



Instituto Tecnológico de Costa Rica

Área académica de ingeniería en computadores

Lenguajes de programación

Tarea 3: breakOutTec

Profesor:

Marco Rivera Meneses

Estudiantes:

Luis David Richmond Soto 2020077226

Sergio Martínez Bonilla 2020039334

Luis David Delgado Jiménez 2020072813

I Semestre 2022

14 de junio

Contenidos

Algoritmos desarrollados.....	3
Estructuras de datos desarrolladas	3
Problemas sin solución	4
Problemas encontrados	5
Plan de actividades realizadas por cada estudiante	6
Recomendaciones	8
Conclusiones.....	9
Bibliografía.....	10

Algoritmos desarrollados

- Manejo de json.
- Manipulación de listas enlazadas.
- Detección de colisiones.
- Cambio de dirección de la bola en la raqueta.

Estructuras de datos desarrolladas

- Nodo: puede representarse como un ladrillo o una bola. Es un elemento de la lista enlazada.
- Listas enlazadas: allí se guardaron los ladrillos y las bolas (cuando hay más de una).
- Json: se utilizó para la comunicación entre los clientes y el servidor en el envío de los ladrillos, las bolas, puntaje y los demás componentes del juego.
- Struct: Se utilizó para generar listas enlazadas en C, usadas para la generación del registro de los ladrillos por parte del servidor.

Problemas sin solución

- Cuando la bola realiza una colisión en horizontal con un ladrillo, la bola lo considera como una colisión que proviene de arriba o abajo (según el ángulo de origen de la bola) y la bola atraviesa el ladrillo hasta llegar al otro lado en horizontal.
- De forma inexplicable y muy rara vez, la bola se congela en medio de la pantalla sin colisionar con ningún objeto. Ocurre con mayor frecuencia cuando se encuentra abierta la ventana del espectador.
- Al intentar agregar más bolas se da un error dentro del JavaFX por motivos desconocidos, por lo que el poder de activar una bola extra no funciona. Se intentó resolver de la misma manera que con los ladrillos, utilizar una cantidad de bolas finita y reconfigurarlas, sin embargo, seguía ocurriendo el mismo error de la interfaz por lo que se descartó el uso de este poder.

Problemas encontrados

- Las físicas actuaban totalmente en desorden si ningún sentido. Para solucionar esto se escribieron totalmente de nuevo utilizando otro método.
- Al querer borrar los ladrillos, lanza una excepción desconocida, se trabajó en eso hasta que se descubrió que esa no era la manera correcta de eliminar. Finalmente los ladrillos no se borran, solamente se reconfiguran en sus atributos.
- Al tocar el borde inferior la bola debe reiniciar su posición, pero la bola seguía estando dibujada en la pantalla, para esto se borró también de los activos de la pantalla, pero esto generaba el mismo error que los ladrillos. Se solucionó de la misma manera.
- A la hora de mandar largo de la cadena por los sockets, a la hora de convertir el int a formato de red, esto retornaba un número negativo lo cual imposibilitaba el envío de datos. Para ello se configuro el json de tal manera de que siempre diera un número positivo.
- Para recorrer los ladrillos en el json, se tenía simulando una lista enlazada donde había un next que apuntaba al siguiente ladrillo peor al parsear el string a json, esto daba un error, entonces a cada ladrillo se le puso un identificador, para poder recorrerlo.

Plan de actividades realizadas por cada estudiante

<i>Estudiante</i>	<i>Actividad</i>	<i>Tiempo</i>	<i>Entrega</i>
<i>Todos</i>	Planeación	2h	30/5/2022
<i>Luis Delgado</i>	Primer intento de Sockets	2 días	1/6/2022
<i>David Richmond</i>	Sockets finales	4 días	13/6/2022
<i>David Richmond</i>	Server	2 días	14/6/2022
<i>Sergio Martínez</i>	Primer intento de interfaz	2 días	12/6/2022
<i>Sergio Martínez</i>	Interfaz del jugador	2 días	14/6/2022
<i>Sergio y Luis</i>	Interfaz del espectador	5 horas	14/6/2022
<i>Luis Delgado</i>	Físicas y ayuda	3 días	14/6/2022
<i>Sergio y David</i>	Comunicación	2 horas	14/6/2022
<i>Luis y David</i>	Documentación y Manual	1 días	14/6/2022

Descripciones de actividad:

- Planeación: se distribuyó la carga de la tarea y se asignó los roles para completarla.
- Primer intento de sockets: Se comenzó con un tipo de sockets que no llevó a una solución completa del problema, entonces se descartaron.
- Sockets finales: Se trabajó con cJSON para C y gson para Java junto con los sockets y se logró ejecutar en tiempo real.
- Servidor: El manejo de datos por parte del servidor funciona correctamente en simulación.
- Primer intento de interfaz: se comenzó la programación de la interfaz con una biblioteca que no presentaba las herramientas necesarias.
- Interfaz del jugador: se cambió de biblioteca gráfica y se completó lo asociado con la interfaz para permitir jugar.

- Interfaz del espectador: se generó la interfaz que utilizará el espectador, interfaz en la que no podrá efectuar ningún cambio el usuario que la utilice.
- Físicas y ayuda: se crearon las físicas para la bola, colisiones y detección de ladrillos, paredes, raqueta y ángulos variables de la raqueta junto con mensajes de ayuda y “cómo jugar”.
- Comunicación: funcionamiento de los sockets puesto en práctica para conectar los dos lenguajes.
- Documentación y Manual: se generaron los documentos correspondientes a la documentación del proyecto, el manual de usuario y bitácora.

Recomendaciones

Para mejorar la utilización de un lenguaje como C, sin clases, es necesario conocer bien el uso de los struct y los punteros, herramientas muy poderosas de bajo nivel muy eficientes.

Al trabajar con punteros en C es importante saber que se necesita una correcta administración de la memoria y no solicitar espacio en memoria desmesuradamente y sin ningún control sobre las ubicaciones en memoria que ya no se utilizan.

Trabajando con Java es importante conocer bien los patrones de diseño y cómo funciona la herencia para aprovechar al máximo el lenguaje y tener un trabajo ordenado y laborar de forma eficiente.

Conclusiones

El lenguaje de programación C es muy poderoso por el bajo nivel en el que se trabaja y el lenguaje Java es muy cómodo de utilizar por el orden debido a la programación orientada a objetos.

La programación orientada a objetos es fácil de comprender, pues se asemeja a la realidad y es fácil de utilizar e implementar.

La implementación de sockets en C puede ser algo complejo, tomando en cuenta que para Windows y Linux son totalmente distintos, por lo tanto, se debe tomar en cuenta en que plataforma se va a realizar los sockets.

Bibliografía

Repositorio de GitHub: <https://github.com/acordobav/SpaceInvadersServer>

Repositorio de GitHub para cJSON: <https://github.com/DaveGamble/cJSON>

Creative Commons. (2007, 4 febrero). *Socket entre C y java*. chuidiang. Recuperado 15 de junio de 2022, de http://www.chuidiang.org/java/sockets/cpp_java/cpp_java.php

Tutoriales point. (s. f.). *JSON.simple Tutorial*. Recuperado 15 de junio de 2022, de https://www.tutorialspoint.com/json_simple/index.htm