Instituto Tecnológico de Costa Rica Ingeniería en Computadores

Lenguajes, compiladores e intérpretes

Tarea 3

Pablo Naranjo Monge - 2020084495

Jan Marschatz Aguilar - 2020026368

Axel Cordero Martínez - 2019052017

PolePositionCR

Bitacora

|  |  |  |
| --- | --- | --- |
| Fecha | Actividad | Participantes |
| 25/3/22 | Nos reunimos en llamada y  creamos los documentos  iniciales | Axel – Jan - Pablo |
| 25/3/22 | Iniciamos la programación creando archivos de server y cliente. | Axel - Jan |
| 29/3/22 | Se añada librería Gson, se empieza manejo de json. | Pablo - Jan |
| 29/3/22 | Se crea ventana en C para el GUI | Pablo |
| 30/3/22 | Se crean las clases serializar, deserializar y informacion. | Pablo -Axel |
| 31/3/22 | Se agregan imágenes al GUI, corrección de errores. | Pablo - Jan |
| 1/4/22 | Se agrega mapa que se mueve y jugador. | Pablo |
| 2/4/22 | Se añada fondo que se mueve.  Listas enlazadas en java | Pablo - Axel |
| 2/4/22 | Se crea conexión entre java y c mediante sockets | Jan – Pablo - Axel |
| 4/4/22 | Se agrega clase evento y ventana, se agrega botones, se termina parte grafica. | Pablo - Jan |
| 6/4/22 | Se agregan pantalla menú y selección de personajes. Listas enlazadas en c. | Jan – Pablo - Axel |
| 8/4/22 | Se eliminan clases de ejemplo, se añaden elementos gráficos, y cierre de ventana. | Jan – Pablo - Axel |
| 8/4/22 | Se hacen pruebas de resolución de imágenes. | Jan – Pablo - Axel |
| 9/4/22 | Se crean listas enlazadas en C y Java, además métodos para parsear JSON. | Jan – Pablo - Axel |
| 10/4/22 | Documentación Interna | Jan – Pablo - Axel |
| 10/4/22 | Creación de manual y documentación técnico | Jan – Pablo - Axel |

* 1. Descripción de las funciones implementadas.

***Parte de programación C:***

**Función makeRequest**:

La función makeRequest es la parte del cliente de c, donde envia y recibe json en forma de mensaje, con un formato de (Jugador,posx,km,disparo).

**Función gameInit:**

En la función gameInit se define todo sobre la ventana de juego como las texturas de fondos y jugadores, mediante una librería llamada SDL2.

Crea la lógica del juego con posiciones en pantalla del jugador y actualizando el fondo.

**Función mainWindow**:

En la función mainWindow se crea los eventos, la ventana, se renderiza los fondos de menú, textos, donde recibe listeners de enter.

**Función selectionMenu:**

En la función selectionMenu se renderiza las texturas del fondo del menú de selección y de los jugadores que se pueden escoger esperando respuesta mediante un keylistener o mouseclick.

**Función addToList:**

Agrega un nuevo elemento a la lista.

**Función crearLista:**

Crea una nueva lista.

**Struct JugadoresLista:**

Nodo de la lista enlazada de jugadores.

***Parte de programación en Java:***

**Clase EventManager:**

En la clase de EventManager se manejan funciones como agregar turbo, disparo, vida, hueco, donde que si se llaman se activa, en un kilómetro especifico.

Tiene función de getJsonEvent donde recibe un resultado en forma de json.

**Clase Disparo:**

En la clase disparo se crea un constructor del objeto donde se define un booleano activo y un flotante kilometro para poder manejar los datos dentro del juego.

Se crea función de set y get activo, set y get kilómetro.

**Clase Hueco:**

En la clase hueco se crea un constructor del objeto donde se define un booleano activo y un flotante kilometro para poder manejar los datos dentro del juego.

Se crea función de set y get activo, set y get kilómetro.

**Clase Turbo:**

En la clase turbo se crea un constructor del objeto donde se define un booleano activo y un flotante kilometro para poder manejar los datos dentro del juego.

Se crea función de set y get activo, set y get kilómetro.

**Clase Vida:**

En la clase vida se crea un constructor del objeto donde se define un booleano activo y un flotante kilometro para poder manejar los datos dentro del juego.

Se crea función de set y get activo, set y get kilómetro

**Clase Ventana:**

Se define y renderiza las partes de la ventana que controla el servidor para activar ciertos ventajas o desventajas dentro del juego. Con la función Ventana() se renderiza imágenes y botones.

Función comprobar() se comprueba el resultado que se llega en string. Y con el actionPerformed()

Se recibe una acción de click en un botón para ver cual acción implementar en el juego.

**Clase Servidor:**

Mediante la función de iniciarServidor() se comienza la conexión abierta del servidor con los clientes.

**Clase LinkedList:**

**Funcion actualizarJugador:**

Actualiza la información de un jugador determinado.

**Funcion addJugador:**

Agrega un nuevo jugador a la lista.

**Función addNode:**

Agrega un nuevo nodo a la lista.

**1.2. Descripción de las estructuras de datos desarrolladas.**

Listas enlazadas: Se utilizaron las listas enlazadas para poder controlar los datos del jugador: posX, Km, Disparo. Se utilizaron en java y en C en forma de json, para poder enviarlos entre cliente y servidor y manipularlos.

**1.3. Descripción detallada de los algoritmos desarrollados.**

Uno de los algoritmos más importantes desarrollados en el proyecto, fue el uso de jsons y listas enlazadas, estas son el motor principal de todo el juego, dado que inicialmente el cliente manda su estado (vida, puntos, posición) al servidor, el cual lee este json y lo carga a una clase EventManager, pera luego poder enviarle esta información a todos los usuarios. Por otro lado, está la parte en C, que recibe este json con la información de todos los usuarios y los eventos como donde hay un turbo, una vida o un hueco, esto para saber donde dibujarlo en el mapa, este json se carga a una lista enlazada para poder manejar de forma más fácil el dibujado de los otros jugadores, realizando un parseo por partes para cargarlo correctamente.

**1.4. Problemas sin solución: En esta sección se detalla cualquier problema que no se ha podido solucionar en el trabajo.**

Uno de los problemas sin solución fue poder renderizar un fondo y el icono con una resolución que se quería, se tuvo que bajar para que sirviera para que funcione para todos los jugadores.

**1.5. Plan de Actividades realizadas por estudiante: Este es un planeamiento de las actividades que se realizarán para completar la tarea, este debe incluir descripción de la tarea, tiempo estimado de completitud, fecha de entrega y responsable a cargo.**

|  |  |
| --- | --- |
| Roles Asignados | Participante |
| Manejo de listas enlazadas y servidor-cliente | Axel |
| Manejo de Json y conexión servidor-cliente | Jan |
| Creación de GUI y lógica del juego | Pablo |

|  |  |  |
| --- | --- | --- |
| Fecha | Actividades | Participantes |
| Semana1 – 2 dias | Hacer reunión e investigar sobre la parte lógica y grafica de C y Java | Axel – Pablo - Jan |
| Semana1 – 2 dias | Empezar a crear conexiones de Java a C | Axel - Jan |
| Semana 1 – 3 dias | Creación de lógica de juego poleposition. | Pablo - Jan |
| Semana 2 - 2 dias | Crear GUI con menu de selección. | Pablo |
| Semana 2 – 2 dias | Documentacion interna y externa. | Axel – Pablo - Jan |
| Semana 2 – 3 dias | Arreglar errores y mejorar condiciones | Axel – Pablo - Jan |

* 1. **Conclusiones y Recomendaciones del proyecto.**

Como principales conclusiones, se desprende que se logró aprender el manejo de listas enlazadas y jsons en Java y C de manera exitosa, ya que se pudo enviar correctamente los datos entre servidor-cliente.

* 1. **Bibliografía consultada en todo el proyecto.**

chuidiang.org. (2023, 3 23). Socket entre C y java. Retrieved from Socket entre C y java:

http://www.chuidiang.org/java/sockets/cpp\_java/cpp\_java.php

retrogames.cz. (2022, 03 23). retrogames. Retrieved from retrogames:

http://www.retrogames.cz/play\_064-Atari2600.php

Wikipedia. (2022, 03 23). Pole Position (videojuego). Retrieved from

https://es.wikipedia.org/wiki/Pole\_Position\_(videojuego