Instituto Tecnológico de Costa Rica

Área de Ingeniería en Computadores

CE-3104: Lenguajes, Compiladores e Intérpretes

Profesor:

Marco Rivera Meneses

Tarea 3: DonCEy Kong Jr

Paradigma imperativo y orientado a objetos

Integrantes:

Carlos Adrian Araya Ramirez

Michael Shakime Richards Sparks

José Andrés Solano Mora

Abril, 2021

Tabla de contenido

1. Breve descripción del proyecto

Donkey Kong Jr es un clásico video juego arcade que consiste en llegar a la cima del mapa utilizando lianas para subir y evitar que los enemigos (kremlings) lo toquen. El objetivo de este proyecto es desarrollar el juego, con la condición crear un servidor en Java para implementar la lógica del juego y crear un cliente en C para controlar el juego.

1.1. Descripción de la utilización de las estructuras de datos desarrolladas.

Para la implementación de este proyecto solo se utilizó la estructura de datos ArrayList, sin embargo, esta estructura ya viene incluida en la librería Java.Util.

Además, se utilizaron dos patrones de diseño:

Singleton: Se utilizó el patrón de programación Singleton en la construcción de la clase Server, de tal manera que fuera posible restringir la instanciación de la clase a una instancia "única", y de igual forma realizar su ejecución una única vez.

MVC: el modelo vista controlador se utiliza en el servidor para mantenerlo lo más organizado y escalable posible. En este caso, se utilizan tres modelos principales que heredan de Entidad, y son Mono, Fruta y Cocodrilo. En el controlador se utilizan cuatro controladores, uno principal llamado GameManager que es donde se maneja toda la lógica del juego, y un controlador para cada entidad movible; CocodriloController, FrutaController y MonoController. Finalmente, en la vista solo se tiene una Ventana donde se muestra una matriz que representa el juego actual y permite al usuario crear cocodrilos y frutas.

1.2. Descripción detallada de los algoritmos desarrollados.

El juego se implementó utilizando una matriz 100x100, en la se almacenan las entidades del juego. Cada entidad tiene una posición actual y un área, la posición actual es el punto de la matriz donde se ubica la entidad y el área es un arreglo de puntos que representan la silueta del objeto, esta área tiene la función de verificar las colisiones entre las entidades.

El movimiento de las entidades se realiza actualizando la posición actual de cada entidad y automáticamente cada entidad tiene un método abstracto para actualizar su área inmediatamente al moverse. El movimiento del mono se realiza únicamente recibiendo comandos WASD, mientras que los cocodrilos se mueven utilizando un hilo que con un

sleep funciona como controlador de velocidad, haciendo que cada vez que se pasa de nivel el sleep disminuya su tiempo.

El salto del mono se implementó utilizando otro hilo y las colisiones del mono, se verifica si el mono está en el suelo comparando los puntos de la posición actual con los puntos de las plataformas y de ser así le permite saltar cada vez que presiona la tecla W, este hilo simplemente le bloquea al jugador la posibilidad de moverse mientras salta y mueve su posición hacia arriba en un intervalo de tiempo y al llegar a una altura máxima empieza a decrementar su posición.

La implementación del uso de las lianas se hace utilizando las colisiones del mono con el entorno, condicionando que si colisiona con una liana el mono deja de caer y tiene la posibilidad de subir en vez de saltar con la tecla W.

1.3. Problemas sin solución

- De manera satisfactoria se logró implementar todo lo necesario según la especificación del proyecto.

1.4. Plan de Actividades realizadas por estudiante

Fecha	Actividad	Responsable	Tiempo estimado	Tiempo requerido
11-abr	Construir el esqueleto del abstract factory en la lógica del Servidor.	Todos	2	2
14-abr	Investigar sobre sockets en Java y C.		2	1
	Investigar sobre json en Java y C.		2	1
	Implementar conexión socket del Servidor en Java.	Shakime	3	2
	Implementar conexion socket del Cliente en C.	Shakime Adrián	5	9
16-abr	Construir metodología de la base de datos en el Servidor.	Shakime	6	7
	Construir metodología unida con la interfaz en el Cliente.	José	5	5
18-abr	Desarrollar el manejo de entradas y salidas json en el Servidor.		3	5
	Desarrollar el manejo de entradas y salidas json en el Cliente.	Adrián José	5	8
19-abr	Unir el lectura y escritura de datos json entre Servidor y Cliente.	Shakime	4	5
21-abr	Investigar sobre interfaz en C (Allegro).	José	1	1
	Investigar sobre el patrón de diseño modelo-vista controlador.		1	1
	Crear la ventana de menú del juego en el cliente.		2	1
24-abr	Implementar secuencia de envio de datos para el menú de juego.	José Shakime	3	2
20-abr	Establecer la funcionalidad del Game Manager en la lógica del Servidor.	Adrián	4	2
	Implementar la creación de las entidades en la lógica Servidor.	Adrián Shakime	1	1
	Implementar la creación de las entidades en la interfaz del Servidor.	Adrián	2	1
21-abr	Implementar movimiento de las entidadades en la lógica del Servidor.		3	2
21-abr	Implementar movimiento y salto del jugador en la lógica del Servidor.		3	4
24-abr	Implementar puntajes y vidas en la lógica del Servidor.		1	1
24-may	Implementar formas de morir y colisiones en el Servidor.	Adrián Shakime	3	3
23-may	Establecer creacion de las entidades en la interfaz del Cliente.	José	2	2
23-may	Crear funciones de dibujo para las entidades y su movimiento en la interfaz.		5	9
25-may	Mostrar en la interfaz de C los puntajes y vidas del jugador.		2	1
25-may	Conectar ventanas de menú y juego principal en la interfaz de C.		2	1
26-may	Debugear y solucionar pulgas	Todos	2	2
Total			74	79

1.5. Problemas encontrados.

- Al trabajar con múltiples hilos en la lógica del juego surgieron muchos problemas de NullPointerException, esto sucedía porque en un principio los campos de la matriz que no pertenecían a una entidad se declaraban como null, y esto ocasionaba choques cuando una entidad cambiaba de posición y dejaba campos de la matriz en null que estaban siendo consultados por otros hilos. La solución a este problema fue utilizar un tipo de entidad Vacio para evitar tratar objetos null y también se utilizaron excepciones para tratar los casos inevitables en los que se intentaba acceder a un campo nulo.
- En la comunicación servidor cliente surgió un problema de envío de datos debido a la variación de la cantidad de bits utilizados para guardar datos primitos entre las diferentes plataformas. Por ejemplo, la máquina virtual Java, define los char como de 2 bytes, mientras que en el resto de los micros habituales suele ser de un byte, razón por la cual al enviar datos de C a Java, estos llegaban incompletos.

  La solución a este problema fue buscar una forma de convertir enviar los datos de una forma modelo e independiente de la micro, conocida como formato estándar de red. (chuidiang.org. (2020, 3 21)).
- Otro problema que surgió fue al intentar agregar la librería de allegro en C para la implementación de la interfaz gráfica del cliente, al trabajar con C en Windows suelen encontrarse muchos problemas de compatibilidad dependiendo del IDE, del compilador y otros factores. La solución a este problema fue una exhaustiva investigación además de prueba y error hasta que al final se logró agregar la librería.

1.6. Conclusiones

- El paradigma de la programación orientada a objetos es muy sencillo de entender en comparación a los demás paradigmas ya que se modela como vemos el mundo real.
- El lenguaje de programación C permite un acceso más privilegiado a la memoria de la computadora, permitiendo desarrollar aplicaciones más robustas a costas de una mayor complejidad y una gran curva aprendizaje.
- La herencia en POO es una herramienta muy potente a la hora de reutilizar código, permitiendo resolver problemas en menos líneas de las que se necesitarían en C.

- El servidor multithread es muy útil cuando se desea realizar operaciones pesadas sin bloquear el flujo del programa. Por ejemplo, interfaces de usuario donde se realiza un procesamiento pesado en segundo plano pero la interfaz de usuario aún está activa.
- El servidor multithread es una forma de introducir paralelismo a un programa.

1.7. Recomendaciones

- Leer el manual de usuario antes de ejecutar el programa.
- En la programación orientada a objetos es sumamente importante el uso de patrones de diseño facilitar y estandarizar los proyectos, permitiendo a los desarrolladores acostumbrarse a un estilo de programación conocido por la mayoría.

1.8. Bibliografía consultada en todo el proyecto

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

Fecha	Estudiante(s)	Actividad
11/04/2021	Todos	Planeamiento del proyecto y asignación de tareas
11/04/2021	Todos	Definir e instalar todos los recursos a utilizar en el proyecto (OS, compilador, IDE)
14/04/2021	Todos	Implementación de los sockets
19/04/2021	Todos	Desarrollo de la lógica del juego sesión 1
20/04/2021	Adrián y Shakime	Desarrollo de la lógica del juego sesión 2
22/04/2021	Adrián y José	Desarrollo de la lógica del juego sesión 3
23/04/2021	José y Shakime	Implementación de la interfaz en C
25/04/2021	Todos	Unión del servidor con el cliente