeridos	

Problemas encontrados

Problema de edición de archivo json

La librería utilizada en C para el manejo de archivos json posee una cantidad muy amplia de métodos que se ponen a disposición del desarrollador, por lo que al querer utilizar los más intuitivos para editar el documento el resultado era la eliminación completa de la información que poseía el archivo. El método de solución fue revisar dentro de la lista de ejemplos y tutoriales que posee el repositorio hasta encontrar funciones que trabajen correctamente.

Problema con la manipulación del archivo json en java

A partir del problema que venia con la edición del archivo en C, este se complico en Java, entonces para resolverlo se tuvo que investigar sobre otra librería de json en Java, ya que primeramente se estaba utilizando Gson, luego se encontró json simple, que fue la que se uso en el proyecto, luego se tuvieron problemas al recorrer el archivo, pero con la siguiente referencia se logró solucionar.

Referencia <https://parzibyte.me/blog/2019/08/05/recorrer-jsonarray-java/>

Conclusiones del proyecto

- Se logra observar como para la implementación de un juego el paradigma de objetos nos facilita el trabajo, ya que trabajando en clases hace que todo se encuentre mucho más ordenado y como con el imperativo se puede complementar.
- Por otro lado, la implementación del Sockets, aclara el concepto de la transferencia de datos, sin importar el tamaño o lo que se quiere enviar, si la comunicación de los sockets se encuentra bien programada estos no presentaran perdida de datos. Observando la comunicación de ambos lenguajes.
- Por ultimo, se logra entender la importancia de la estructuras de datos que fueron utilizadas en la creación de los bloques del juego y el archivo json que se transmitió por sockets.

Recomendaciones del proyecto

- Se recomiendo investigar sobre la manipulación de archivos en los distintos formatos, ya que este fue uno de los principales problemas que tuvimos
- Investigar sobre sockets ya que para entenderlos correctamente se debe saber desde como es que se realiza la transmisión de datos.

Bibliografía consultada

Socket entre C y java. (s. f.). Chuidiang.

http://www.chuidiang.org/java/sockets/cpp_java/cpp_java.php

Richard. (2018, 27 mayo). *Gson (JSON) en Java, con ejemplos.* Jarroba.

<https://jarroba.com/gson-json-java-ejemplos/>

https://github.com/RicardoMoya/Gson_Example

<https://blog.katastros.com/a?ID=00250-976cb580-3744-41a7-a4be-7efbeba8cf98>

https://www.youtube.com/watch?v=dQyXuFWylm4&t=3s&ab_channel=HathibelagalProductions

<https://progur.com/2018/12/how-to-parse-json-in-c.html>

<https://github.com/rbtylee/tutorial-jsonc/blob/master/tutorial/edit.md>

<https://github.com/json-c/json-c>

<https://stackoverflow.com/questions/34151774/how-to-write-a-json-file-in-c>

<https://www.geeksforgeeks.org/generating-random-number-range-c/>

<https://stackoverflow.com/questions/11656532/returning-an-array-using-c>

<https://refactoring.guru/es/design-patterns>

<https://refactoring.guru/es/design-patterns/factory-method>

Bitácora de los estudiantes

Bitácora Grupal		
Fecha	Descripción	Asistentes
30 de mayo	Primera reunión grupal donde se pone claro el método de trabajo y se planea un plan de actividades, donde se define en que parte debe trabajar cada estudiante, con este se busca hacer el mayor esfuerzo posible para que se cumplir el tiempo de entrega de lo solicitado a cada estudiante	Gregory- Rachel- Bryan
3 de junio	En esta reunión se observa lo que cada estudiante ha avanzado o investigado sobre lo asignado la última reunión, a partir de esto se	Gregory- Rachel- Bryan

	continúa trabajando a partir de lo planteado en el plan de Actividades	
7 de junio	En esta reunión de igual manera cada estudiante presenta sus avances y propone ideas ante los problemas que tiene los otros compañeros para poder continuar. Ya que se debe contemplar lo que realiza el otro	Gregory- Rachel- Bryan
10 de Junio	En esta reunión se realiza la unión de los programas, se encuentran errores en la transmisión de archivos. También se comienza con el proceso de documentación del código que se tiene hasta el momento. Proponiendo idea de solución.	Gregory- Rachel- Bryan
15 de Junio	Se realizan cambios finales para que el funcionamiento del juego sea de la forma más optima posible y se finaliza con todo lo relacionado a la documentación. Se realiza el manual de usuario.	Gregory- Rachel- Bryan

Bitácora de Gregory Alpízar	
Fecha	Descripción
3 de junio	Inicio de investigación de manejo de datos en C. No se encuentra información clara.
5 de junio	Se continúa la investigación, entre las opciones se tienen: XML, txt y json.
8 de junio	Luego de discutir el tema con los compañeros de grupo se decide el uso de json para el manejo de datos del servidor y su comunicación con el juego.
10 de junio	Luego de investigar se determina que no existe un método específico en el lenguaje C para manejar archivos json, se hace uso de una librería desarrollada en github.
11 de junio	Se hacen pruebas de parseo de json, el método es tedioso y complicado, pero ya se pueden obtener cadenas de texto, números y valores booleanos.
13 de junio	Se inicia pruebas para editar archivos json, esta etapa es un poco más sencilla que la anterior, se necesitan menos líneas de código.
12 de junio	El manejo de archivos json está completamente implementado, existe una función para leer toda la información del archivo y convertirla en cadena de texto para que sea utilizada por el servidos, una función que reinicia (edita) los valores

	de cada bloque en el archivo y una función que configura cada bloque individualmente para iniciar la partida de juego.
--	--

Bitácora Rachel Pereira González	
Fecha	Descripción
1 de junio	Investigo sobre los sockets
2 de junio	A partir de el material dado por el profe y lo investigado monto un ejemplo de cliente y servidor el cual no funciona.
4 de junio	Luego de ciertas recomendaciones de mis compañeros las aplica a lo que tenia y logro una conexión entre ellos y que se envíen un mensaje
6 de junio	Continuo con la implementación de un método de conversación entre los sockets ya que tengo problemas con los mensajes continuos.
8 de junio	Con las retroalimentaciones de la última reunión, trato de incluir otras funciones para solucionar el problema
13 de junio	A partir de lo que mi compañero realiza con el archivo json lo incluyo en el servidor y comienzo con la unión del programa del juego y el cliente
14 de junio	Finalizo con la unión de cliente y juego, implementando la función de leer el archivo json enviado. Continuando con la implementación de cliente espectador, pero consigo un problema que no me deja avanzar
15 de junio	Revisamos la unificación del código e intento realizar algo con el Espectador, pero no se logró nada , finalizo la documentación restante

Bitácora Bryan Gómez Matamoros	
Fecha	Descripción
3 de junio	Investigo sobre el paradigma de programación de objetos, ya que me tocaba la realización del juego
5 de junio	Mediante la observación de varios videos tomo las primeras ideas para la implementación del juego, así como también retomo la programación de GUI en Java
7 de junio	Luego de obtener ciertos consejos de mis compañeros, continuo con la implementación de la lógica principal del juego
9 de junio	Completo la implementación de la ventana de juego, así como también con la implementación de varios objetos necesarios para el juego como el jugador y los bloques
11 de junio	Completo otros aspectos lógicos del juego como el movimiento de la bola y los poderes de cada bloque
13 de junio	Termino con los últimos detalles de la Interfaz como el menu y las condiciones de derrota
14 de junio	Comento todo el código (Javadoc)

15 de junio	Continuamos con la unificación del código. Termino mi parte de la documentación.
-------------	--