



Instituto Tecnológico de Costa Rica Área
Académica Ingeniería en Computadores.
Lenguajes, Compiladores e intérpretes (CE3104)
Grupo 2

Tarea 3 - DonCEy Kong jr

Profesor:

Marco Rivera Meneses

Estudiantes:

Juan Pablo Carrillo

Sebastián Moya

Erick Madrigal

Lunes 26 de abril del 2021
I Semestre

1. Documentación.

1.1 Descripción de algoritmos.

- Algoritmo para la gravedad: para la parte gráfica del juego se desarrolló un algoritmo que simula el efecto de la gravedad, donde el algoritmo consiste en verificar si el jugador (Donkey Kong jr) se encuentra posicionado sobre una plataforma o en una liana, de no ser así, de ser así, el jugador no se ve afectado por la gravedad, en cambio, si no se cumple ninguna de estas dos condiciones, el jugador sufre los efectos de la gravedad y se le obliga a empezar a caer hasta que se agarre de una liana o caiga sobre alguna plataforma.

1.2 Funciones implementadas.

Java

Servidor:

solicitarAccion (mensaje): Valida el mensaje enviado por el cliente y retorna la respuesta correcta, recibe el mensaje y retorna la respuesta.

iniciar(): crea el socket que conecta con el cliente. Retorna el socket.

run(): inicia la comunicación con el cliente.

analizarPiezas (mensaje): analiza mensajes compuestos enviados por el cliente y retorna la respuesta. Recibe un string con el mensaje compuesto.

C

jugador1(SOCKET s)/jugador2(SOCKET s): funciones encargadas de crear el loop de juego y llamar a las funciones de actualización

gravedad(Junior* mono,SOCKET s)/gravedad2(Junior* mono,SOCKET s): función encargada de crear un efecto de gravedad en el mono y hacer que este baje si no está en una liana o plataforma.

void cargarCrocs1(SOCKET s) / void cargarCrocs2(SOCKET s): Función que pide la lista de cocodrilos al servidor y los agrega al array de cocodrilos.

colidCoc(Junior* mon,Croco* croc,SOCKET s) / colidCoc2(Junior* mon,Croco* croc,SOCKET s): Función que detecta si el mono y algún cocodrilo se tocaron.

colitLiana(int x,int y): Funcion que compara la posición en x del mono para saber si está en contacto con una liana

colitPlat(int x,int y): Función que compara la posición en x y la posición y del mono para saber si está en contacto con una plataforma.

moverCrocs(Croco* croc):función que actualiza la posición del cocodrilo enviado según el tipo de cocodrilo que sea.

removeCrocs():función que libera el espacio en memoria utilizado por cada cocodrilo

lifes(SOCKET s) / lifes2(SOCKET s): Función que le pide al servidor la cantidad de vidas del mono.

posmono(Junior *mon,SOCKET s) / posmono2(Junior *mon,SOCKET s):Funcion que actualiza la posición del mono según los datos obtenidos del servidor

crearSocket(): crea el socket del cliente y lo retorna

enviar(Socket s, char *mensaje, char* respuesta): envía el mensaje al servidor y retorna la respuesta. Recibe el socket, el mensaje que se desea enviar y un char donde se guardará la respuesta.

Partida:

getPosJugador(): Obtiene la posición en x y y del jugador, separado por punto y coma.

getFruta(java.lang.Integer numFrut): Método que devuelve una fruta en específico.

getCrocodile(java.lang.Integer numCroc): Método que retorna un cocodrilo en específico.

moverDere(): Método que devuelve la posición del jugador en x aumentada en 10

moverIzq(): Método que devuelve la posición del jugador en x decrementada en 10

moverArriba(): Metodo que devuelve la posición del jugador en y aumentada en 10

moverAbajo(): Método que devuelve la posición del jugador en y decrementada en 10

desconEspec(): desconecta un espectador de la partida.

getFrutas(): Método que obtiene todas la fruta y sus atributos y los convierte en string

getCocodrilos(): Método que obtiene todas los cocodrilos y sus atributos y los convierte en string

getJugador(): Método que obtiene los atributos del jugador

eliminarCroc(java.lang.Integer numCroc): Método que elimina un cocodrilo por su id

eliminarFruit(java.lang.Integer numFrut): Método que elimina las frutas por su id

attackCroc(): Método que le elimina una vida del jugador al ser atacado por un cocodrilo y lo devuelve al inicio

crearCocodrilo(java.lang.Integer posx,java.lang.Integer posy,java.lang.String

tipo,java.lang.Integer speed): Método que crea un nuevo cocodrilo en la partida

reset(): Método que resetea el juego

win(): Método que aplica cuando gana el jugador, le da una vida y lo devuelve al inicio.

Player:

modificarPosX(java.lang.Integer pasos): Función que aumenta la posición en x del jugador

modificarPosY(java.lang.Integer pasos): Función que aumenta la posicion en y del jugador

addPoints(java.lang.Integer points): Función que aumenta los puntos del jugador por cada fruta obtenida.

loseLifes(): Función encargada de eliminar vidas al jugador.

aumentarVidas(): Da una vida extra al jugador.

1.3 Estructuras de datos.

- Arrays: Se implementaron arrays para organizar el manejo de los jugadores, específicamente para el almacenamiento de las partidas generadas para cada jugador. También se creó un arreglo para almacenar los structs de los cocodrilos en la interfaz
- Listas: Se implementaron listas para organizar el manejo de las instancias creadas en cada partida, específicamente para realizar el almacenamiento de las frutas y los cocodrilos de cada partida en el servidor.
- Structs: Se utilizaron structs principalmente para definir las características de los objetos dinámicos del mapa de juego, como lo es el jugador principal (Donkey kong jr), y los cocodrilos que van apareciendo .

1.4 Problemas sin solución.

- Los cocodrilos del jugador y los del observador pueden ir descoordinados.

1.5 Problemas encontrados.

- Durante el desarrollo de la interfaz se producía el error "Multiple definition of 'function' ". Esto se producía porque se estaba importando el archivo .c que contenía la función directamente. Se solucionó creando un archivo .
- Al realizar la ejecución de la función gravedad, esta ocasiona que el servidor se cierre. Esto se corrigió creando un delay entre peticiones
- Hubo un problema a la hora de compilar la librería Allegro por lo que se cambio el juego a la libreria SDL2

1.6 Plan de actividades realizadas por estudiante:

Descripción de la tarea	Tiempo estimado de completitud (horas)	Responsable a cargo	Fecha de entrega
Identificación y definición de clases para el juego en Java	2	Juan Pablo	13/04/2021
Creación de clases para el juego en Java	5	Juan Pablo	15/04/2021
Creación de función para que el cliente en C puede usar los datos enviados por el servidor	2	Juan Pablo	20/04/2021
Búsqueda de librería para la interfaz	3	Erick	18/04/2021
Creación de una interfaz	10	Erick	22/04/2021
Conexión del socket cliente	4	Sebastián	22/04/2021
Creación del servidor multicliente	4	Sebastián	22/04/2021
Conexión del servidor con la lógica del juego	2	Sebastián	22/04/2021

1.7 Conclusiones:

- Se logró realizar un proyecto donde se unificaron dos paradigmas de programación distintos, implementando por una parte el paradigma de programación imperativo, con el lenguaje de programación C y por otra parte el paradigma de programación orientado a objetos, utilizando Java, cumpliendo con los debidos requisitos de cada paradigma, logrando realizar una conexión entre las dos partes independientes del proyecto mediante la ayuda de Sockets.
- A la hora de trabajar con Sockets utilizando dos lenguajes distintos, es fundamental conocer bien cómo se deben realizar las peticiones, su debida respuesta y la estructura de las dos, debido a que en Java el manejo de strings y de datos en general que son enviados mediante los sockets se puede realizar de una manera más simple, mientras que por parte de C, al tener que usar arreglos de caracteres puede volver más complicado el manejo de los datos recibidos y enviados a través de los sockets.

1.8 Recomendaciones:

- A la hora de trabajar con una interfaz gráfica en C, es importante investigar y definir una buena librería para la implementación del apartado gráfico, debido a que algunas librerías populares funcionan muy bien en C++, pero a la hora de querer usarlas también en C, puede que se presenten algunos errores inesperados o incluso que el programa no llegué a compilar. Por eso es de suma importancia tener referenciada una buena librería de la que se pueda tener confianza para trabajar con ella en futuros proyectos similares.

1.9 Bibliografía consultada en todo el proyecto.

SDL Wiki. (s. f.). SDL Wiki. <https://wiki.libsdl.org/>

Singh, C. (2017, 24 septiembre). *C - Strings and String functions with examples.* Beginnersbook.Com.

<https://beginnersbook.com/2014/01/c-strings-string-functions/>

chuidiang.org. (2020, 3 21). Socket entre C y java. Retrieved from Socket

entre C y java:

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

2. Bitácora con el trabajo realizado durante el proyecto por cada miembro del equipo.

Fecha	Trabajo realizado por Sebastián(horas)	Trabajo realizado por Erick (horas)	Trabajo realizado por Juan Pablo (horas)
09/04/2021			
10/04/2021			
11/04/2021	Lectura de la especificación de manera grupal (0.5)	Lectura de la especificación de manera grupal (0.5)	Lectura de la especificación de manera grupal (0.5)
12/04/2021		Investigación sobre librerías para GUI en C (2)	
13/04/2021		Investigación sobre la librería SDL(2)	Identificación y definición de clases para el juego en Java (1.5)
14/04/2021	Investigación sobre sockets en JAVA (2)		Creación de clases para el juego en Java (2.5)
15/04/2021	Creación de sockets en JAVA (3)	Investigación de instalación de SDL(2)	Creación de clases para el juego en Java (2)
16/04/2021		Investigación sobre gtk en C (1,5)	
17/04/2021	investigación de sockets en C (2)		
18/04/2021	Creación de sockets en C (2,5)	Investigación sobre la librería Allegro(2)	Investigación y creación de funciones para que el cliente en C reciba los datos del server. (3)
19/04/2021	Reestructura del servidor/sockets cliente-servidor(4)	Creación de estructuras iniciales para la interfaz(2)	Investigación y creación de funciones para que el cliente en C reciba los datos del server. (1)
20/04/2021		creación de ventanas en allegro(3)	
21/04/2021		creación de los eventos principales(2)	Investigación sobre la librería a usar para trabajar con la parte de la interfaz en C.(3)
22/04/2021	creación de condiciones de mensajes y encapsulación del cliente (3)	creación de las colisiones(2)	Uso de la librería para crear un menú inicial para el juego (2)
23/04/2021	Creación del menu en la interfaz(1,5)	Solución de errores en la interfaz y la definición de los datos	Investigación sobre otras librerías a usar para trabajar con la parte de la

			interfaz en C.(2)
24/04/2021		Búsqueda de solución de errores de compilado de Allegro (3)	
25/04/2021	Corrección de errores en el servidor (2)	Investigación sobre sdl y creación de funciones de choque y la interfaz en sdl (10)	
26/04/2021	Crear función para interpretar los datos del servidor en C (4)	Conección de la interfaz con el servidor (7)	
27/08/2021	Cambio de las respuestas del servidor(2)	Creación de la pantalla del espectador y conexión con el servidor (10)	
27/08/2021		Resolución de errores(5)	

Reuniones realizadas:

Fecha: 11/04/2021

Se analizó entre todos la especificación del proyecto. Se identificó los temas sobre los cuales se debería investigar para la correcta realización del proyecto.

Fecha: 12/03/2021

Se realizó la identificación y posteriormente la distribución de tareas para cada integrante a completar para desarrollar el proyecto con éxito. Además se definieron los plazos estimados para cumplir con dichas tareas y se establecieron algunos otros aspectos técnicos. Esta reunión fue principalmente para analizar el proyecto y organizarnos como equipo de trabajo.

Fecha: 18/04/2021

Se realizó una reunión para presentar los avances que cada uno llevaba hasta el momento y para comunicar al grupo las tareas completadas con éxito y las tareas en las cuales se presentaron problemas o algún inconveniente inesperado.

Fecha 20/04/2021

Se realizó una reunión para definir entre todos los integrantes del grupo como realizar la conexión entre la parte del cliente en c, con el servidor en Java e identificar las partes que fueran necesarias adaptar o modificar por parte del cliente o servidor para que los 2 puedan comunicarse de forma correcta.

Fecha: 22/04/2021

Se realiza una reunión debido a que se presentaron unos problemas con la librería usada para la interfaz del cliente en c, entonces nos reunimos todos para decidir que hacer y buscar posibles soluciones para seguir trabajando con esta librería o evaluar de ser necesario hacer un cambio a otra librería.

Fecha: 25/04/2021

Nos reunimos para realizar unas últimas pruebas al proyecto que vamos a presentar