**Nombre:**

**Carlos Andrés Sibaja Chinchilla**

**Carné:**

**2018280833**

**Profesor:**

**Antonio González Torres**

**Grupo 4**

**Introducción a la Programación**

**I Semestre**

Tabla de Contenidos:

[Introducción: 3](#_Toc514098620)

[Descripción del Problema: 3](#_Toc514098621)

[Análisis de resultados: 3](#_Toc514098622)

[Bitácora: 5](#_Toc514098623)

[Estadística de Tiempos: 5](#_Toc514098624)

[Conclusión: 6](#_Toc514098625)

# Introducción:

PyDeathRace es un videojuego de carreras en circuito con su base en Pygame y Tkinter que consta de mecánicas de manejo como aceleración y cambio de dirección que permiten al jugador desplazarse por toda la pantalla y que además posee bots los cuales deben ser eliminados por el jugador con un sistema de disparo diseñado, obteniendo puntaje una vez elimine un bot en la pista o también cada vez que de una vuelva completa a la misma.

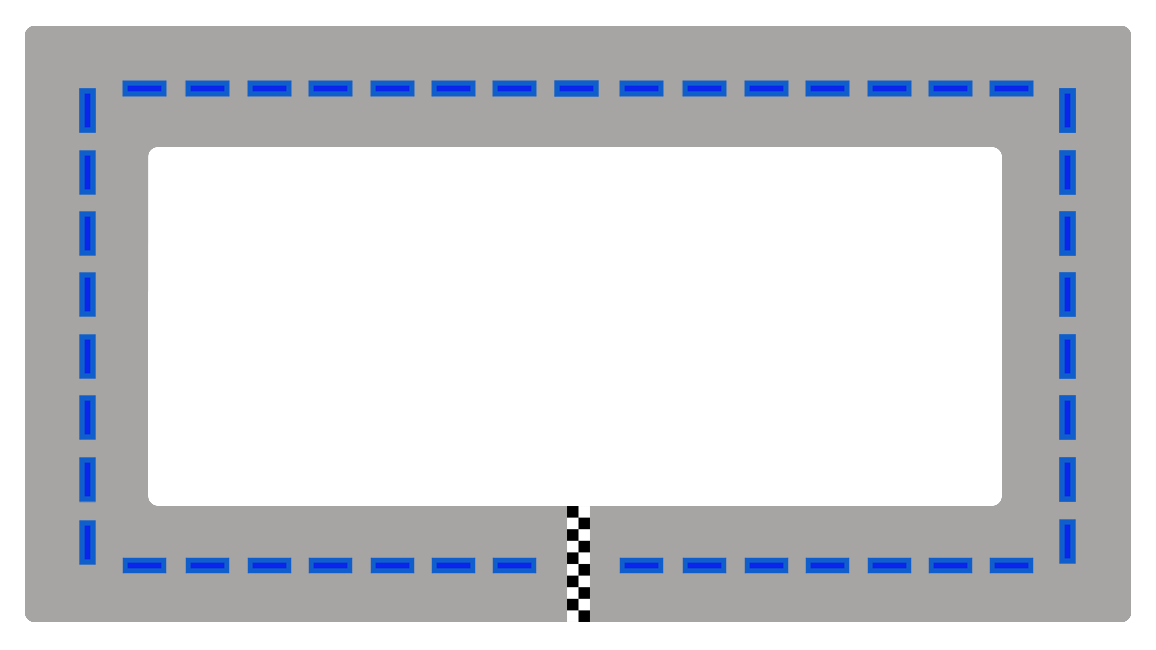
# Descripción del Problema:

El diseño de un videojuego de carreras mejorado que ahora incluya un sistema de niveles pasado el tiempo establecido como nuevos obstáculos alrededor de las diferentes pistas, aumentando su número por nivel para incrementar la dificultad y además un sistema de sonido por niveles. Además, de que el juego debe finalizar al chocar con los bordes.

# Análisis de resultados:

Para esta etapa del videojuego inicialmente se establecen los límites de la pista y los bordes del videojuego donde si el auto del jugador se sale de la pista automáticamente comienza a restar puntos de su puntuación total con el objetivo de darle al jugador un reto mayor.

  
 Esto es posible gracias a que un sistema de máscaras que una vez que choque con el jugador activa una variable que resta la puntuación del jugador, colocando 2 imágenes por separado que se hacer ver como una misma y que permite separarlo como una variable de posicionamiento.

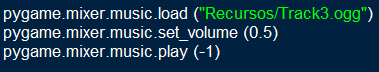


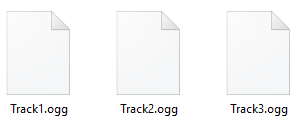
Además de este sistema, se desarrolló un sistema de colisiones con objetos diseñados por mí mismo, los cuales tienen cada uno una característica diferente para que el jugador se vea en la obligación de maniobrar para evitarlos y lograr aumentar su puntuación, donde el cono, quita 100 puntos en la puntuación total del jugador, la mancha de gasolina disminuye en una gran medida la velocidad del automóvil y la roca lo hace revotar con el objetivo de que no pueda avanzar hasta que cambie su recorrido, con lo cual ya se tienen diversas herramientas para aumentar la dificultad en pantalla.



 Además de estos objetos, se agregó un nuevo display de puntos y tiempo en pantalla para que el jugador tenga siempre presente a que se refiere cada valor y sepa cuanto tiempo de juego hay restante en todo momento, esto para mejorar la calidad de juego.

Sumado a esto, se agregó la música a cada nivel del videojuego para lograr disfrutar más de los niveles, con temáticas similares al estilo grafico del videojuego escogido. Para esto se sigue el siguiente código:





# Bitácora:

Para esta etapa del proyecto primeramente analicé detalladamente lo que el profesor esperaba para este avance, logrando entender la mayoría de los requerimientos para posterior a esto comenzar a con la etapa de investigación donde detenidamente comencé a buscar en fuentes de internet para de esta forma tener ejemplos de, por ejemplo, el sistema de colisión y la música del nivel, aplicando un alrededor de unas 5 horas en su totalidad a esta parte del proyecto. Viendo algunos ejemplos y con el código ya existente comencé a editarlo y también me di a la tarea de diseñar cada aspecto grafico del proyecto, donde hice los objetos de colisión, niveles, texto con la ayuda de Photoshop, para esta parte del proyecto invertí un alrededor de unas 5 horas por igual. Escribiendo todo el código y analizando todos los problemas encontrados invertí un alrededor 12, agregando instancias de objetos, colisiones, puntuación y principalmente el manejo dummys en las otras pistas ya que el recorrido era algo más desafiante en el diseño del nivel 2 y 3. Luego para finalizar invertí en la parte de documentación, tanto interna como externa un total de 3 horas, definiendo bien cada detalle agregado al proyecto.

Estadística de Tiempos:

|  |  |
| --- | --- |
| Análisis de Requerimientos | 2 Horas |
| Diseño de la aplicación y diagrama de clases | 3 Horas |
| Investigación de funciones | 2 Horas |
| Programación | 13 Horas |
| Documentación Interna | 1 Hora |
| Pruebas | 1 Hora |
| Elaboración del documento | 2 Horas |
| **Total** | 24 Horas |

# Conclusión:

A manera de resumen, esta fase del proyecto fue divertida, el agregar los obstáculos y la música fue algo practico de hacer y disfrute bastante en su desarrollo, además, aprendí de todos los módulos de guardado y de reproducción de música que posee pygame que es algo nuevo para mí, lo que me hace pensar lo practico que pueden ser estas opciones de diseño a la hora de realizar un videojuego a cuenta propia y también como nos permite explotar la creatividad mientras se aprende algo nuevo en el entorno de programación, el problema enfrentado, como en la etapa anterior, es la falta de fuentes explicativas de cómo realizar algunos métodos como el de guardado específicamente para pygame, lo que afecto el desarrollo del proyecto.

# Bibliografia:

KidsCanCode (2016) Game Development with Pygame [Archivo de Video]. Recuperado de: https://www.youtube.com/playlist?list=PLsk-HSGFjnaH5yghzu7PcOzm9NhsW0Urw

ChellingTutorials (2014) Programacion de Juegos Pygame [Archivo de Video]. Recuperado de: https://www.youtube.com/playlist?list=PL46E99FE946C1C946