TEC | Tecnológico de Costa Rica

Escuela de Computadores

Lenguajes, Compiladores E Intérpretes

Grupo 1

Proyecto

Donkey kong jr

Estudiantes:

Isa Córdoba Quesada

Josué Granados Chacón

Jordy Calderón Najera

Profesor: Marco Rivera Meneses

Fecha de entrega: 16 de junio 2023

1° Semestre 2023

Descripción detallada de los algoritmos de solución desarrollados.

Creación y actualización de interfaz

El funcionamiento del cliente puede ser descrito mediante tres procesos individuales:

Inicialización de la interfaz

Mediante el uso de la librería Win32 API se crea una ventana, donde son cargados todos los elementos (botones, imágenes y un timer interno). Y una vez listos son desplegados al usuario. Allí se presentará al usuario una pantalla de selección entre jugador o espectador.

Inicialización del socket

Una vez seleccionado el modo de juego se lleva al usuario a la selección de servidor. Al escoger se define un flag que determinara el puerto al cual escuchará el socket. En ese instante el socket es creado. En caso de éxito de conexión enviará un primer mensaje identificándose como jugador o espectador, con el fin de que el servidor evalúe si es posible aceptarlo en el servidor, o si este ya se encuentra lleno o hace falta el jugador y se intenta conectar un tercer espectador.

Refrescamiento de la interfaz

Una vez verificada la conexión y con el visto bueno del servidor el flag de conexión es activado, lo que abre lugar a los ciclos de refrescamiento de la interfaz. Según el timer definido durante la primera etapa, cada vez que se termine se enviará al servidor un query pidiendo la información de la partida actual, el servidor responderá con un string que contiene la posición de todos los elementos de la partida. El cliente procederá entonces a eliminar todos los elementos pasados y mostrar las nuevas posiciones de cada elemento y aquellos que sean nuevos. Adicionalmente si es un jugador durante este periodo las teclas WASD y flechas del teclado están habilitadas y envían mensajes al servidor para que actualice su posición.

Parser de string lado cliente

Para mostrar cada elemento enviado por el servidor en la posición correcta se ha condensado esta información en un string de la siguiente forma:

“Vidas;Puntos;PlayerX,PlayerY;Color1,Enm1X,Emn1Y:Color2,Enm2X,Emn2Y:...;Fruta1X,Frut a1Y : Fruta2X , Fruta2Y : ...”

Inicialmente el string cargado, teniendo un array de strings donde respectivamente se tienen: las vidas, los puntos, la posición del jugador ,separada por una “,” (x,y), los enemigos, separados por “:”, y las frutas , separados por :. Una vez hecho esto se comienza a recorrer el array. Teniendo un substring de posición del jugador que de igual forma se separa por las “,”.

Lógica del juego con la interfaz del servidor

Inicialización del servidor

Cuando el servidor inicia aparece una interfaz para controlar dos juegos de DonCEy Kong Jr. Cada uno se inicia con la clase Juego que tiene un jugador, los puntos de la partida, la velocidad de los enemigos dependiendo del nivel, una lista vacía de enemigos, una lista vacía de frutas y una lista con 10 lianas que se crean automáticamente con la función crear_lianas().

Creación y eliminación de objetos desde la interfaz

En la interfaz hay tres listas desplegables para seleccionar cómo modificar el juego: Uno para decidir si se quiere eliminar o crear un objeto (lista desplegable de opción), otro para decidir el tipo de objeto (lista desplegable de objeto) y el último para seleccionar la liana en la que se quiere crear (lista desplegable de liana). También hay un slider para indicar la posición de la liana en la que se quiere crear el objeto donde 0 es el inicio de la liana y 100 es el final. Por último hay un botón para confirmar la acción y enviarla al jugador.

Al crear una fruta en la liana X(lista desplegable de liana) en la posición Y(slider) se llama a la función crear_fruta(X, Y) que convierte el valor de Y que va de 0-100 en un valor de Y relativo a la liana X, crea una fruta en esa posición y la agrega a la lista de frutas del juego. Al eliminar esa fruta se llama a la función eliminar_fruta(X, Y) que busca en la lista de frutas una que se encuentre en esa posición y la elimina.

Al crear un enemigo rojo o uno azul en la liana X en la posición Y se llama a la misma función crear_cocodrilo (Tipo, X, Y), el tipo es “1” si se quiere crear un enemigo rojo y “2” si se quiere crear uno azul. La función crear_cocodrilo llama a la función crear_rojo(X, Y) y crear_azul(X, Y) respectivamente que convierten el valor de Y a uno relativo a la liana, crea un nuevo objeto(Rojo o Azul) en esa posición y lo agrega a la lista de cocodrilos. Para eliminar el enemigo se llama la función eliminar_cocodrilo(Tipo, X) se busca, en la lista cocodrilos, un enemigo de ese tipo en esa liana y se elimina.

Manejo de respuestas del servidor

Mensaje default

Cuando el cliente le pide al servidor toda la información del juego al servidor se llama a la función game_string() de Juego que se encarga de construir el string principal del juego y llama a las funciones que mantienen al juego en movimiento. El string se construye con las vidas del jugador, los puntos del juego, la posición del jugador y las funciones cocodrilos_str() y frutas_str().

Para armar el string del juego primero se toman las vidas y los puntos, seguidamente se verifica si el jugador ganó con la función won() que si el jugador está en la misma posición que la llave, vuelve al jugador a su posición inicial, suma 1000 pts y aumenta la velocidad de los enemigos. Después de esto, se toma la posición del jugador y se llama a la función cocodrilos_str() que recorre la lista cocodrilos, verifica si hay una colisión con el jugador y se asegura de eliminar a los enemigos que se salen de la pantalla y después va formando un nuevo string con la posición actual de cada uno y llama a la función de cocodrilos move() para que estén moviéndose constantemente.

Por último se llama a la función frutas_str() que verifica si hay una colisión con el jugador para eliminar la fruta y sumarle puntos y si no va formando un string con las posiciones de las frutas. La unión de todos estos strings se envía al cliente cada 50ms para que actualice su interfaz.

Movimiento del jugador

Cuando el cliente especifica una tecla WASD se llama a la función de juego move_player(command) que se encarga, de verificar si el jugador puede realizar ese movimiento y si se puede llama a las funciones de jugador move_up(), move_down o move_sideways() que cambian la posición del jugador según la tecla que reciba

Descripción de la ejemplificación/uso de las estructuras de datos desarrolladas.

1. **Lista enlazada**: Una lista enlazada es una estructura de datos en la que los elementos están dispuestos en forma de nodos, donde cada nodo contiene un valor y una referencia al siguiente nodo de la lista. Dentro del proyecto, se utiliza dentro del servidor de java, para guardar los objetos creados, siendo los objetos los nodos de la lista. Estos objetos pueden ser los enemigos, frutas, liana.
2. **Array:** Es una colección ordenada de elementos, donde cada elemento se identifica mediante un índice o posición. Dentro del servidor de java se utilizan para guardar información a mostrar dentro de la interfaz gráfica. Dentro de la interfaz gráfica de C, se utilizaron para llevar un control de esta, siendo unos conformados por imágenes y otros por char* para recibir las indicaciones del servidor.

   .

Problemas sin solución

Utilizar imágenes .png sin fondo.

Por los métodos de creación de imágenes, no fue posible utilizar archivos .png, teniendo que utilizar archivos .bnp. Los cuales presentan el inconveniente de no poder utilizar un fondo transparente, lo que genera que la interfaz gráfica del cliente no sea tan prolija, afectando fuertemente la calidad visual del usuario.

Creación del cocodrilo azul

Al crear un cocodrilo azul en la primera liana; como esta está encima de donde aparece el jugador; el cocodrilo aparece encima de uno. Este le hace daño a uno inmediatamente, y como sigue bajando y al no haber cuadros de invisibilidad el cocodrilo le va a causar a uno el daño suficiente para matar al jugador casi que instantáneamente. Recomendamos no crear un cocodrilo azul en la primera liana.

Reinicio del juego

Al terminar una partida, el server socket dentro del servidor no se logra desocupar, por lo que no se puede conectar un juego o espectador al mismo server, para esto hay que cerrar y volver a abrir la aplicación.

Tasa de refresco

Como fue mencionado en el problema anterior las imágenes nos han causado varios problemas en la parte visual del proyecto. En este caso al haber movimiento, un conteo de vidas y conteo de puntos los cuales se deben de enviar constantemente al servidor desde el cliente y viceversa se debe de refrescar para que ambos estén con el estado más reciente recibido. Esto causa que su tasa de refresco haga que las imágenes no tengan la mejor fluidez cuando nuestro personaje o algún objeto se mueva o en ciertos casos en la pantalla de inicio que los botones aparezcan que están parpadeando.

Problemas encontrados

La conexión cliente servidor en Java no reconoce los strings.

Al enviar un string desde el cliente en C al servidor en Java por medio de un string, el servidor no detecta o lee el string enviado. Esto se arregló al leer la documentación de java y encontrar que el socket lee strings terminados en "\" por lo que se agregó esto al final de los mensajes.

Mantener dos partidas en simultáneo.

Para poder generar dos clientes al mismo tiempo, teniendo dos partidas jugables. Para esto se utilizó una clase llamada ServerHandler que permite recibir mensajes dentro de un mismo puerto. Cuando se realiza una llamada, se genera un ServerThread para

manejar los mensajes del cliente en específico. ServerThread tiene una referencia al Serverhandler que lo genera, de esta forma El ServerHandler puede ser modificado por las instancias de ServerThread y así ejercer el control sobre cuantas más solicitudes de cliente acepta o si alguno de los clientes se ha desconectado.

Parsear el string que se recibe del server

Al comienzo para parsear el string se utilizaba el resultado de strtok pero por retornar un puntero al manipular el resultado de dicho puntero se perdía la referencia, por lo que se utilizan arrays guardar todos los punteros creados al recorrer todos los resultados de strtok y luego operar sobre los array.

Mostrar las imágenes en la interfaz del servidor

Las imágenes de la interfaz del servidor no se lograban mostrar de la forma que se quería ya que en lugar de estar en el fondo se colocaban a la par del resto de componentes gráficos. Para resolver este problema se da ImagePanel derivada de la clase JPanel de java swing que dibujaba la imagen en el panel al crearlo.

Mantener la velocidad de los enemigos

Cuando un jugador gana una partida, la velocidad de los enemigos debe aumentar pero inicialmente solo aumentaba la velocidad de los enemigos existentes y no los nuevos que se creaban. Para resolver este problema se cambió la variable velocidad y se hizo estática.

Movimiento del jugador

El movimiento del jugador tenía varios problemas. Lo primero es que aunque su movimiento en X si estaba restringido para que solo se moviera de una liana a otra, su movimiento en Y le permitía moverse sin restricciones. Para resolver esto se agregó un atributo Liana a la clase Jugador para saber en cual liana se encuentra y se agregó una condición para que solo se moviera en el rango de inicio y final de dicha liana.

Plan de Actividades

Plan de Actividades: PDC

Fecha de Inicio: 2 de junio del 2023

Fecha de Finalización: 16 de junio del 2023

Objetivo del Proyecto: Crear un juego de don king kong en base en C que se conecte con un server socket en java

Actividades y Tareas:

1. Crear un server utilizando Java por medio de sockets
    - Descripción: crear un socket server en java para poder manejar el juego de DK
    - Responsable: Jordy Calderón Najera.
    - Fecha de inicio: 1 de junio del 2023
    - Fecha de finalización: 10 de junio del 2023
2. Enviar instrucciones al juego por medio de un servidor..
    - Descripción: el servidor de java debe conectarse con la interfaz en C, enviando instrucciones y puntuaciones.
    - Responsable: Jordy Calderón Najera.
    - Fecha de inicio: 10 de junio del 2023
    - Fecha de finalización: 15 de junio del 2023
3. Generar una interfaz gráfica para un juego de DK.
    - Descripción: generar una interfaz gráfica en C para un juego de DK..
    - Responsable: Isa Cordoba Quesada.
    - Fecha de inicio: 1 de junio del 2023.
    - Fecha de finalización: 10 de junio del 2023.
4. Permitir que la interfaz gráfica se genere a partir de instrucciones dadas.
    - Descripción: Permitir que la interfaz gráfica se modifique mediante instrucciones dadas.
    - Responsable: Isa Cordoba Quesada.
    - Fecha de inicio: 10 de junio del 2023.
    - Fecha de finalización: 15 de junio del 2023.
5. Realizar la conexión entre el juego y el server..
    - Descripción: Realizar una conexión por medio de sockets entre la interfaz en C y el servidor en Java.
    - Responsable: Josué Granados Chacon.
    - Fecha de inicio: 1 de junio del 2023.
    - Fecha de finalización: 10 de junio de 2023.
6. Crear un parser que entienda las instrucciones generadas por el server y las traduzca para la interfaz gráfica.
    - Descripción: Generar un traductor para las instrucciones del servidor, hacia la interfaz gráfica, para que esta sepa lo que hacer con esto.
    - Responsable: Josué Granados Chacon.
    - Fecha de inicio: 10 de junio del 2023.

- Fecha de finalización: 15 de junio de 2023.

Conclusiones.

En lo que era la parte de C podemos concluir que la librería de Windows Win32 API nos facilita la creación de interfaces gráficas y crear sockets que logran conectar al cliente en C con el servidor en java. También podemos concluir que podemos representar un array de string como una matriz o un array de char*. Estos punteros nos facilitan en ciertas funciones y en el almacenamiento de datos aunque si uno pierde registro fácilmente pueden causar problemas.

En lo que consta de Java, la librería Java Swing nos facilita la creación de interfaces gráficas dado a su gran cantidad de componentes y orden de crear nuevos. Las propiedades de programación orientada a objetos también se puede notar en Java Swing con la herencia de los objetos creados. En Java se pueden utilizar listas y arreglos para almacenar datos. Al Java ser orientada a objetos esta nos facilita programar la lógica del juego de manera ordenada.

Recomendaciones.

Asegurarse que al abrir y cargar el proyecto que este esté al mismo nivel, en otras palabras los documentos de java json y c deben de estar al mismo nivel en término de profundidad de path. Esto para asegurarse que cada una tenga mas facilidad de comunicarse entre ellas sin tener que ver donde esta el path de cada uno o que se dejó sin definir diferentes clases por la profundidad que tenga.

Mantener la definición de clases limpia y fácil de entender para poder crear varios hijos sin que estos pierdan su herencia; también hecho de tal manera que uno pueda cambiar ciertas características de solamente unos hijos.

Link al repositorio.

https://github.com/DuesanNoften/Doncey-kong-jr

Bibliografía

Kernighan, B. W., & Ritchie, D. M. (1978). *The C Programming Language*. https://usuaris.tinet.cat/bertolin/pdfs/c_programming_language.pdf

Java Documentation - Get Started. (2023, 31 Enero). Oracle Help Center. https://docs.oracle.com/en/java/

Patrones de diseño en Java. (s. f.). https://refactoring.guru/es/design-patterns/java

Play Free Classic Games - Online Nintendo NES Emulator - Donkey Kong Jr - Online browser play of classic Nintendo NES, retro Atari games and original Sega Arcade games - Free play. (2020, 9 diciembre). Free80sArcade. https://www.free80sarcade.com/nesonline_Donkeykongjr.php