

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

Tabla de contenidos:

Tabla de contenidos	1
Introducción	1
Descripción del problema	1-2
Análisis de resultados	2
Bitácora	2-3
Estadística de tiempos	4

Introducción:

Este proyecto tiene como objetivo comprender la interacción del software y hardware por medio de un control que deberá ser adaptado para el juego Dakar death, se deberá utilizar el microcontrolador Arduino y la librería pyserial.

Este juego utiliza tkinter, pygame, la versión de python 3.7.2 y la versión del IDE de Arduino 1.8.9. El juego consiste en un mapa abierto con temática de desierto donde 2 jugadores compiten entre sí 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. El control esta conformado por un joystick y un botón que controlaran el movimiento del jugador.

Descripción del problema:

Por medio de un microcontrolador Arduino se debe crear un control que controle las funciones del jugador como: su movimiento, disparo, frenado, entre otros. En el caso de

los disparos se deberá encender un LED cada vez que el jugador dispare. Para el movimiento se deberá utilizar un joystick, que moverá al jugador dependiendo del movimiento del joystick.

También se deben arreglar los problemas que quedaron en el proyecto anterior, en nuestro caso los detalles a resolver son: La pausa y detalles con los puntajes.

Análisis de resultados:

Por medio del conocimiento previo de Arduino de uno de los estudiantes se creó el programa que utiliza el microcontrolador se montaron los componentes del control en una protoboard, se logró hacer que el movimiento del jugador sea controlado por medio de un joystick, además de la función de disparo sea controlada por un botón al igual que otras funciones del juego.

Bitácora:

Aquí se muestra la bitácora del segundo proyecto unida a la del tercer proyecto, ya que este último es una continuación del segundo.

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, coordenadas del jugador y coordenadas 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.

03/06/2019 (10:00 - 15:00) 5h

Se construye un programa de arduino para utilizar un joystick y un botón para el juego. Además, se investiga cómo utilizar la librería serial de Python y cómo manejar la información que envía el arduino. Se opta por tratar la información de arduino como si fuera un archivo json.

05/06/2019 (18:00 - 23:00) 5h

Se agregan los botones de guardar score y banderas de la partida actual y poder cargarlos en otra partida, también se arreglaron algunos errores que tenía el código. Se integran más botones para el juego y se corrigen errores de la versión anterior, como el guardado de scores.

8/6/2019 (14:00 – 16:00) 2h

Se termina de hacer la documentación externa y se agrega el diagrama UML.

Actividad Realizada	Número de horas
Análisis de Requerimientos	9 horas
Investigación de funciones	18 horas
Programación	25 horas
Documentación	11 horas
Pruebas	22 horas
Elaboración del Documento	8 horas
Total	93 horas