

Instituto Tecnológico de Costa Rica

Área Académica de Ingeniería en Computadores

Programa de Licenciatura en Ingeniería en Computadores

Curso: CE-3104 Lenguajes, Compiladores e intérpretes



PolePositionCR

Tarea Corta #3

Realizado por:

Alexis Gavriel Gómez	2016085662
Emmanuel Aguilar Sánchez	2016009338
Crisptofer Fernández Fernández	2016048367

Profesor:

Marco Rivera

Fecha: Cartago, Junio 18, 2018

1.1 Manual De Usuario	3
• Requerimientos para usar el software:	3
• Guía de uso:	3
Una vez que se ha conectado se debe seleccionar el color	7
1.2. Descripción de las estructuras de datos desarrolladas.	10
1.3. Descripción detallada de los algoritmos desarrollados.	10
1.4. Problemas conocidos.	11
1.5. Actividades realizadas por estudiante.	11
1.6. Problemas encontrados.	12
1.7. Conclusiones y Recomendaciones del proyecto.	12
1.8. Bibliografía consultada en todo el proyecto	13
2. Bitácora en digital.	14

1.1 Manual De Usuario

El software es una versión del popular juego Pole Position el cual, consiste; en su versión clásica en completar una carrera la cual definirá las posiciones de los corredores en la carrera por el campeonato mundial. En la versión clásica el juego sólo podía ser utilizado por un jugador a la vez.

Esta versión de Pole Position, conserva muchas de las características del juego clásico, incorporando nuevos elemento en su jugabilidad. Estos son las posibilidad de que varios jugadores en la misma partida y agrega la posibilidad de que el dueño de la pista puede agregar huecos en la pista que bajan la velocidad de los jugadores si estos pasan sobre estos.

- **Requerimientos para usar el software:**

Todos los cliente y el dueño de la pista deben estar conectados en la misma red local.

Para el caso del dueño de la pista, debe correr su programa en el sistema operativo Linux, en cualquiera de sus muchas distribuciones.

Para el caso de los jugadores, para correr el aplicativo, se debe tener instalado java 8, y un compilador de java. Se deben tener agregar las librerías de LWJGL 2.9.3 y Slick-Util para poder ejecutar la aplicación.

- **Guía de uso:**

Dueño de la pista

Cuando el dueño de la pista desee ejecutar su aplicación, deberá buscar la carpeta en donde se tiene guardado el juego. Deberá abrir la carpeta “ServerII”. Una vez ahí, deberá abrir una terminal en esa carpeta, para ello, deberá dar click derecho en cualquier parte y seleccionar “Abrir en terminal”.

Una vez dentro de la terminal de Linux, escribir “make” en la terminal para compilar el programa. Cuando termina de compilar, deberá escribir “./output” para ejecutar el juego.

Una vez ejecutado el juego, el dueño de la pista podrá enviar “turbos” o “huecos”...

```
ema0898@ema0898-Inspiron5559: ~/Programas/C/ServerII
Archivo Editar Ver Buscar Terminal Ayuda
ema0898@ema0898-Inspiron5559:~/Programas/C/ServerII$ make
gcc -c main.c
gcc -c server/server.c
gcc -c logic/logic.c
gcc main.o server.o logic.o -lpthread -o output
ema0898@ema0898-Inspiron5559:~/Programas/C/ServerII$ ./output
Socket created
bind done
Waiting for incoming connections...
Ingrese el obstaculo que desee agregar a la pista: 
```

Cuando el dueño de la pista haya ejecutado el programa, le aparecerá un pequeña interfaz en la cual, podrá ingresar el obstáculo y mejora que desee.

```
ema0898@ema0898-Inspiron5559: ~/Programas/C/ServerII
Archivo Editar Ver Buscar Terminal Ayuda
ema0898@ema0898-Inspiron5559:~/Programas/C/ServerII$ make
gcc -c main.c
gcc -c server/server.c
gcc -c logic/logic.c
gcc main.o server.o logic.o -lpthread -o output
ema0898@ema0898-Inspiron5559:~/Programas/C/ServerII$ ./output
Socket created
bind done
Waiting for incoming connections...
Ingrese el obstaculo que desee agregar a la pista: turbo,45,45
Ingrese el obstaculo que desee agregar a la pista: 
```

Si desea agregar un turbo a los jugadores, deberá escribir “turbo” seguido de la posición donde quiere que aparezca, todo esto seguido por separado por comas. En este caso debería ser “turbo,45,45”.

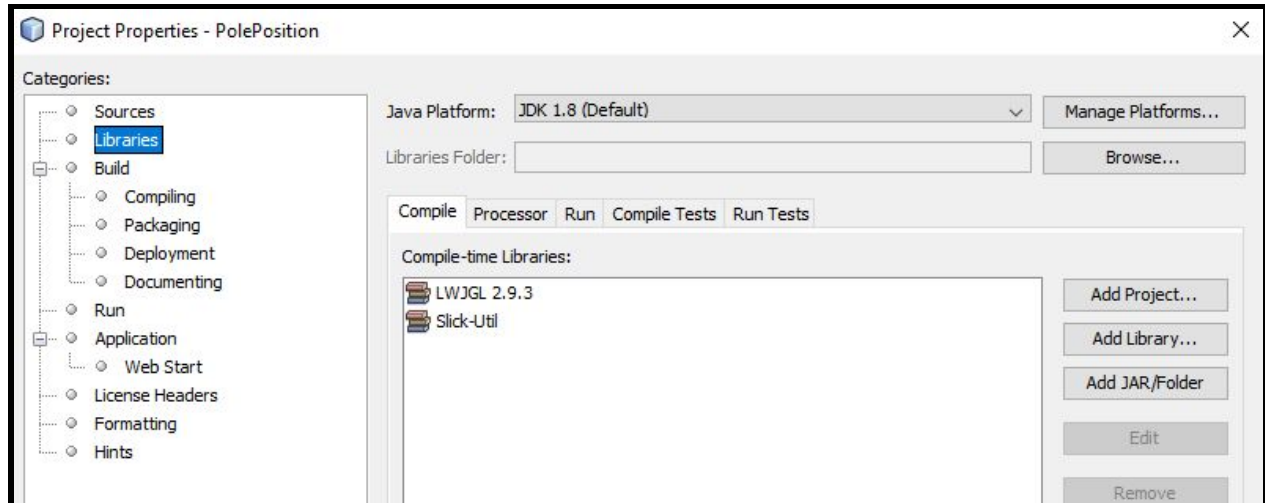
```
ema0898@ema0898-Inspiron5559: ~/Programas/C/ServerII
Archivo Editar Ver Buscar Terminal Ayuda
ema0898@ema0898-Inspiron5559:~/Programas/C/ServerII$ make
gcc -c main.c
gcc -c server/server.c
gcc -c logic/logic.c
gcc main.o server.o logic.o -lpthread -o output
ema0898@ema0898-Inspiron5559:~/Programas/C/ServerII$ ./output
Socket created
bind done
Waiting for incoming connections...
Ingrese el obstaculo que desee agregar a la pista: turbo,45,45
Ingrese el obstaculo que desee agregar a la pista: hueco,34,34
Ingrese el obstaculo que desee agregar a la pista: 
```

Si desea poner un hueco a los jugadores, deberá escribir “hueco” seguido de la posición donde quiere que aparezca, todo esto seguido por separado por comas. En este caso debería ser “hueco,34,34”.

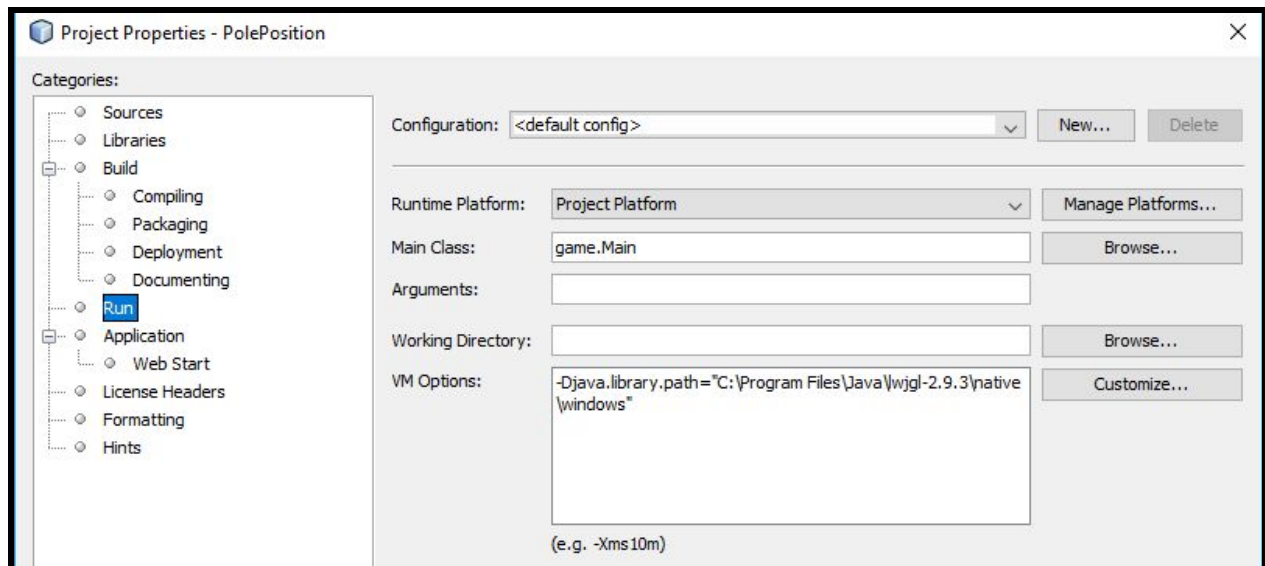
Jugador

Se debe abrir el proyecto en un compilador de Java, revisar que las librerías estén agregadas correctamente y que el proyecto se corra bajo algunos argumentos.

Librerías:



Configuración para correr:

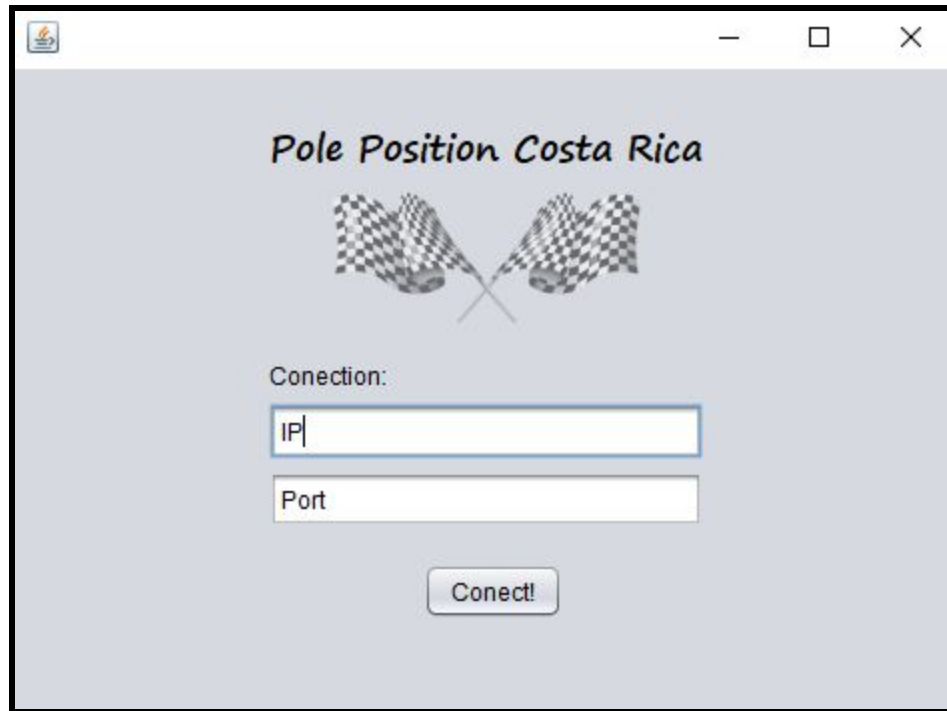


El argumento es: `-Djava.library.path="dir"`

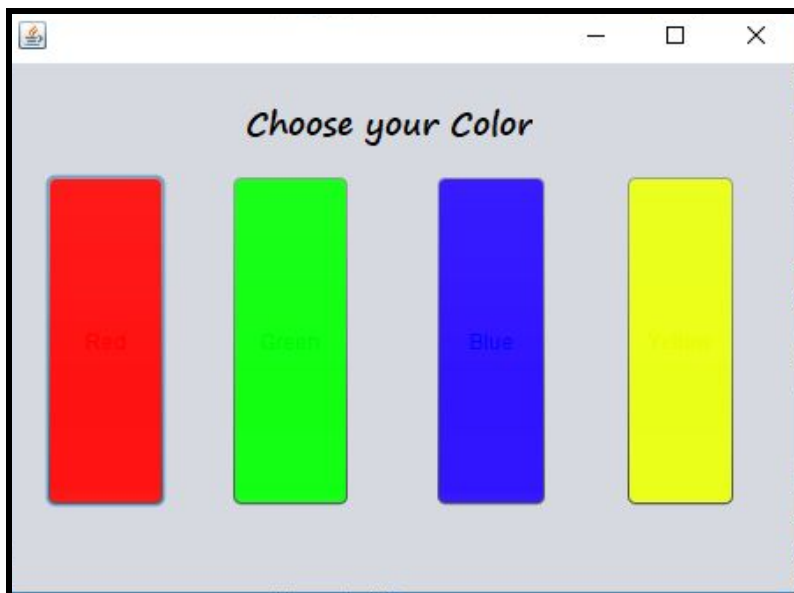
En donde "dir" es la dirección en donde se encuentra la carpeta "\\native\\OS " dentro de la carpeta de la librería LWJGL.

En donde OS es el sistema operativo que se utiliza para correr el cliente.

Para correr el programa se debe correr el “Main.java” que se encuentra dentro del paquete “game”. Una vez iniciado el programa se deberá ingresar la IP y el Puerto en donde se encuentra el servidor (se debe borrar las palabras “IP” y “Port”) seguido de hacer clic en el botón “Connect!”.

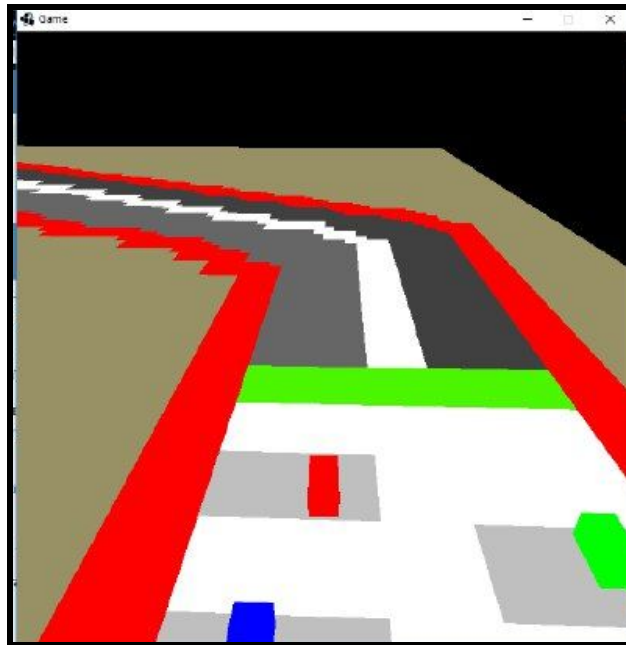


Una vez que se ha conectado se debe seleccionar el color



No se puede seleccionar un color previamente seleccionado por otro jugador.

En el juego:



Una vez en el juego el jugador se mueve utilizando “WASD” o las flechas. Para disparar se tecldea la barra espaciadora.

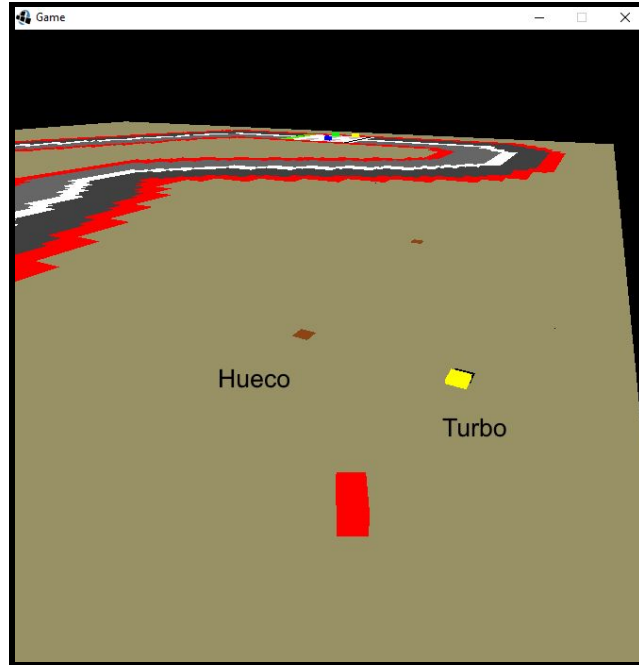
Cuando el jugador se intenta mover hacia adelante el carro empieza a acelerar, aumentando su velocidad poco a poco.

Si no se está intentando mover hacia adelante el carro intentará frenar, disminuyendo su velocidad poco a poco. Si se intenta retroceder se incrementará la tasa con la que la velocidad se reduce a 0.

Al teclear alguna de las direcciones izquierda o derecha se modifica el ángulo de adonde se mueve hacia adelante.

Cuando el jugador se encuentra en una posición de matriz que corresponde con un borde la velocidad baja a la mínima.

Si el jugador es disparado por otro jugador no podrá moverse ni disparar.



Si la posición en matriz del jugador corresponde a un hueco, la velocidad se disminuye a la mínima.

Si la posición en matriz del jugador corresponde a un turbo, la velocidad se incrementará, esta se irá reduciendo hasta llegar a la velocidad máxima definida.

1.2. Descripción de las estructuras de datos desarrolladas.

Listas enlazadas: Se utilizan listas doblemente enlazadas para almacenar y editar los disparos en el juego.

Struct: Se utilizan para guardar los atributos de los clientes en el servidor y posteriormente utilizarlas para los fines necesarios.

1.3. Descripción detallada de los algoritmos desarrollados.

Al ejecutar el cliente se pasa por varias etapas:

Conexión con el servidor: El cliente debe conectarse al servidor ingresando la IP y el puerto.

Selección de color: El cliente selecciona un color, el cual no puede ser repetido entre clientes.

Lectura de la matriz: el servidor envía la matriz del mapa de juego a los jugadores.

Inicialización de la interfaz gráfica: se crea la ventana de juego y los objetos del juego.

Ciclo de juego:

Actualizar: Se actualiza la información de cada uno de los objetos de juego que lo requieran.

- Cuando el jugador se intenta mover hacia adelante el carro empieza a acelerar, aumentando su velocidad poco a poco, y cuando no, el carro intentará frenar, disminuyendo su velocidad poco a poco.
- Cuando el jugador se encuentra en una posición de matriz que corresponde con un borde la velocidad baja a la mínima.
- Cuando el jugador se encuentra en una posición de matriz que corresponde con un hueco o turbo se realiza la modificación de la velocidad correspondiente.
- Revisión de colisiones: Se revisa si alguno de los vértices de un objeto de juego se encuentra dentro de el espacio delimitado por los vértices del segundo objeto.

Dibujar: Se dibujan los objetos de juego que lo requieran, a partir de su información.

Limpiar y Cerrar: Al cerrar la ventana del juego se detiene el ciclo de juego y luego se limpia la ventana.

1.4. Problemas conocidos.

- Existe un movimiento espontáneo de los jugadores. Estos se mueven algunas veces a posiciones distantes de la posición inicial.
- Si el servidor se cierra abruptamente el puerto de conexión queda abierto, por lo que hay que editar el código para que se abra de nuevo el servidor en un puerto distinto.
- No se logra reducir la vida del jugador cuando se detecta que ha recibido un disparo de otro jugador.
- Hay un mal manejo de recursos que provoca que el programa corra lentamente.

1.5. Actividades realizadas por estudiante.

Alexis: Lógica del juego y de la interfaz del juego

Crisptofer: Interfaz gráfica del juego y conexión del juego con el cliente.

Emmanuelle: Lógica del servidor y conexión servidor-cliente

1.6. Problemas encontrados.

Cuando el cliente se desconecta de forma abrupta, los puertos en el servidor no se cierran de forma correcta lo cual, produce errores a la hora de ejecutar el servidor. La solución a este problema consiste en cambiar el puerto de conexión lo cual implica cambiarlo tanto en el servidor como en el cliente.

El uso de cadenas de caracteres en C, puede ser difícil si no se realiza con cuidado, ya que puede generar errores de tipo “segmentation fault”. La forma de arreglarlo es hacer un buen uso de estos lo cual, evitará errores.

1.7. Conclusiones y Recomendaciones del proyecto.

Conclusiones

- OpenGL es una excelente librería que permite realizar juegos tanto en 2D como en 3D.
- Utilizar un servidor permite que varios clientes compartan información entre sí.
- El paradigma orientado a objetos es una gran herramienta para modelar una solución a un problema, permite una fácil modularidad.
- Utilizar el lenguaje de programación C para crear un servidor permite que se dé una respuesta rápida entre servidor y clientes.

Recomendaciones

- Utilizar de forma responsable las cadenas de caracteres en C
- Utilizar OpenGL para realizar juegos en 2.5D y 3D.
- Realizar funciones para revisión de colisiones óptimas.

1.8. Bibliografía consultada en todo el proyecto

B, E. (2010). Java an C++ socket communication. Retrieved from

<http://www.drdobbs.com/jvm/java-and-c-socket-communication/222900697>

H, J. (2015). Strtok reference. Retrieved from

<http://www.cplusplus.com/reference/cstring/strtok/>

Thebennybox Jul 2014, <https://www.youtube.com/playlist?list=PL513808FE7D9A5D68>

Stack Overflow. (2013). 2D Moving Camera (LWJGL). [online] Available at:

<https://stackoverflow.com/questions/18971836/2d-moving-camera-lwjgl> [Accessed 16 Jun. 2018].

Forum.lwjgl.org. (2013). How to move LWJGL's window after it has been created?.

[online] Available at: <http://forum.lwjgl.org/index.php?topic=5278.0> [Accessed 16 Jun. 2018].

2. Bitácora en digital.

Emmanuelle Aguilar		
Fecha	Actividad	Horas
2/6	Realizar el servidor en C y el cliente en Java	2
3/6	Corregir errores de conexión interpretación de mensajes de cliente	1
15/6	Enviar la matriz de juego y actualizaciones de la misma	2
16/6	Realizar el thread de los puntos, enviar códigos de mensaje al cliente e interpretación de dichos mensajes por el cliente	2.5
17/6	Terminar la documentación del programa, documentar el código en C y terminar de implementar el cliente con el resto de la interfaz de juego.	4

Crisptofer Fernández		
Fecha	Actividad	Horas
10/06/19	Investigación acerca de openGL y la librería LWJGL 2.9.3	4
15/06/19	Investigación para la implementación de texturas al juego y pruebas unitarias.	6
16/06/19	Implementación de los disparos al Jugador.	3
16/06/19	Integración de la clase Cliente que permite la conexión entre servidor y cliente.	7
17/06/19	Creación de las Interfaces de inicio del juego.	6

Alexis Gavriel		
Fecha	Actividad	Horas
2/6	Se crea la función que lee desde un archivo de texto la matriz y la traduce a una matriz de enteros.	1
3/6	Se implementó el caso base del jugador moviéndose alrededor del mapa y con la capacidad de disparar.	2
4/6	Se agregó la visualización del mapa con colores y se implementó la aceleración del jugador. Cuando el jugador está en una posición de matriz que corresponde con un borde la velocidad baja a la mínima.	3
11/6	Se recorta la matriz creando una submatriz con la que se dibuja el juego para así recortar la vista que el jugador tiene.	2
12/6	Se investiga sobre cómo dibujar en 3D y cómo trabajar con una cámara que despliega una imagen en 3D desde un punto de vista	2
14/6	Se investiga acerca de cómo obtener una submatriz más fácilmente recorriendo adecuadamente la original, sin copiarla.	2
15/6	Se logra dibujar la parte de la matriz alrededor del jugador, y que ésta se actualice cuando el jugador se mueve. Se decide descartar estos intentos para reemplazar la forma de obtener el punto de vista correcto, utilizando la clase Cámara.	5
16/6	Acomodo de la cámara y el movimiento. Se implementó las colisiones con otros jugadores	5
17/6	Documentación del proyecto y documentación interna del programa	3