

Instituto tecnológico de Costa Rica

Carrera:

Ingeniería en Computadores

Curso:

Taller de Programación CE-1102

Proyecto programado I

Nombre:

Romario Ramírez Ramírez

Profesor:

Antonio González Torres

Índice

Tabla de contenido

Introducción	3
Descripción del problema.....	4
Análisis de resultados	4
Bitácora de actividades	10
Estadísticas de tiempos.....	12
Conclusion Personal	13

Introducción

Programar es la acción en el cual se logra dar instrucciones necesarias por medio de un código fuente para que se ejecute y se pueda llegar a solucionar el problema presente. Este código fuente puede ser escrito en diversos lenguajes de programación como lo es Python, Java, C++ y muchos otros más.

En este proyecto llevaremos a cabo la programación de un juego por medio del lenguaje de programación de Python, que si bien es cierto es un lenguaje de programación poderoso, no es tan difícil de aprender y cuenta con estructuras de datos eficientes y de alto nivel. Se presentara la descripción del problema y los diferentes métodos que se utilizaron para llegar a la solución del problema.

Descripción del problema

En este proyecto programado se presentan diversas acciones que deben ser cumplidas para llevar a cabo la ejecución de DeathRace el cual es un juego que consiste en una carrera de autos en donde se debe implementar opciones como registrar nombre de usuario el cual será el que usara el jugador y a su vez se le relacionara un puntaje que debe ser ilustrado al seleccionar esta opción.

El juego debe cumplir ciertas reglas básicas como es el tiempo de la partida que debe ser de 3 minutos que permite también que uno o dos jugadores compitan. La velocidad de los vehículos de los jugadores deben ser variable mientras que la de los vehículos controlados por la maquina es constante.

El puntaje será dado según las vueltas que se le den a la pista y de los vehículos que elimine, un competidor no puede eliminar al otro cuando se está jugando multijugador. A su vez se debe investigar acerca del uso de archivos JSON, las bibliotecas graficas de Python como Tkinter o Pygame y de la utilización de hilos (threads) para la ejecución de diversas acciones.

Debe de implementarse la opción de guardar y reiniciar partida en el menú así como también debe poseer al menos 3 niveles donde se va superando una vez finalizado los 3 minutos del cada nivel. Cada pista debe ir con obstáculos el cual aumente el grado de dificultad de las pistas. Se debe implementar archivos de sonido para aceleración, desaceleración, choques y frenado.

Análisis de resultados

Para llevar a cabo este proyecto se ha utilizado un conjunto de módulos del lenguaje de programación Python llamado Pygame, con el fin de establecer un conjunto de algoritmos para la solución del problema presentado anteriormente.

Dentro de estos algoritmos se utilizó ciclos como el “while” y el “for”, condicionales (if,elif,else) y también se utilizó el paradigma de programación orientada a objetos(POO).

Se estableció una pantalla de inicio donde se implementa el contenido del juego con sus respectivas opciones, se establece las dimensiones de la pantalla con su fondo y las indicaciones presentadas en texto como se muestra en la figura 1 en donde se muestra el código fuente y su resultado

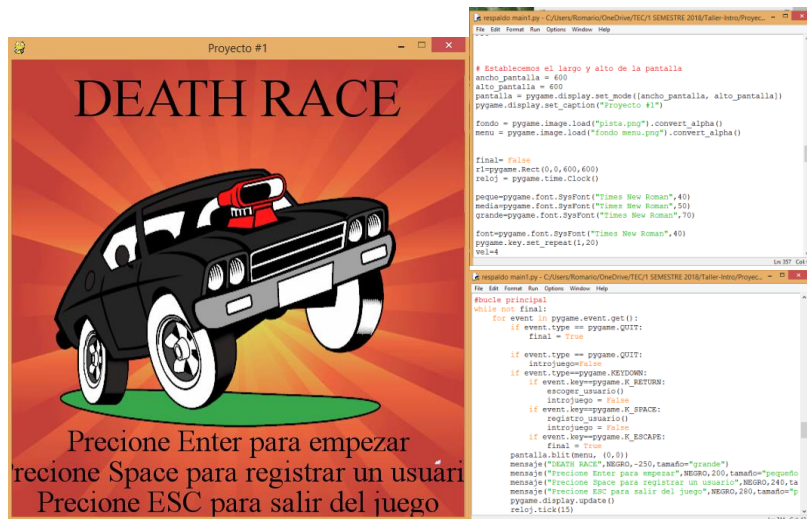


Figura1. Pantalla de inicio del juego

Se utiliza para la opción de registrar usuario un “text box” con ayuda de otras funciones que son importadas al inicio del código. Figura 2

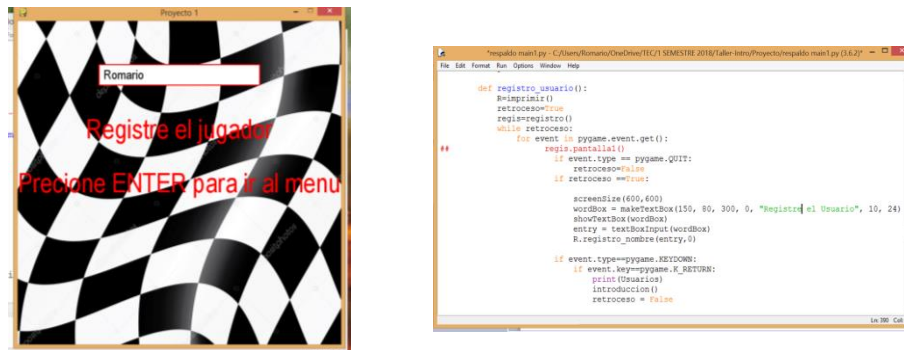


Figura 2. Registro de Usuario

Que a su vez al ser ingresado el usuario es enviado y almacenado en un archivo JSON para ser utilizado durante el juego para delegar futuros puntajes. Figura 3

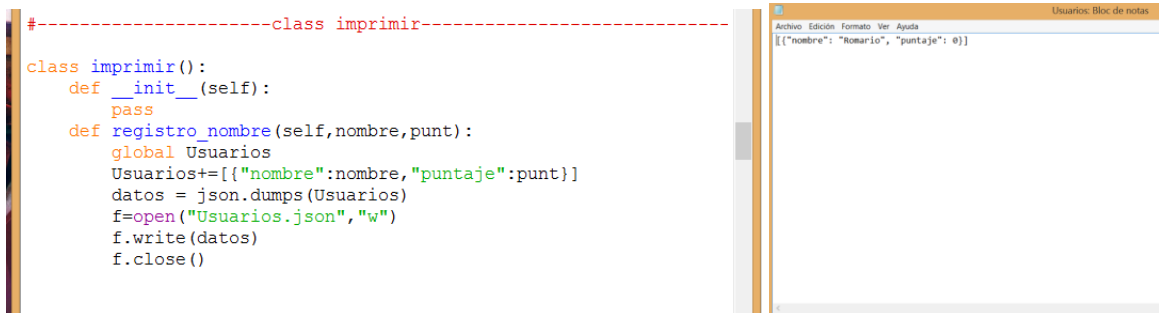


Figura 3. Almacenamiento del usuario al archivo JSON

Se establecio la funcion de enviar un mensaje al usuario si este quisiera entrar al juego sin haber algun usuario registrado. Figura 4

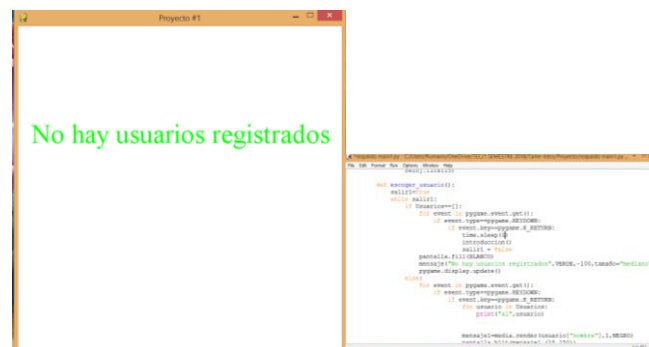


Figura 4.

Una vez ingresado el usuario y almacenado se ingresa al juego, el cual antes de empezar la partida se debería escoger el usuario a utilizar. En esta parte se logro presentar sobre la pantalla el titulo que indica que se seleccione el usuario a utilizar pero la funcion concreta no se cumple pues no se presentan los usuarios ni se logra escoger uno para seleccionarlo. Figura 5

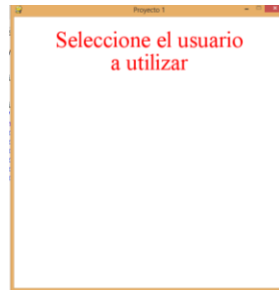


Figura 5

Una vez ingresado a la partida se logra establecer la pantalla de juego con la pista y los vehiculos. Algunos vehiculos son los manejados por la maquina y otros los que son manejados por el usuario. Se logra establecer el temporizador, no con los 3 minutos programados como se plantea en el problema, pero si funcional ya que va marcando el tiempo y cuando llega a cero termina la partida. Figura 6



Figura 6

Se logro delimitar la pista, que los autos dispararan, se trato de implementar la colicion del disparo contra los vehiculos mandados por la maquina pero no se logro. Figura 7

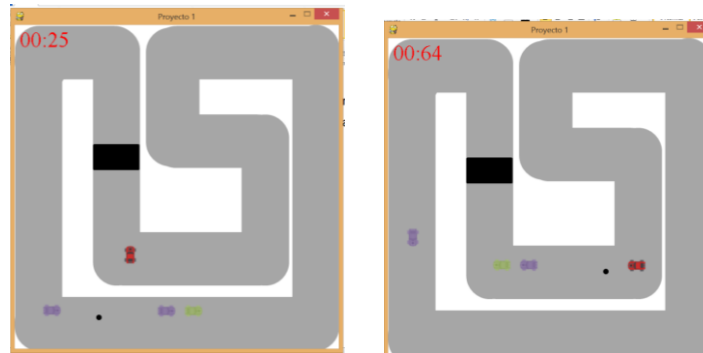


Figura 7

Se trato de limitar la pista, de incorporar las colisiones entre vehiculos y del efecto de eliminar un vehiculo, asi tambien como los otros objetivos planteados en el problema, se trataron diferentes algoritmos para lograrlo pero no se consiguio el objetivo. Se tratara de analizar detenidamente y con tiempo el problema y los posibles algoritmos para allar la solucion de los problemas.

Con este segundo avance se trato de acomodar el codigo fuente puesto que se le debia introducir mas niveles, obstaculos, entre otros. Se cambio un poco la estructura del codigo para el cual se restablecio el menu utilizando la biblioteca grafica de python Tkinter, asi tambien como el "input tex" para la funcion de registro de usuario. Figura 8

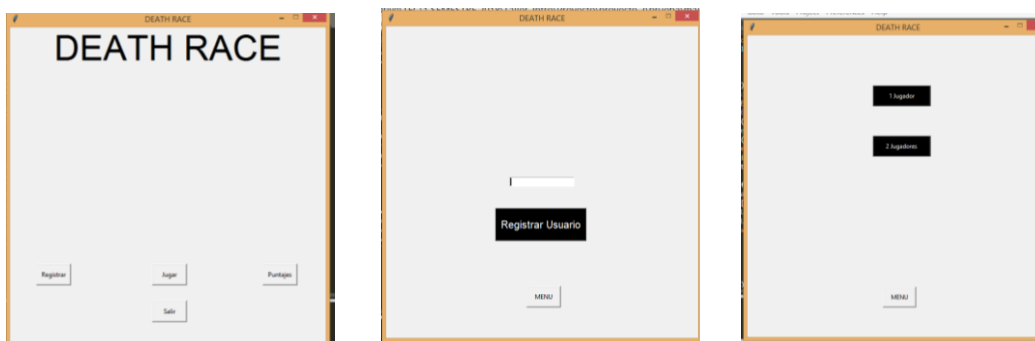


Figura 8

Se puede observar que el menu esteticamente es simple al igual que las pantallas de cada nivel, la idea es darle una apariencia mejor con animaciones y demas

pero esto demoraria tiempo el cual fue aprovechado para la solucion de otros problemas requeridos en el proyecto.

Se establecieron los limites de la pista donde si el jugador se sale de ella sufrira una disminucion de velocidad asi tambien como la disminucion de puntaje, tambien se limito la pantalla donde si el jugador sale de ella el juego acaba y por ende no lo dejaria avanzar a los siguientes niveles. Se implementaron 3 niveles que son ejecutados nivel por nivel cada ves que se finalice uno sin sufrir colisiones (figura 9). Al nivel 1 se le implementa una cancion de ambiente el cual la idea es poner una en cada nivel pero no se hizo por falta de tiempo.

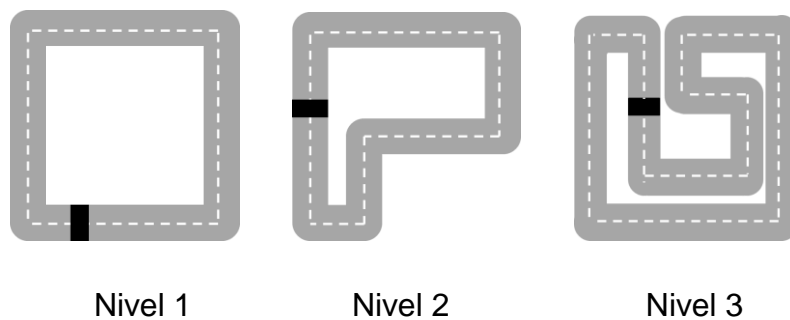


Figura 9. Niveles del juego

Se implemento efecto de sonido de aceleracion a los autos de los jugadores, asi tambien como efecto de sonido al disparo de cada jugador. Se coloco un obstaculo con el fin para mostrar la idea de cómo iran los obstaculos dentro de las pistas(figura 10), pero no se concluyo el proposito por falta de tiempo.

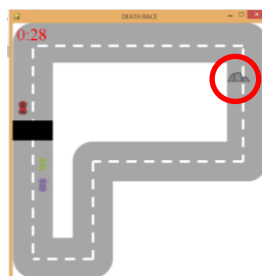


Figura 10

Bitácora de actividades

Miercoles 2 de mayo, investigacion sobre las bibliotecas graficas Tkinter y Pygame para la eleccion de una para llevar a cabo la realizacion del proyecto, con una duracion de una hora y media.

Viernes 4 de mayo, programacion de la pantalla principal y creacion de diferentes funciones. Tambien analizar diferentes algoritmos que iria a utilizar. Con una duracion de 3 horas.

Domingo 6 de mayo investigacion sobre diferentes algoritmos para poner en marcha el vehiculo utilizando documentacion de pygame como lo es key, rect, blit entre otras. Duracion 3 horas

Lunes 7 de mayo. Investigando como llevar a cabo las funciones de giro del auto, establecer el menu principal del juego ademas la rotacion de las imágenes para el giro de los autos. Duracion 8 horas. Ademas la asistencia a tutorias para la aclaracion de dudas. Duracion 2.5 horas

Martes 8 de mayo. Investigacion de cómo trabajar con archivos JSON para el almacenamiento de datos. Duracion 3 horas

Miercoles 9 de mayo. Implementacion de archivos JSON. Duracion 3 horas

Jueves 10 de mayo. Tratar el uso de clases para la facilitacion de diferentes funciones. Duracion 3 horas

Viernes 11 de mayo. Mas investigacion acerca de la implementacion de los archivos JASON ya que posteriormente no se logro adaptarlo. Duracion 5 horas.

Sabado 12 de mayo. Se detecto fallos en la eficacia de la funcion de movimiento de los vehiculos manejados por la computadora, programacion de un algoritmo mas efectivo. Duracion 4 horas

Domingo 13 de mayo, finalizacion de implementacion de los archivos JSON e investigacion de colisiones y disparos de los autos de los jugadores. Duracion 5 horas

Lunes 14 de mayo reacomodo del codigo implementacion de los disparos y la implementacion del multijugador, cambio de pista, vehiculos, implementacion de clases. Duracion 11 horas

Segundo proyecto

Miercoles 16 de mayo investigacion acerca de comandos de tkinter para la reestructuracion del menu del juego. Duracion 3 horas

Viernes 18 de mayo reacomodo de codigo con el menu y sus opciones nuevas en tkinter. Duracion 4 horas

Domingo 20 de mayo analisis de cómo implementar los nuevos niveles en el codigo fuente. Duracion 3 horas

Lunes 21 de mayo correccion de algunos errores de los inicios del juego e implementacion de algunas funciones que no se lograron colocar. Duracion 6 horas.

Miercoles 23 de mayo se empezo a investigar un poco acerca del manejo de archivos para las funciones de guardado y reiniciado de partida. Duracion 3 horas

Viernes 25 de mayo se enfoco en detallar funciones que tenian pequeñas fallas, asi como la investigacion de implementar efectos de sonidos y musica al juego. Duracion 4 horas.

Sabado 26 de mayo se implementa los efectos de sonido y se detallan unos puntos que estaban dando problema en el movimiento de los autos, en el cierre del juego y en los limites de la pista. Duracion 4 horas

Domingo 27 de mayo se terminaron unos detalles se coloco la idea del obstaculo que el cual se implementaran mas adelante. Duracion 4 horas

Estadísticas de tiempos

Primer proyecto

Análisis de requerimientos	5 horas
Diseño de la aplicación y diagrama de clases	5 horas
Investigación de funciones	11 horas
Programación	12 horas
Documentación interna	1 hora
Pruebas	14 horas
Elaboración documento	4 horas
TOTAL	52 horas

Segundo proyecto

Análisis de requerimientos	5 horas
Diseño de la aplicación y diagrama de clases	4 horas
Investigación de funciones	5 horas
Programación	6 horas
Documentación interna	1 hora
Pruebas	6 horas
Elaboración documento	4 horas
TOTAL	31 horas

Conclusion Personal

Desde mi punto de vista a sido un trabajo muy bueno pues nos obliga a la investigacion de diversas funciones, documentaciones, metodos y muchas otras cosas mas que se utilizan en la programacion. En algunos puntos de la materia si me pareceria necesario recibir aprendizaje basico sobre posibles soluciones a problemas y tambien, se sabe que la programacion es una tarea muy laboriosa, entonces al ser una experiencia nueva se torna mucho mas duro el trabajo tanto de investigacion como el de programacion.

Con el segundo proyecto, que es una actualizacion del juego (primer proyecto), puedo decir que ha sido un proyecto en el cual se aprende y se desarrolla mucho mas la destresa del manejo del lenguaje de programacion de python y de sus bibliotecas graficas tkinter y pygame. Asi tambien como la capacidad de analizar y descubrir algoritmos que sean eficientes para la solucion de problemas. Se familiariza mas con el mundo de la programacion y da una buena introduccion a este, haciendonos una idea de lo que es la programacion y todo lo que esta conlleva , para lo que viene a futuro con los siguientes cursos de programacion.