

Carrera: Ingeniería en Computadores

Curso: CE-1102 Taller de Programación, I Semestre 2018

Proyecto programado III, grupo 4

Valor: 100 puntos, 25%

Trabajo en grupo: 2 personas

Entrega: 12 de junio del 2018 a las 11:55 p.m.

Defensa: 13 de junio del 2018 de acuerdo con la hora de la cita acordada

Profesor: Antonio González Torres

Consultas: antonio.gonzalez@tec.ac.cr

1. Instrucciones

Los alumnos escogerán una de las versiones de PyDeathRace 2.0 desarrolladas por los miembros del grupo y realizarán las mejoras necesarias para que cumpla con los requerimientos solicitados. El trabajo se enviará por correo electrónico el día 12 de junio y se defenderá al día siguiente en el horario que se acuerde con cada grupo.

2. Objetivo

Implementar un controlador de hardware para el juego pyDeathRace 3.0 **utilizando un microcontrolador Arduino y la biblioteca PySerial** con el fin de comprender la interacción entre los dispositivos de hardware y los sistemas de software.

3. Introducción

El diseño de sistemas que involucran una parte de hardware y otra parte de software es una de las áreas de trabajo de un Ingeniero en Computadores, y en consecuencia este proyecto consiste en el desarrollo de un sistema simple que toma en cuenta las interrelaciones entre estos. Aunque el diseño de hardware es simple, permite que se relacione y aprecie el potencial de algunos componentes que serán objeto de estudio en cursos futuros, en los cuales podrá concretar otros trabajos con mayor fundamento teórico y conocimiento para optimizarlos.

En este proyecto se utilizará la tecnología de microcontroladores para construir un circuito electrónico sencillo que trabajará con un componente de software. En concreto, el trabajo consiste en construir un dispositivo de control para el juego pyDeathRace 3.0, el cual es la tercera versión del juego y se basa en la versión 2.0 que fue desarrollada en el segundo proyecto del curso.

4. Descripción del proyecto

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.

5. Diseño del control

En esta sección se presentan los diferentes elementos que debe considerar para el diseño del control. Así, en la Tabla 1 se resumen los comandos que debe enviar y recibir la tarjeta Arduino para el desarrollo del control.

Tabla 1. Comandos para interactuar con el Arduino

Instrucción	Efecto	IN/OUT
'S'	Enciende LED	IN
'B1'	Botón B1 presionado	OUT
'J-Left'	Joystick girado al lado izquierdo	OUT
'J-Right'	Joystick girado al lado derecho	OUT
'J-Down'	Joystick girado hacia abajo	OUT
'J-Up'	Joystick girado hacia arriba	OUT

La Figura 1 muestra una vista superior del Arduino con la numeración de los pines de entrada/salida que serán usados como base para el diagrama eléctrico que se presenta en la Figura 2.



Figura 1. Vista superior del Arduino.

La Figura 2 describe la conexión del joystick con la tarjeta Arduino. Se recomienda el uso de un conector DB9 (juego hembra-macho) para la interconexión de las terminales del resto de componentes con el Arduino.

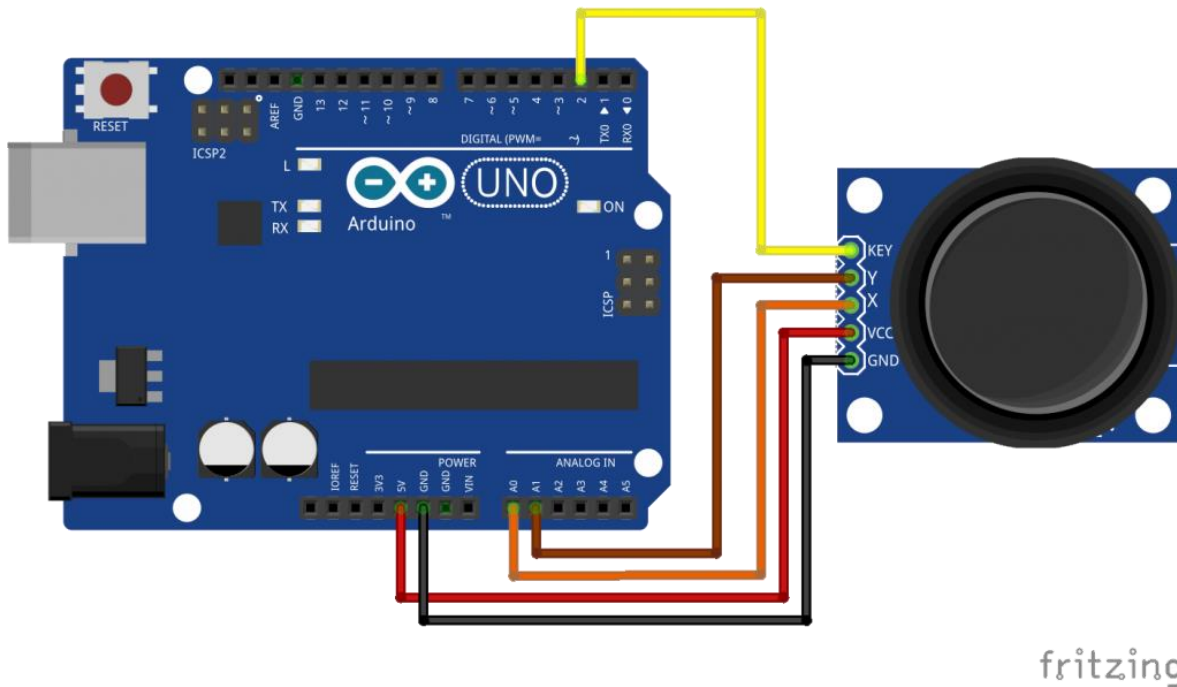


Figura 2. Conexión del joystick con el Arduino.

6. Materiales requeridos

La siguiente es la lista de materiales necesarios para la construcción del circuito:

1. Placa pre-perforada con lands de cobre sin contactos en la parte de cobre.
2. 1 push button normalmente abierto.
3. 5 resistencias de 330 ohms.
4. Un LED pequeño de color amarillo.
5. Cable UTP Cat5 para la preparación de conexiones con el Arduino.
6. Un conector DB9 hembra con su carcasa (deseable)
7. Un conector DB9 macho con su carcasa (deseable)

7. Investigación

El estudiante debe investigar sobre UML, el uso de la biblioteca pySerial y sobre cómo mejorar pyDeathRace 2.0 con clases y métodos bien definidos para el manejo de archivos JSON y CSV, y archivos de sonido que permitan escuchar los disparos y las aceleraciones de los vehículos.

Documentación.

El estudiante debe entregar documentación interna y externa del proyecto. La documentación interna se refiere a la inclusión de comentarios en el programa fuente, al menos antes de definir cada función. Estos comentarios deben explicar el detalle sobre lo que realiza la función, las entradas, salidas y restricciones. En el código que define la interfaz gráfica, se debe identificar cada componente utilizado y qué función realiza.

En tanto, la documentación externa debe incluir la siguiente información o secciones:

1. Tabla de contenidos o índice
2. Introducción
3. Descripción del problema.
4. Análisis de resultados (incluyendo corridas de ejemplo).
5. Bitácora de actividades: se deben ir anotando todas las actividades, tipo de actividad, su descripción y duración.
6. Estadística de tiempos: un cuadro que muestre un resumen de la Bitácora de Actividades en cuanto las horas REALES invertidas como el que se muestra en la tabla 1.

Actividad realizada	Número de horas
Análisis de requerimientos	xx horas
Diseño de la aplicación y diagrama de clases	xx horas
Investigación de funciones	xx horas
Programación	xx horas
Documentación interna	xx horas
Pruebas	xx horas
Elaboración documento	xx horas
TOTAL	xx horas

Tabla 1. Ejemplo de bitácora con el registro de actividades

1. Conclusión personal

8. Evaluación

- Documentación: 15%
 - ✓ Interna: 5%
 - ✓ Externa incluyendo: 10%
- Defensa: 15%
- Resultados (ejecución, eficiencia, presentación): 70%

9. Aspectos administrativos

- La tarea se debe hacer en grupos de dos personas y se debe entregar a más tardar el martes 12 de junio del 2018 a las 11:55 p.m. en forma electrónica, en un archivo comprimido que contenga TODO lo necesario para ejecutarla.
- El formato del archivo comprimido debe seguir el siguiente formato con el formato: primera letra del nombre – primer apellido – proyecto3 (agonzalez-proyecto3).
- El correo para enviar los trabajos es antonio.gonzalez@tec.ac.cr.
- **No se aceptarán tareas después de la fecha y hora indicadas.** Debe enviarse un archivo readme.txt con la versión de Python a utilizar para la revisión y alguna otra indicación que se considere importante.
- Se puede utilizar cualquier elemento de la interfaz gráfica de Tkinter o Pygame y **la presentación será un elemento importante dentro de la calificación** (no puede usar ninguna otra biblioteca para la presentación visual).
- Se debe adjuntar la documentación solicitada en formato .pdf.
- Cualquier falta a los aspectos aquí enunciados implicará pérdida de puntos.
- En caso de probarse algún tipo de fraude en la elaboración de la tarea se aplicarán todas las medidas indicadas al inicio del curso, incluyendo una carta al expediente del estudiante.
- **El profesor se reserva el derecho de calificar forma y fondo de las actividades tomando como referencia la mejor actividad presentada**
- **Citas:** La revisión presencial de los trabajos será el miércoles 13 de junio del 2018 a la hora que sea acordada con cada grupo en la oficina 422 del edificio K1 de la Escuela de Electrónica. Las citas serán acordadas de forma oportuna.

10. Bibliografía.

Documentación técnica Python y Tkinter o Pygame.

11. Consultas.

Puede dirigir cualquier consulta a antonio.gonzalez@tec.ac.cr.