Instituto Tecnológico de Costa Rica.

Escuela de Ingeniería en Computadores.

**Curso:**

Taller de Programación.

**Actividad:**

Proyecto II de Programación.

**Realizado por:**

Fabián Guevara Camacho - 2021144297

Greivin Carrillo Rodríguez - 2021020997

Roger Solano Suárez - 2021151577

**Profesor:**

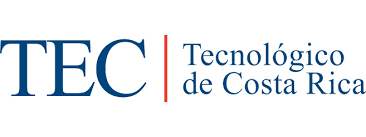
Antonio Gonzales Torres

**Grupo:**

Taller de programación, grupo 5

**Año:**

2021



**24/06/2021**

Tabla de Contenido.

[**Introducción:** 3](#_Toc75465139)

[**Descripción del problema:** 4](#_Toc75465140)

[**Análisis de resultado:** 5](#_Toc75465141)

[**Bitácora y Estadística de tiempo:** 20](#_Toc75465142)

[**Conclusión Personal:** 21](#_Toc75465143)

# **Introducción:**

El proyecto tiene como objetivo la creacion de una version 2.0 del videojuego llamado pyDakarDeath , en las cuales se elaboraran mejoras con respecto a la primera version, el cual es compatible con sistema Windows.

En la primera parte de la investigación se buscará desarrollar los conocimientos necesarios para la creación de una versión mejorada del juego, por lo cual se usara el programa Python que cuenta con una interfaz amigable para nuevos usuarios, además de que sirve como base confiable para futuros trabajos de la carrera, con respecto al juego, algunos de los puntos en los cuales se busca una mejoría serian la capacidad de registrar el nombre de un usuario nuevo, tener la posibilidad de ver puntajes, además de iniciar, guardar y reiniciar las partidas cuantas veces sea necesario, con el fin de obtener una experiencia más óptima para el jugador.

Como se menciono anteriormente se utilizara el programa Python para realizar todo el desarrollo técnico del videojuego, además se hará uso de TKinter el cual nos brindara una biblioteca que nos permitirá abrir diversas ventanas para iniciar el videojuego. Asimismo, se utilizó la plataforma Pygame para gran parte del desarrollo del videojuego, esta interfaz es recomendable para iniciar en la programación de videojuegos.

En la primera parte de la programación se establecieron las variables fijas las cuales se usaron para establecer el tamaño de la ventana del juego, ademas se utiliza la biblioteca de Pygame para establecer un nuevo título y logo para la ventana inicial, asimismo se exporta un archivo de música el cual se fijara en loop para darle una mayor ambientación al videojuego.

Además, se usaran hilos (threads) y diversas herramientas sobre la programación orientada a objetos, uso avanzado de archivos JSON y utilización de bibliotecas de sonido, entre otras instrumentos que nos brindaran una amplia gama de posibilidades para desarrollar mejoras en el juego, así como favorecer las facultades en el proceso de aprendizaje de los estudiantes.

# **Descripción del problema:**

La versión anterior no contaba con diversas funciones que harían más amena la experiencia del jugador ahora tiene sistema de sonido, más oponentes y puntuación y algunos errores puntuales que tenía fueron solucionados.

Ahora la movilidad del auto fue mejorada ya que este ni se movía correctamente y hacia que se detuviera el programa, también cuenta con niveles de dificultad que pondrá a prueba al jugador

pyDakarDeath es un juego de carrera de automóviles en el que pueden competir dos

más jugadores, en la que ambos jugadores usan la misma computadora. El juego utilizará una

arquitectura cliente-servidor, en la cual el tablero es controlado por el servidor y desplegado

por cada cliente en su computadora. Las acciones que realizan los clientes son enviadas al

servidor y este actualiza los tableros de todos los jugadores participantes. El formato de la

pista y el diseño de los carros lo deben ser definidos y corregidos.

En este juego el competidor debe sobrepasar o eliminar a los oponentes que son

controlados por el mismo juego

# **Análisis de resultado:**

Texto

Descripción generada automáticamente

Se importan las bibliotecas necesarias y se crean variables globales para su utilización más adelante.

Texto

Descripción generada automáticamente

Se crean funciones que reproducen archivos .wav para poder utilizar música en el juego, se usan distintas funciones para la música del menú y la del juego. También se crea una funciona para cargar las imágenes con la implementación de una ruta la cual es una carpeta que contiene las imágenes requeridas.

Texto

Descripción generada automáticamente

Esta función regresa al menú principal y reinicia las variables modificadas durante la aplicación del juego.

Texto

Descripción generada automáticamente

Esta siguiente función verifica si el jugador ha perdido todas sus vidas y si es así, tira el ‘Fin del Juego’ y un botón para regresar al menú. También obtiene los datos de el nombre y puntaje y los guarda en un archivo .txt.

Texto

Descripción generada automáticamente

Se crean clases con funciones para el movimiento de los enemigos y los obstaculos, estas funciones tambien detectan las colisiones entre objetos y si se ternima el juego borra todos los elementos de la pantalla

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

La clase estrella y roca se diferencian porque la roca al chocar con ella se le restan 2 vidas al jugador en lugar de 1 como si hubiera chocado con un enemigo normal. Y el objeto estrella si se choca con el se suma una vida al jugador y se otorgan puntos extra.

Texto

Descripción generada automáticamente

La clase Fondo es para el movimiento del fondo, lo que crea un loop de la imagen de fondo y crea la ilusión de que la pista es infinita, pero solo se utilizan 2 imágenes. Las cuales se mueven en el eje y de la ventana creada.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Luego se procede a crean los objetos de las clases para su utilización en las funciones del Juego Principal.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Se crea una función controladora que otorga puntos con el tiempo, pero más importante se encarga de la creación de los enemigos y obstáculos dentro del juego con la implementación de Threads para realizar varios procesos a la vez

Texto

Descripción generada automáticamenteTexto

Descripción generada automáticamente

Luego se crea la función de aplicación del juego, la cual hace un llamado a la función de imágenes para cargar las ilustraciones de los enemigos, obstáculos, fondos y jugador. Posteriormente se crean dentro de la ventana del juego una columna con información para que el jugador este al tanto durante el transcurso de la partida. Estos datos son la velocidad del jugador, su nombre, puntaje y la cantidad de vidas restantes.

Pantalla de computadora con letras

Descripción generada automáticamente con confianza mediaTexto

Descripción generada automáticamente

Dentro de la función juego, se crea otra función que es la encargada del movimiento del jugar y el aumento de su velocidad para tirarlo en pantalla.

Texto

Descripción generada automáticamente

Todavía dentro de la función juego, pero afuera de la función de movimiento del jugador se crean vínculos al presionar las teclas WASD para su utilización la función mover y posteriormente se crean nuevos Treads encargados del movimiento del fondo, reproducción de la música del juego y de la función controlador.

Texto

Descripción generada automáticamente

La función validar se utiliza para que el jugador no pueda iniciar el juego sin haber brindado un nombre y una verificación de estar listo. Y si lo intenta tira ventanas para hacerle saber que necesita hacer.

Texto

Descripción generada automáticamente

La función velocidad cambia la velocidad de los enemigos y obstáculos con respecto a los puntos que haya obtenido el jugador. Lo cual nosotros lo vemos como un cambio de nivel.

Texto

Descripción generada automáticamente

Se crean 4 funcioes con funcionamiento similar, las cuales son botones en el menu que nos proporcionan informacion como los creditos, como jugar, informacion basica del juego o los mejores jugadores. Esta info se puede observar al presionar 1 de los botones en el menu principal del juego y estos nos tiran una nueva ventana con la informacion requerida.

Texto

Descripción generada automáticamenteTexto

Descripción generada automáticamente

Después se crean funciones que nos ayudaran a obtener la información de los mejores jugadores de un archivo externo, estas funciones leen el archivo y modifican el archivo para ordenar las mejores 6 puntuaciones y una función auxiliar crea una lista con esas mejores puntuaciones escribe las nuevas puntuaciones y si es mayor que alguna en el top 6 reordena el orden y elimina la puntuación menor. También se crean funciones que toman los datos del nombre y puntuación del jugador para poder utilizarlos para comparar los mejores puntajes.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Finalmente se crea la ventana del menú con sus respectivos botones para la implementación de las funciones de información, de validación y la iniciadora del juego.

# **Bitácora y Estadística de tiempo:**

|  |  |
| --- | --- |
| Análisis de requerimientos | 14 horas |
| Diseño de la aplicación y diagrama de clases | 13 horas |
| Investigación de funciones | 20 horas |
| Programación | 35 horas |
| Documentación interna | 7 horas |
| Pruebas | 7 horas |
| Elaboración documento | 6 horas |
| **TOTAL** | 80 horas |

Lunes 7 de junio nos reunimos a las 7:00pm a 11:40 pm.

Marte 8 de junio organizamos las ideas y empezamos investigando.

Miércoles 9 de junio seguimos investigando.

Jueves 10 de junio seguimos investigando.

Viernes 11 de junio empezamos a poner en práctica lo aprendido.

Sábado 12 de junio hacer que aparezca la ventana y corra cosas básicas como el movimiento.

Domingo 13 de junio hacer que aparezca la ventana y corra cosas básicas como el movimiento.

Lunes 14 de junio investigamos sobre los archivos JSON.

Martes 15 de junio agregar la música y el puntaje.

Miércoles 16 de junio intentar agregar el tiempo.

Jueves 17 de junio intentar agregar el tiempo.

Viernes 18 de junio los niveles.

Sábado 19 de junio terminación de código, documentación y afinación de los últimos detalles.

Domingo 20 de junio afinando detalles.

Lunes 21 de junio verificando la documentación y el código.

Martes 22 de junio arreglando detalles.

Miércoles 23 de junio dando los últimos retoques.

Jueves 24 de junio entrega.

# **Conclusión Personal:**

**Roger Solano Suárez**: En conclusión, al programar el segundo videojuego me dí cuenta que sigue siendo un proceso largo y lento, tiene que ser con buen tiempo y bien determinado. Me dí cuenta que tenemos que poner el codigo mejor para que se vea ordenado, bonito y formado con clases. Descubrí el uso de sprites en los videojuegos que son muy importantes en el desarrollo de videojuegos. Tambien que sin tener muchos recursos o conocimiento uno puede realizar cosas demasiado interesantes con la ayuda de otros.

**Greivin Carrillo Rodríguez:** A diferencia del primer juego, en este nos encontramos con nuevos problemas, pero también teníamos conocimientos nuevos que nos ayudaron en la resolución de dichos problemas. También observe que tuvimos un mejor manejo del tiempo y una mejor comunicación; pero también nos resulto mas complicado con los horarios y los trabajos y exámenes de otros cursos que nos limitaron en el tema de las reuniones, sin embargo, se logro llegar a un buen resultado.

**Fabian Guevara Camacho:** En lo personal lo que mas me gusto fue la hora de poner el fondo porque siento que lo hace mas original lo que mas me costo fue poner las imagenes de los autos y los enemigos. Porque aveces el codigo no las reconocia aunque fue dificil creo que hicimos un buen trabajo.