Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Computación

Área de Ingeniería en Computadores

Compiladores, Lenguajes e Intérpretes

Tarea Programada #3: DonCEy Kong Jr

María José Zamora Vargas 2018026607

Erick Barrantes Cerdas 2017125821

Jose Acuña Solano 2018145020

Grupo 02

Fecha de entrega: 22 Septiembre 2019

Contents

[Descripción del problema 3](#_Toc20087990)

[Descripción de las estructuras de datos utilizadas. 4](#_Toc20087991)

[Descripción detallada de las funciones desarrolladas ordenadas por clases. 5](#_Toc20087992)

[Problemas conocidos. 8](#_Toc20087993)

[Problemas encontrados. 8](#_Toc20087994)

[Plan de Actividades. 9](#_Toc20087995)

[Conclusiones & Recomendaciones. 10](#_Toc20087996)

[Bibliografía. 11](#_Toc20087997)

[Bitácora 12](#_Toc20087998)

# Descripción del problema

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Desarrollar una aplicación que simule el funcionamiento del juego Donkey Kong Jr. Este es una secuela del videojuego arcade, Donkey Kong, fue lanzado en 1982 y esta vez Mario tiene prisionero a Donkey Kong y su hijo un pequeño gorila llamado Donkey Kong Jr tiene que rescatarlo. En la primera escena Jr, tiene que ir trepando por lianas y caminando por plataformas evitando trampas animadas que le arroja Mario (parecidos a los kremlings, enemigos de los juegos de Donkey Kong Country), hasta colocarse en una plataforma al lado de Donkey Kong.



# Descripción de las estructuras de datos utilizadas.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* Listas

En esta aplicación se utilizaron para el manejo de los enemigos, es decir, de los cocodrilos ya que estos pueden crecer dinámicamente en cantidad durante la ejecución del programa.

* Arrays.

Los arreglos fueron implementados para almacenar las plataformas y cuerdas creadas en cada nivel. También, un array de bitmaps el cual se recorre para cambiar el sprite del avatar y que concuerde con su movimiento.

# Descripción detallada de las funciones desarrolladas ordenadas por clases.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* CollisionHandler

1. isCollidingWithPlatform

Entradas: objeto Junior, objeto Plataforma

Salidas: 1/0

Compara las coordenadas en (x,y) entre el avatar Junior y todas las plataformas, si estas colisionan retorna un 1 equivalente a TRUE y 0 para FALSE.

1. isCollidingWithRope

Entradas: objeto Junior, objeto Cuerda

Salidas: 1/0

Compara las coordenadas en (x,y) entre el avatar Junior y todas las cuerdas, si estas colisionan retorna un 1 equivalente a TRUE y 0 para FALSE.

1. isCollindingWithCoco

Entradas: objeto Junior, objeto LinkedList

Salidas: 1/0

Compara las coordenadas en (x,y) entre el avatar Junior y una lista de cocodrilos, si estos colisionan retorna un 1 equivalente a TRUE y 0 para FALSE.

* GameWindow

1. initializeWidgets

Entradas: objeto Allegro Display

Salidas: void

Inicializa los widgets del juego, crea el objeto Junior, Plataforma y Cuerdas. También, inicializa la lista de cocodrilos.

1. createPlatforms

Entradas: ninguna

Salidas: ninguna

Creación de n plataformas en la pantalla del juego.

1. createRopes

Entradas: ninguna

Salidas: ninguna

Creación de n cuerdas en la pantalla del juego.

1. gameLoop

Entradas: ninguna

Salidas: entero, variable win

Mantiene el hilo principal de juego en ejecución mientras el jugador no gane o bien, mientras la variable playing sea TRUE.

1. eventManager

Entradas: ninguna

Salidas: 1/0

Manejo de eventos para el display mediante un queue de eventos.

1. redrawDisplay

Entradas: ninguna

Salidas: ninguna

Re-dibuja los gráficos de la pantalla con sus coordenadas actuales.

1. deleteWidgets

Entradas: ninguna

Salidas: ninguna

Destruye y libera la memoria de todas las entidades instanciadas.

* Croco

1. moveCroco

Entradas: objeto Croco

Salidas: ninguna

Se encarga del movimiento de los cocodrilos ya sea azul o rojo.

* Entity

1. updateRPoss

Entradas: objeto Entidad

Salidas: ninguna

Actualiza las coordenadas en (x,y) de la entidad suministrada.

# Problemas conocidos.

* Observadores

Problema: No se implementaron los clientes observadores por problemas con múltiples hilos y cambios de lógica de cliente jugador a observador.

# Problemas encontrados.

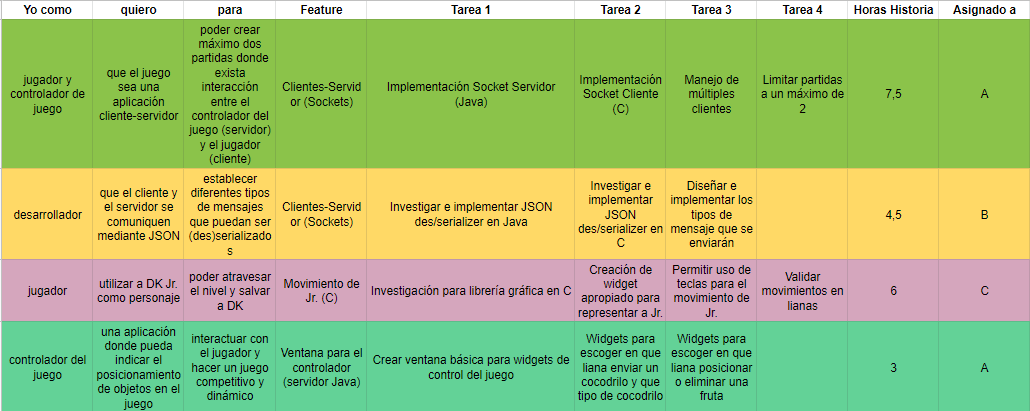
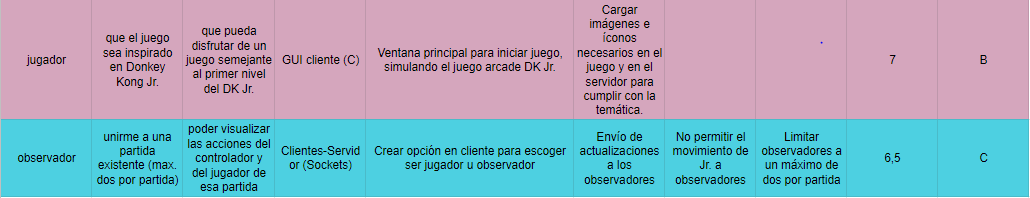
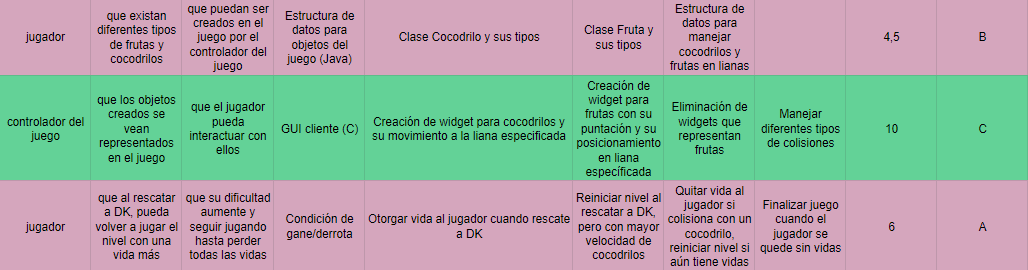
* Sockets

Problema: los sockets no mantenían un flujo dinámico de datos, lo que provocaba que a veces los mensajes llegaran incompletos o no llegaran en su totalidad.

Solución: Cerrar el socket después de que el cliente envía un mensajes y recibe una respuesta.

# Plan de Actividades.

Para el planeamiento y análisis previo al desarrollo del proyecto, se hizo uso del método de historias de usuario, posteriormente se dividieron en tareas con sus respectivas horas, un encargado a desarrollarla. En la bitácora se puede observar el avance de las tareas.



# Conclusiones & Recomendaciones.

1. Conclusiones

La versión final del proyecto se compone de una interfaz grafica funcional desarrollada en C y conectada a través de sockets con un servidor en Java. Aparte de esto, se logró implementar una simulación funcional del juego DonCEy Kong Jr.

Gracias a este proyecto pudimos aplicar en la práctica los paradigmas funcional y orientado a objetos. Debido a que la comunicación entre sockets se realiza por un formato que estos dos lenguajes admiten, se facilita la conexión de los paradigmas en una aplicación cliente-servidor que emplea ambos.

1. Recomendaciones

Consideramos que para desarrollo de interfaz específicamente, el proceso en un lenguaje como C es más largo que en otras bibliotecas de otros lenguajes. Si se quisiera mantener la flexibilidad de lenguaje mientras se maneja una GUI más amigable podríamos tomar en cuenta el desarrollo de la aplicación con C++ Qt Creator.

# Bibliografía.

Allegro 5 reference manual. (2019, Septiembre 22). Recuperado Septiembre 16, 2019, de <https://liballeg.org/a5docs/trunk/>

Socket Programming in C/C++. lcmgcd, MichaelThomasKloos. GeeksforGeeks. Tomado el 13 Set 2019 de <https://www.geeksforgeeks.org/socket-programming-cc/>

Winsock tutorial – Socket programming in C on windows. Silver Moon (April 23, 2019). BinaryTides. Tomado el 19 Set 2019 de <https://www.binarytides.com/winsock-socket-programming-tutorial/>

Getting started guide. Allegro Wiki. Tomado el 19 Set 2019 de <https://liballeg.org/a5docs/trunk/getting_started.html>

API udp/json-parser. James McLaughlin. Tomado el 20 Set 2019 de <https://github.com/udp/json-parser>

API udp/json-builder. James McLaughlin. Tomado el 20 Set 2019 de <https://github.com/udp/json-builder>

# Bitácora

María José Zamora Vargas

Lunes 16 Septiembre:

Planteamiento de lógica e interfaz en Java y C. Investigación sobre manejo de sockets y GUI en C. Inicio de pantallas en GTK, aprendiendo sobre la biblioteca y cómo utilizarla.

Miércoles 18 Septiembre:

Reunión de equipo. Investigación sobre bibliotecas para GUI en C. Cambio de biblioteca GUI por Allegro, más adecuada para juegos. Instalación de Allegro y comienzo de set-up.

Jueves 19 Septiembre:

Investigación sobre Allegro, comienzo de pruebas. Continuación de la documentación técnica.

Viernes 20 Septiembre:

Error en CMake con biblioteca Allegro, investigación para soluciones. Continuación de la documentación técnica.

Sábado 21 Septiembre:

Desarrollo de pantalla de inicio, input de dirección. Problema con CMake y Allegro. Continuación de documentación técnica e inicio de manual de usuario.

Erick Barrantes Cerdas

Viernes 13 de septiembre:

Planificación del proyecto con historias de usuario y tareas

Miércoles 18 de septiembre:

Se instala GTK y se inicia una ventana inicial con esta librería. Luego de una mayor investigación, se reemplaza GTK por Allegro.

Jueves 19 de septiembre:

Creación de ventana de juego, de struct y sprite para DK Jr., struct para Platform.

Loop básico del juego a 60FPS.

Movimiento horizontal de DK Jr.

"Gravedad" del juego es aplicada para DK Jr.

Se realizan colisiones de Jr. con plataformas para que Jr. camine sobre ellas

Diseño del juego se realiza con variables globales implementadas en var.h

Viernes 20 de septiembre:

Movimiento de salto para Jr.

Arreglo de bugs en colisiones con plataformas.

Muerte de Jr. cuando cae de la ventana de juego.

Cambio de tamaño en plataformas.

Implementación de Ropes en el juego y su colisión con estas, un segundo tipo de plataformas con otro sprite, movimiento de Jr. en Ropes.

Sábado 21 de septiembre:

Se termina diseño del juego.

Se modula más el código de la ventana de juego.

Arreglos de múltiples bugs en colisiones de Jr. con ropes y platforms, también en movimientos en las ropes.

Creación de lista enlazada para Crocs.

Domingo 23 de septiembre:

Se agrega condición para terminar el nivel y que sea posible volver a jugarlo.

Se agregan sprites para los crocos.

Implementación del movimiento para los dos tipos de crocos, de frutas con su sprite y su colisión con estas.

Colisión con crocos y Jr. es implementada, el juego termina si esto sucede.

Lista enlazada es genérica, se usa tanto para frutas como para crocs.

Documentación de código.

Jose Acuña Solano

Viernes 13 Set:

Planeamiento de la estructura del servidor en Java, interfaz grafica del servidor

Crea estructura básica del socket para las clases Server y Cliente, una interfaz para el servidor, widgets customizados para hacer la interfaz grafica dinámica

Miércoles 18 Set:

Creado estructura de las Entidades

Se implementa el Server a la interfaz por medio de un Thread y el Cliente al juego

Jueves 19 Set:

Agrega allegro al proyecto

Reestructuración de los sockets para que se cierren al terminar cada petición

Sockets funcionales con varios clientes

Viernes 20 Set:

Serialización y parseo de JSON

Se agrega un ID a cada cliente, posibilitando pestañas individuales en el server

Server puede espectar clientes

Domingo 22 Set:

Animación de los sprites de Donkey Jr

Serialización de entidades (cuerdas, plataformas)

Documentación interna y externa