



Ingeniería en Computadores
Taller de Programación

Proyecto programado 2:

pyDeathRace 3.0

Profesor:
Antonio Gonzáles Torres

Elaborado por:
Adrián José Chacón Campos 2018194775
Antony Fabian Fallas Elizondo 2018178906

I Semestre, 2018

Tabla de Índices

<u>Introducción.....</u>	<u>1</u>
<u>Descripción del Problema.....</u>	<u>1</u>
<u>Análisis de Resultados.....</u>	<u>3</u>
<u>Bitácora de actividades.....</u>	<u>5</u>
<u>Conclusión Personal</u>	<u>8</u>
<u>Bibliografía.....</u>	<u>9</u>

Introducción

Existen distintos tipos de videojuegos, como los shooter en primera persona, juegos de rol, simuladores, estrategia en tiempo real, entre otros; detrás de cada videojuego, hay cientos de líneas de código que se deben ejecutar para que el juego sirva, distintos juegos requieren de una lógica diferente para manejar los componentes; la mayoría del tiempo, los videojuegos no funcionan a la perfección, esto se debe a que existen varias personas trabajando en el código, y cada uno tiene una forma distinta de programar, debido a esto, a la hora de unir el trabajo realizado existe la posibilidad de que hayan funciones que no sean compatibles unas con otras.

Descripción del Problema

Este proyecto requiere que desarrolle un control para el juego pyDeathRace usando un Arduino, 2 joysticks, 2 botones, dos LED y un programa que envía mediante una comunicación serial entre el microcontrolador y una computadora los códigos de los joysticks y botones cuando son presionados.

El juego es multi-jugador y puede ser usado por dos jugadores de forma simultánea, en donde cada uno puede jugar haciendo uso de un joystick y un botón. Los joysticks permiten que los jugadores se muevan en diferentes direcciones y ataquen con disparos a los vehículos “dummy” y obstáculos. El funcionamiento de los joysticks, botones y LEDs es el siguiente:

1. Se utiliza un joystick para mover el vehículo del jugador hacia adelante, atrás, izquierda y derecha.
2. El botón se utiliza para disparar. Cuando es botón es presionado lanza un disparo y se enciende el LED. El objetivo, un vehículo “dummy” o un obstáculo, se destruye cuando es alcanzado por un disparo.
3. El LED se enciende cada vez que se hace un disparo y se apaga de forma inmediata.

Para establecer la comunicación serial entre el Arduino y la computadora utilice la biblioteca pySerial. La instalación se puede realizar desde la consola, tanto en Windows como Linux, con el comando `Python -m pip install pyserial`. Una vez que se ha realizado la instalación, se puede hacer la lectura de los datos del dispositivo de control usando el programa en Python con el comando `“import serial”` para importar pySerial. Tenga en

cuenta que los datos enviados y recibidos por el Arduino son datos tipo string codificados en ASCII.

pyDeathRace 3.0 es la tercera versión de un juego en el que pueden competir uno o dos jugadores usando como control las teclas de una computadora y de acuerdo con su puntaje pueden pasar de un nivel a otro, por lo que el juego contará con al menos 3 niveles.

En esta versión del juego el sistema contará con un menú con las siguientes opciones:

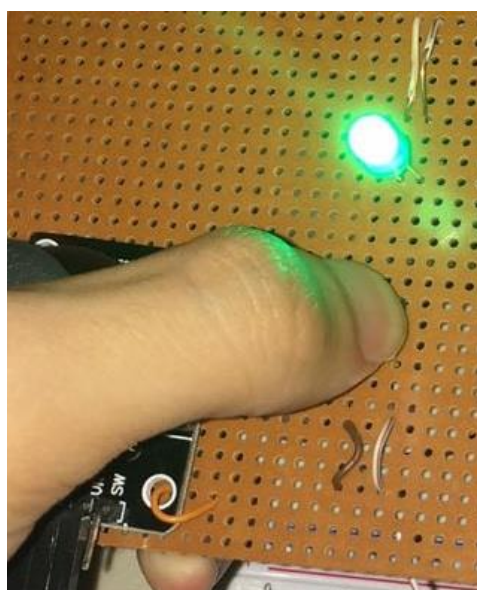
1. Registrar nombre de usuario: El sistema permite que el jugador ingrese el nombre de su usuario al cual se asociará el puntaje más alto que logre en las competencias. El puntaje de cada usuario se almacenará en un archivo JSON.
2. Ver puntaje: Cuando esta opción es seleccionada, el sistema muestra el puntaje de los 5 mejores jugadores.
3. Iniciar partida: Al seleccionar esta opción el jugador debe seleccionar el nombre de usuario con el cual jugará la partida.
4. Guardar partida: El jugador puede guardar el estado de una partida al seleccionar esta opción de menú. El sistema debe almacenar las posiciones actuales de todos los carros del juego y los puntajes de los jugadores.
5. Reiniciar partida: Esta opción es utilizada para continuar una partida que había sido iniciada con anterioridad, con toda la información de las posiciones de los carros y los puntajes de los jugadores.
6. Salir: El usuario se sale del sistema y termina la partida, si estuviera jugando en ese momento.

Los jugadores pasan de nivel cuando el jugador ha superado 3 minutos de juego sin sufrir una colisión. Los niveles superiores del juego cuentan con pistas más complicadas (definidas por el programador – la mayor dificultad de las pistas favorece el otorgamiento de más puntos) y con un mayor número de obstáculos en la pista, que además de más carros puede incluir rocas, muros y árboles. Además, esta versión cuenta con sonidos de aceleración, disminución de la velocidad, frenado y explosiones al chocar.

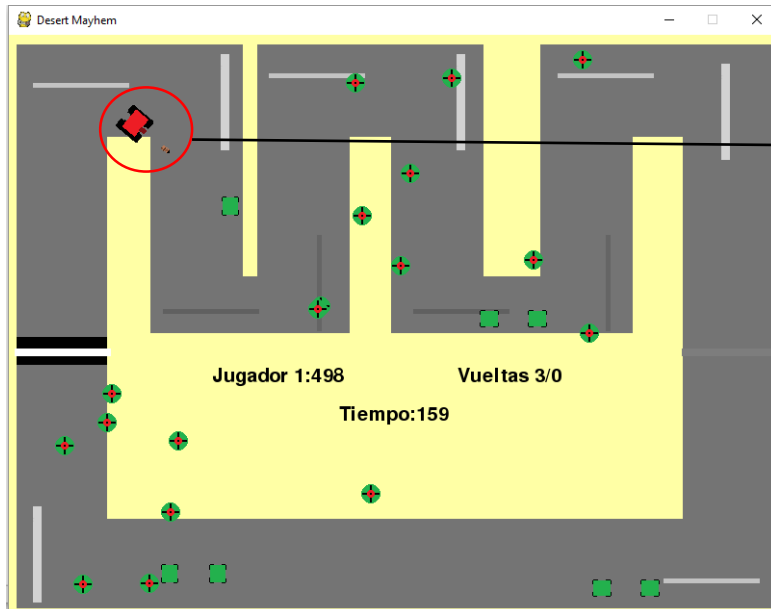
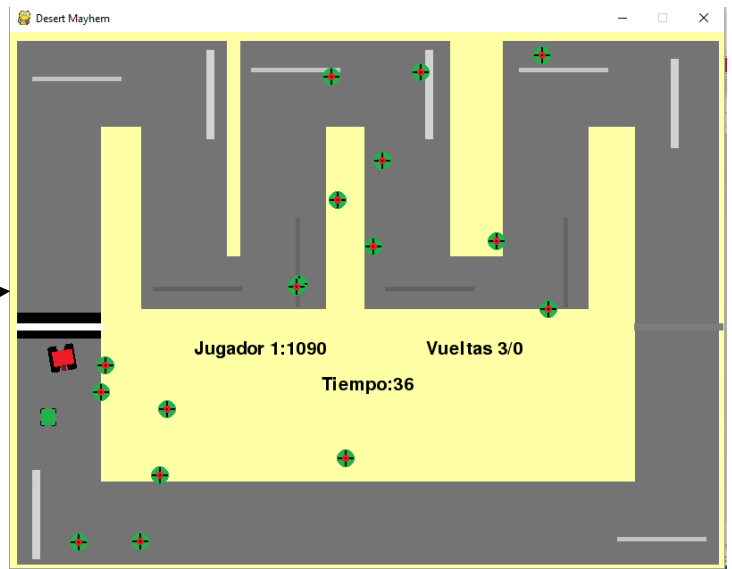
Al igual que la versión anterior, en este juego el competidor debe sobrepasar o eliminar a los oponentes que son controlados por el mismo juego. Las reglas básicas del juego son las siguientes:

1. El tiempo de cada partida del juego es de 3 minutos.
2. El juego permite que uno o dos jugadores compitan a la vez.
3. El sistema controla un gran número de “dummy vehicles” que se mantienen en la pista y corren a una velocidad constante.
4. El punto de partida de la competencia, y donde se completa cada vuelta, se encuentra indicado en la pista por la zona marcada como un tablero.
5. La velocidad de los vehículos de los jugadores es variable y depende de las maniobras que hagan con las teclas de las computadoras.
6. El puntaje de cada jugador se calcula con base en el número de vueltas que hace al circuito de carreras y el número de “dummy vehicles” que elimina durante la competencia.
7. Un jugador no puede eliminar el vehículo del competidor rival.
8. El jugador que se sale de la pista sufre el rebajo de puntos, de acuerdo con las reglas que defina el programado.
9. Si un jugador sufre una colisión con un objeto al salirse de la pista puede quedar destruido.

Análisis de Resultados

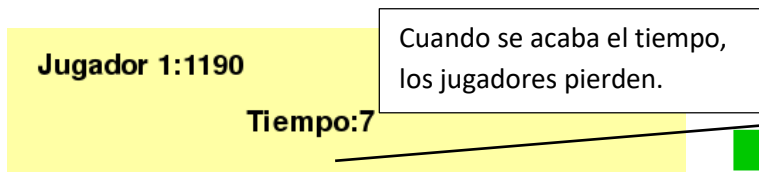


Desarrollo de partida



Cuando el jugador presiona la tecla o el botón de disparo, un misil se dispara del sprite del jugador, si el misil colisiona con un dron, los puntos del jugador aumentan considerablemente; si el misil colisiona con una mina, los puntos del jugador aumentan un poco; si el misil se sale del mapa sin colisionar con nada, desaparecerá

Se acabo el tiempo.



Pista completada



Estas pantallas aparecen cuando los jugadores completan una pista y cuando terminan una tanda de juego (3 pistas) respectivamente

Bitácora de actividades

Investigación	Arduino	Fecha:	09/05/2018
Hora Inicial:	9:00 am	Hora Final:	3:00 pm
Investigación sobre Arduino			

Investigación	PyGame	Fecha:	10/05/2018
Hora Inicial:	9:00 am	Hora Final:	3:00 pm
Investigación Pygame			

Investigación - Manufacturación	Arduino - Control	Fecha:	12/06/2018
Hora Inicial:	11:00 am	Hora Final:	5:00 pm
Investigación sobre programar Arduino y manufacturación de partes del control			

Programación - Manufacturación	Lógica - Control	Fecha:	13/06/2018
Hora Inicial:	12:00 pm	Hora Final:	2:00 am
Programación de lógica del juego			

Manufacturación	Control	Fecha:	14/06/2018
Hora Inicial:	1:00 pm	Hora Final:	4:30 pm
Manufacturación del control			

Programación	Lógica	Fecha:	14/06/2018
Hora Inicial:	4:30 pm	Hora Final:	7:40 pm
Programación de lógica del juego			

Estadística de Tiempos (\pm 30 minutos)	
Análisis de Requerimientos	01 h
Diseño de la aplicación y diagrama de clases	00 h
Investigación de funciones	17 h
Programación	23 h
Documentación interna	01 h
Pruebas	01 h
Elaboración documento	02 h
Total	45 h

Conclusión Personal

Al realizar este proyecto aprendimos sobre distintas características de Python, Arduino y la fabricación de dispositivos electrónicos, como la forma en que se manejan algunas variables y librerías como math y time; no utilice Tkinter pero me pareció que PyGame ofrece una forma fácil e intuitiva de programar videojuegos sencillos; también aprendí a utilizar programación orientada a objetos y personalmente me pareció que es bastante útil.

Bibliografía

Albert Sweigart. (2012). Making Games with Python & PyGame.

Sentdex. (2014). PyGame with Python 3 Game Development. 2018, de YouTube Sitio web:

https://www.youtube.com/watch?v=ujOTNg17LjI&list=PLQVvvaa0QuDdLkP8MrOXLe_rKuf6r80KO

codingchili. (2017). py-race. 2018, de GitHub Sitio web:
<https://github.com/codingchili/py-race>

Making Games With Pygame. Sitio web:
<https://www.pygame.org/docs/tut/MakeGames.html>