

Instituto Tecnológico de Costa Rica

Semestre I

2021

DonCEy Kong Jr.: Bitácoras

Lenguajes, compiladores e intérpretes

Tarea#3: Programación imperativa y orientada a objetos

Prof. Marco Rivera Meneses

José Alejandro Chavarría Madriz

Natalia González Bermúdez

2019067306

2019165109

Miércoles 14 de abril

Se realiza una reunión inicial para la creación del plan de actividades y discusión de generalidades del proyecto

Jose Alejandro: Creación de documentos para la documentación y repositorio.

Jueves 15 de abril

Jose Alejandro: Adecuación de los ambientes, se descarga el JDK para java y el compilador para C.

https://code.visualstudio.com/docs/languages/cpp#_intellisense

Se comienza a intentar implementar sockets en ambos lenguajes.

<https://www.codejava.net/java-se/networking/java-socket-server-examples-tcp-ip>

<https://stackoverflow.com/questions/31323015/c-socket-programming-under-windows>

<https://docs.microsoft.com/en-us/windows/win32/winsock/complete-client-code>

Viernes 16 de abril

Jose Alejandro: Con el fin de esclarecer el funcionamiento de los sockets se continúa con la implementación de una librería gráfica en C. Como se está programando para windows se decide utilizar Win32.

https://www.youtube.com/watch?v=9JMQkUOhW1s&list=PLWzp0Bbyy_3i750dsUj7yq4JrPOIUR_NK&index=4

http://www.winprog.org/tutorial/message_loop.html

<https://social.msdn.microsoft.com/Forums/vstudio/en-US/aa02e263-1f65-453b-90ae-492912e6e56c/disable-the-maximize-button-in-a-win32-program?forum=vcgeneral>

<https://stackoverflow.com/questions/16546371/how-to-add-gdi32-lib-from-command-line/16546503>

<https://docs.microsoft.com/en-us/windows/win32/learnwin32/keyboard-input>

<https://docs.microsoft.com/en-us/windows/win32/inputdev/virtual-key-codes>

Natalia: Se realiza el setup de los ambientes y de los compiladores

Sabado 17 de abril

Jose Alejandro: Se continúa con la implementación de de la interfaz.

<https://stackoverflow.com/questions/6275718/hiding-a-control-in-windows>

Se encuentra el problema que al actualizar las imágenes rápidamente en la pantalla hay un parpadeo extraño tras una pequeña investigación se encuentra una solución.

<https://social.msdn.microsoft.com/Forums/vstudio/en-US/e9b060eb-3396-4989-ba8c-6146601e8a5e/how-to-clearscreen-in-windows-application?forum=vcgeneral>

[https://forums.codeguru.com/showthread.php?338319-Screen-flicker-problem-InvalidateRect\(\)](https://forums.codeguru.com/showthread.php?338319-Screen-flicker-problem-InvalidateRect())

También se comienza a implementar un programa para poder parsear el string que se espera sea enviado desde el servidor.

<https://www.educative.io/edpresso/splitting-a-string-using-strtok-in-c>

<https://stackoverflow.com/questions/29863751/how-to-copy-convert-a-char-string-to-a-wchar-t-string-in-c>

Finalmente se intentó utilizar imágenes .png sin fondo pero no se logró encontrar ninguna manera que funcionara con la implementación realizada. Cualquier otra forma, además de tener una complejidad importante suponía básicamente comenzar desde 0 la interfaz, por lo que se decide desistir de las imágenes .png

Natalia: Se crean todas las clases del juego de enemigos Cocodrilo, Azul y Rojo, la de Jugador, Fruta y se termina una lógica inicial del juego en la que se pueden crear y eliminar frutas y enemigos con las funciones crear_rojo(X, Y), crear_azul(X, Y), crear_fruta(X, Y), eliminar_enemigo(X, Y) y eliminar_fruta(X, Y).

Domingo 18 de abril

Jose Alejandro: Se logra finalmente, tras hacer una serie de correcciones importantes ahora tomando en consideración que al dividir un string el resultado es un puntero, que el cliente pueda interpretar correctamente el string y posicionar todas las imágenes en la posición indicada. Además se implementa una animación simple para el personaje principal.

Natalia: Se continúa con la lógica del juego, se realizan las funciones en caso de que el jugador gane y se realizan las funciones de colisión de frutas y enemigos para saber si la posición del jugador es igual que la del objeto, en caso de que colisione con la fruta se suman puntos y se elimina dicha fruta y en caso de que colisione con un enemigo se realizó la función hit() en jugador que lo devuelve a la posición inicial y le quita 1 vida al jugador.

Lunes 19 de abril

Jose Alejandro: Se trabaja en la integración de selección de servidor en la interfaz cliente y en un manejo de errores de desconexión y otros asociados al servidor.

Natalia: Se cambian algunos de los aspectos de lógica del juego, principalmente en el movimiento del jugador y se realizan las funciones en jugador move_up(), move_down(), move_sideways() que cambian la posición del jugador.

También se crea la función `game_str()` que une todos los datos del juego para enviarlos al cliente que recopila los puntos, las vidas, la posición del jugador y la posición de todos los enemigos y las frutas. Utiliza las funciones auxiliares `cocodrilos_str()` que mueve a los cocodrilos según su patrón y busca colisiones para retornar un string con todos los enemigos y `frutas_str()` que detecta colisiones y arma un string con las frutas.

Se realiza una interfaz de prueba utilizando la librería de Java Swing con un `JTextField` que actúa por comandos. Para crear y eliminar enemigos se utiliza `Cenemigo,a,b,c` y `Eenemigo,a,b,c` y para las frutas `Cfruta,b,c` y `Efruta,b,c` en donde `a` es el tipo de enemigo, `b` es la liana en la que se quiere crear y `c` es un número del 0-100 donde 0 es la Y inicial de la liana y 100 es la Y final de la liana.

Se integra a lógica con el servidor de java y se realiza un ajuste de la velocidad de los enemigos.

Martes 20 de abril

Jose Alejandro: Se trabaja en algunas correcciones menores y se comienza con la implementación de un server múltiple en java.

Natalia: Se realiza una corrección del movimiento del jugador ya que cuando ganaba no guardaba bien la posición y hacía movimientos impredecibles por un momento y además se arreglan las colisiones ya que solo funcionaban si el jugador y el objeto estaban en el mismo punto y eso era difícil que sucediera entonces se pone un rango en el que podría haber una colisión.

Miércoles 21 de abril

Jose Alejandro: Se termina con la implementación del multi servidor en java y se corrigen algunos errores importantes en la comunicación, manejo de errores y arreglos en la animación. Además las adaptaciones necesarias en el cliente para el correcto funcionamiento del multiserver.

Documentacion

Jueves 22 de abril

Jose Alejandro: Documentación interna y documentación

Viernes 23 de abril

Jose Alejandro: Documentación

Natalia: Se realiza una interfaz más amigable con el cliente, se cambia el JTextField con 3 JSpinners, uno para seleccionr si crear o eliminar, uno para seleccionar el objeto y otro para seleccionar la liana y se pone un JSlider para seleccionar la parte de la liana en la que se quiere poner y un botón para enviar el mensaje al cliente. Esto se hizo por cada uno de los juegos.

<https://stackoverflow.com/questions/39922172/how-to-change-the-background-color-of-a-jspinner-number-model>

<https://docs.oracle.com/javase/tutorial/uiswing/components/slider.html#looks>

<https://docs.oracle.com/javase/7/docs/api/java/awt/Color.html>

http://www.java2s.com/Tutorials/Java/Swing_How_to/Basic/Add_Background_image_to_JPanel.html

Sabado 24 de abril

Jose Alejandro: Documentación y debuggeo

Natalia: Se trabaja en la estética de la aplicación del servidor, se crea la clase ImagePanel que se extiende de JPanel para poder poner una imagen de fondo y se realiza la customización de cada uno de los componentes.

Domingo 25 de abril

Jose Alejandro: Documentación y debuggeo

Natalia: Documentación