Instituto Tecnológico de Costa Rica
Ingeniería en computadores
Antonio González Torres
Harold Espinoza Matarrita (2019185140)
Fabricio Mena Mejia (2019042722)
Grupo #4

Tabla de contenido

Introducción: 1

Descripción del problema: 1

Análisis de resultados: 2

Bitácora: 2

Estadística de Tiempos: 3

Conclusión Personal: 4

Introducción:

Este proyecto tiene como objetivo aprender acerca de los sockets y la estructura cliente-servidor usando hilos y una interfaz gráfica de usuario mediante la programación del juego “Dakar death”. Este juego utiliza tkinter, pygame y la versión de python 3.7.2. El juego consiste en un mapa abierto con temática de desierto donde 2 jugadores compiten entre si para ver quien consigue alcanzar la meta, los carros tienen 3 niveles, inicia en el nivel 1 y conforme obtiene puntaje va subiendo de nivel, gana el jugador que alcance la meta primero.

Al iniciar el programa aparecen los dos jugadores en nivel 1 y aparecen “carros tontos”, el mapa también posee zonas de peligro que reducen los puntos del jugador, la meta está en el centro, pero solo puede ser accedida si el jugador tiene 300 pts y las 4 banderas necesarias, los puntos son obtenidos disparando a los carros tontos y con las banderas.

Aparecen 8 enemigos o carros tontos los cuales pueden hacer respawn una vez y pueden subir de nivel, conforme suben de nivel tienen regeneración de vida

El juego termina cuando la meta es alcanzada o si el jugador selecciona cierra el juego.

Descripción del problema:

Por medio de python tkinter, pygame, sockets y threads se debe desarrollar el juego “Dakar Death” el cual debe contener: dos jugadores, carros tontos, minas, disparos, además el jugador puede tener 3 niveles posibles que funcionaran como upgrades del carro.

El juego también deberá poder introducir el nombre de usuario y en caso de ser uno de los puntajes más altos se almacenará en un archivo json junto con su puntaje, los 5 puntajes más altos serán mostrados si el jugador selecciona la pestaña “file” -> “highscores”, la cual abrirá una ventana que mostrará lo highscores.

Además debe haber colisiones con el jugador

Análisis de resultados:

Reutilizamos el menú creado anteriormente en el proyecto “Space Invaders” pero con ligeros cambios, esto para ahorrar tiempo de programación. Después conseguimos a partir de un código base que muestra una ventana de pygame, implementar una ventana de pygame luego de seleccionar el botón de “play”.

Incluimos una ventana para los highscores con el conocimiento previo de tkinter y los “menubar”.

Investigamos acerca del funcionamiento de los sockets y la estructura cliente-servidor, utilizamos un programa de chat para entender su funcionamiento, luego buscamos información de como implementarlo al juego y a partir de otros códigos utilizados como referencia lo adaptamos al juego. Creamos los sprites y los introducimos al juego

Posteriormente creamos los enemigos y obstáculos que aparecen en la partida, también aparece el jugador2 y sus disparos en la pantalla del jugador1

Bitácora:

06/05/2019 (19:00 - 21:00) 2h

Se crea un nuevo menú usando como referencia el menú del proyecto anterior. Además, se agrega un submenú en la pestaña "file" llamado "Highscores" que abre una ventana la cual muestra los puntajes más altos

07/05/2019 (14:00 – 21:00) 7h

Investigamos acerca de los sockets e hicimos un programa de chat como ejemplo para el entendimiento de los sockets unido a los hilos. Se investiga sobre cómo podemos crear una ventana en pygame y que pueda mover el "background" que tiene un tamaño más grande que la ventana

08/05/2019 (19:00 - 22:00) 3h

Comenzamos a programar el juego, le agregamos un background, y sprites de movimiento. implementamos un botón de play el cual cierra la ventana del menú y abre la del juego. Después investigamos como adaptar el servidor al juego

09/05/2019 (14:00 - 20:00) 6h

Se le agregaron disparos al jugador. Se consigue implementar el servidor y los sockets al juego

10/05/2019 (20:00 - 22:00) 2h

Se comienza a adaptar el código del juego con el de servidor y cliente. El profesor nos recomendó utilizar el multicast para facilitar la conexión así que se añade el multicast al juego

11/05/2019 (16:00 – 21:00) 5h

Se le ponen sprites a los jugadores

12/05/2019 (13:00 – 22:00) 9h

Se añaden disparos y otros ajustes al juego en base a otro código que hicimos aparte con pruebas

14/05/2019 (16:00 – 21:00) 5h

Se inserta el formato de coordenadas al servidor para: el número de cliente, coord del jugador y coord de las balas enemigas

16/05/2019 (13:00 – 23:00) 10h

Se inicia la documentación externa del juego. Se logra que aparezcan: el jugador2 y sus disparos en la pantalla del jugador1

17/05/2019 (19:00 - 23:00) 4h

Se comenta el código del juego para la documentación interna. Se realizan pruebas con el multicast para intentar adaptarlo al servidor que ya tenemos. Además, añadimos zonas de peligro para el jugador

18/05/2019 (13:00 – 21:00) 8h

Se adaptan los obstáculos y enemigos creados en el código de pruebas

19/05/2019 (12:00 – 22:00)10h

Se solucionan problemas de lag y del erróneo dibujado de los enemigos. Además, se añaden las imágenes restantes

20/05/2019 (13:00 – 23:00)10h

Se añaden datos extra para enviar por el socket y se corrigen errores con el score y también se añade la pausa, y se finalizan los documentos

Estadística de Tiempos:

Análisis de Requerimientos	07 horas
Investigación de funciones	16 horas
Programación	23 horas
Documentación	10 horas
Pruebas	18 horas
Elaboración del Documento	07 horas
Total	**81 horas**

Conclusión Personal:

Aprendimos a utilizar sockets, la estructura cliente-servidor y el envío de datos por medio de este. También adquirimos conocimiento de la interfaz gráfica de pygame, como manejar las clases de esta, además utilizamos el conocimiento previo de la interfaz gráfica de tkinter y el manejo de archivos json