

**Instituto Tecnológico de Costa Rica**

**Ingeniería en Computadores**

**Asignatura:**

**Taller de Programación CE 1102**

**Profesor:**

**Antonio Gonzales Torres**

**Trabajo:**

**Proyecto número 1, Death Race**

**Estudiante:**

**Mario Alexis Araya Chacón 2018319178**

**Año:**

**2018**

## Índice

Introducción .....	3
Descripción del problema.....	4
Análisis de resultados .....	5
Bitácora .....	6
Conclusión .....	7
Bibliografía .....	8

## Introducción

Esta es una actualización de la versión anterior de Death Race, la cual cuenta con mejoras en varios aspectos. Entre ellos están:

La creación de dos nuevos niveles con distintas dificultades, en estos nuevos niveles uno es más sencillo que el original, el cual es una pista cuadrada sin obstáculos, y el otro es una pista con más curvas que la original que presenta obstáculos como muñecos de nieve, bolas de nieve y picos de hielo que sirven de obstáculo para el usuario y actúan como paredes que debe esquivar.

Otro aspecto mejorado fue la implementación de sonidos como la música de fondo que hay presente en cada nivel y cambia según el usuario vaya completando niveles, y los sonidos de las explosiones los cuales se activan en el momento en que un carro colisiona ya sea con una pared o un misil.

También cuenta con las funciones de pausar la partida al presionar la tecla “p” del teclado, guardar la partida al presionar “g” y un botón en el menú principal que sirve para cargar la partida que fue previamente guardada. Solamente se puede tener una partida guardada ya que se sobrescriben los datos en el archivo de texto cada vez que se presiona la “g”.

Además, entre los cambios realizados se encuentra el cambio de muchas variables locales a globales como los jugadores y los niveles, esto con el fin de poder guardar los datos de la partida deseada con mayor facilidad.

Finalmente, se creó una nueva ventana en la cual se muestran los puntajes de los mejores tres jugadores basado en la puntuación obtenida por cada uno en las partidas jugadas. Esta ventana puede ser accesada desde el menú principal por medio de un botón o bien cuando el jugador llega al nivel 3 y elimina a todos los carros enemigos.

### Descripción del problema

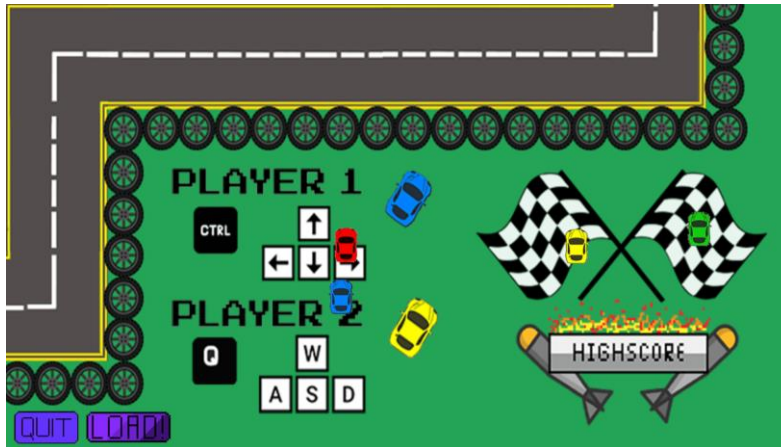
En este proyecto se presentaron varios problemas, entre los cuales se encuentran problemas a la hora de pausar el juego, guardar los archivos de texto y cargar el juego.

En primer lugar, se presentaba el problema de que a la hora de pausar el juego el tiempo seguía corriendo, los carros se detenían, pero los misiles seguían viajando y podían impactar a los carros inmóviles. Esto se solucionó al crear una variable global llama PAUSA, la cual si esta activada no permite que ningún objeto ya sea de tipo carro o misil pueda cambiar su posición, además la función que ejecuta la cuenta regresiva del tiempo también no se ejecuta si PAUSA no lo permite.

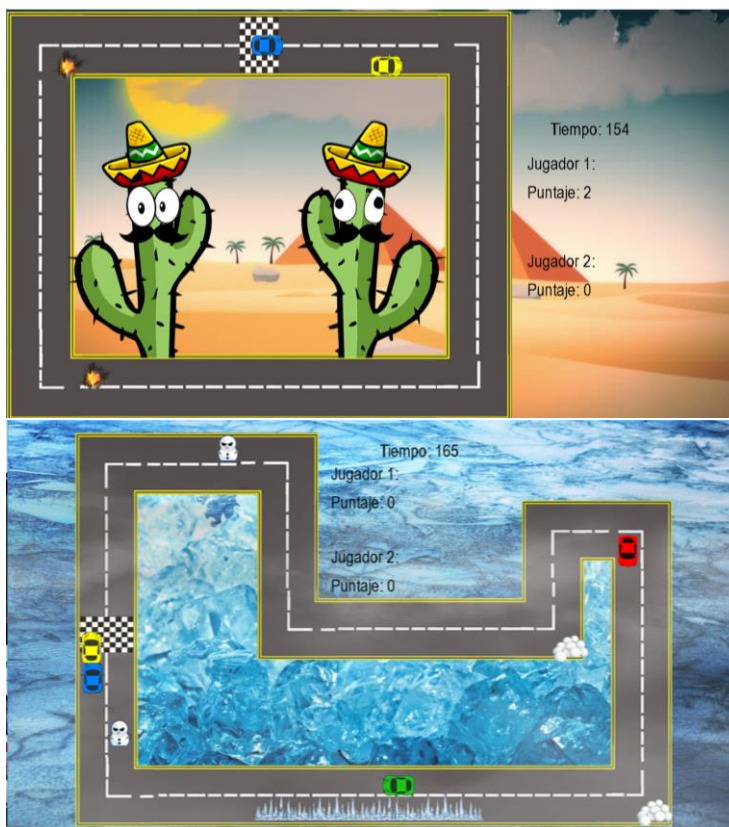
Por otro lado, el guardado del juego presentaba varios problemas. Ya que los archivos de texto utilizados no son trabajados en formato JSON, sino que son trabajados de manera que las variables necesarias son convertidas a Strings y guardadas en un documento de tipo “.txt” separándolas por el símbolo “@” lo cual permite que puedan ser trabajadas como una lista a la hora de leerlas. Esto presentó varios problemas ya que se tuvieron que convertir las variables al tipo que es aceptado por el archivo de texto, y a la hora de importarlas volverlas a convertir al tipo que necesita la clase Carro.

Finalmente, el mayor problema se presenta a la hora de cargar el nivel. Esto se debe a que los carros se crean correctamente con los datos previamente guardados pero no se carga el nivel con correctamente con ellos por lo que no se pueden mover.

## Análisis de resultados



El principal error que presenta el juego es a la hora de cargar como ya se mencionó en la descripción de los problemas, que los carros se generan pero el nivel no carga bien y no permite moverlos. Además, otro problema presente es el puntaje de los jugadores que no se conserva cuando el usuario pasa de nivel, sino que se reinicia a 0.



## Bitácora

Actividad	Fecha
Creación de las nuevas imágenes de fondo en Photoshop.	5/15/2018
Delimitación de las paredes y los mensajes a mostrar en las pantallas.	5/16/2018
Delimitación de paredes, creación de obstáculos y arreglo de las rotaciones de los bots en los nuevos niveles.	5/17/2018
Búsqueda, descarga, importación e implementación de los sonidos en el código.	5/21/2018
Modificación del código en cuanto a la implementación de variables globales para el cambio de nivel y sus funciones correspondientes.	5/22/2018
Delimitación de paredes de los nuevos niveles y creación de las rutas que deben seguir los bots según el nivel y mejora de los misiles.	5/23/2018
Corrección de errores en la función de puntuación, creación del botón para pausar el juego y modificación de las funciones que mueven los objetos.	5/24/2018
Creación de funciones para guardar y cargar la partida. Creación de la imagen de fondo para la pantalla del puntaje.	5/25/2018
Corrección de errores en las funciones de guardar y cargar el nivel. Modificación de la forma de guardado de la lista.	5/26/2018
Creación de la documentación externa	5/27/2018

Actividad	Horas invertidas
Creación de fondos en Photoshop	3 horas y media
Delimitación de paredes y mensajes en pantalla	1 hora y media
Delimitación de paredes, obstáculos y rotación de bots	2 horas y media
Descarga e importación de sonidos	1 hora
Modificación del código e implementación de variable para el cambio de nivel	3 horas
Corrección de paredes de los niveles y creación de rutas para los bots	3 horas
Corrección de los errores en la puntuación, creación del botón para pausar	2 horas
Creación de funciones de guardado, carga y el fondo para el puntaje	3 horas y media
Corrección de errores de guarda y cargar el nivel	4 horas
Creación de la documentación	3 horas

## Conclusión

Con la elaboración de este proyecto se logro la mejora del código anterior, la comprensión del funcionamiento de sonidos en pygame y las variables globales.

Además, se logró comprender de mejor manera la forma en la que funcionan los archivos de texto y la importancia que estos tienen en la creación de un juego como lo es para guardar niveles y puntaje. También ayudo para la comprensión del porque es más sencillo trabajar con clases a la hora de programar ya que facilita y agiliza la creación del código evitando líneas innecesarias al poder reutilizar el código y modificarlo desde cualquier parte.

Finalmente, el proyecto ayudo para entender el uso de pygame, sus ventajas y desventajas, las cuales se pueden solucionar complementándolo con tkinter en el caso de entradas de texto.

## Bibliografía

Craven, P. V. (n.d.). *Programar juegos arcade con Python y Pygame*. Retrieved from programarcadegames:  
[http://programarcadegames.com/python\\_examples/show\\_file.php?file=move\\_with\\_walls\\_example.py](http://programarcadegames.com/python_examples/show_file.php?file=move_with_walls_example.py)

Pygame. (n.d.). *Pygame Documentation*. Retrieved from Pygame: <https://www.pygame.org/docs/>

Paget Teaches. (2017/01/09). Pygame\_functions:Text Outputs and Inputs. Retrieved from <https://www.youtube.com/watch?v=G933SOLWbbg>

Sentdex (2014/08/27). Game Development in Python 3 with PyGame – 1- Intro. Retrieved from [https://www.youtube.com/watch?v=ujOTNg17Ljl&list=PLQVvva0QuDdLkP8MrOXLe\\_rKuf6r80KO](https://www.youtube.com/watch?v=ujOTNg17Ljl&list=PLQVvva0QuDdLkP8MrOXLe_rKuf6r80KO)

Codigofacilito (2014/09/22). Curso de Pygame. Retrieved from [https://www.youtube.com/watch?v=PGRhWuYjPdw&list=PLpOqH6AE0tNherBf6bzGiDM1uIy\\_E0WJH](https://www.youtube.com/watch?v=PGRhWuYjPdw&list=PLpOqH6AE0tNherBf6bzGiDM1uIy_E0WJH)