

# **Tecnológico de Costa Rica**

**Ingeniería en computadores  
CE-1102 Taller de Programación**

***pyDakarDeath***

**Profesor:**

**Antonio González Torres**

**Estudiante:**

**Kai Sebastian Chaves Alfaro**

**Braulio Retana Murillo**

**I Semestre, 2021**

# Índice

<b>Índice</b>	<b>1</b>
<b>Introducción</b>	<b>2</b>
<b>Descripción del problema</b>	<b>3</b>
<b>Análisis de resultados</b>	<b>4</b>
<b>Bitácora de actividades</b>	<b>5</b>
<b>Estadística de tiempos</b>	<b>6</b>
<b>Conclusión</b>	<b>7</b>

## **Introducción**

A lo largo de este proyecto se investigó cómo utilizar archivos JSON así como conseguir que la actividad de los jugadores se refleje en otras ventanas además de la principal mediante el uso de sockets.

Como parte de las mecánicas de juego y reglas se planteó la existencia de vehículos manejados por la computadora que correrán a una velocidad constante, estos podrían ser destruidos por los jugadores para ganar puntos, sin embargo los jugador no podrán golpear el auto de su contrincante. Otro elemento incluido fueron unos cactus, ya que el juego tiene lugar en el desierto, los cuales bajan puntos al jugador en caso de que lo toquen.

## **Descripción del problema**

Entendimiento del uso de Pygame, archivos de sonido, archivos JSON, sockets, programación orientada a objetos e hilos para gestionar el comportamiento de cada objeto gráfico, con el fin de la creación de un juego, en este caso de autos, así como el uso de físicas de juego para los autos y acciones de los jugadores, cambios de nivel de juego y puntaje.

## **Análisis de resultados**

Primeramente con pantallas simples que únicamente tendrían un fondo negro y los autos de ambos jugadores se realizó la creación y se comprobó el funcionamiento del servidor que se utilizaría para el juego.

Lo segundo en realizarse fue el código para guardar varias puntuaciones sin que se sobreescribiera la puntuación anteriormente guardada, para esto se utilizó igualmente una pantalla simple con un botón rojo que detectaría las veces que sería clickeado, así como el nombre del jugador que se solicitaría antes de iniciar. Además de esto se creó un código que permitiría ordenar las puntuaciones de mayor a menor y este se probaría directamente en la consola.

Una vez se había conseguido guardar las puntuaciones y abrir dos ventanas que se reflejaran se continuó con la programación del resto del juego, niveles, dummyscars, cactus, fondos, sonidos, etc.

Por último se añadió el puntaje y tiempo a cada partida, así cuando se termina el tiempo del nivel se avanzaría al siguiente, el puntaje se guarda en un dos documentos tipo *.txt* , uno se guardaría al finalizar el nivel tres y sería el que se utiliza para la calificación en la leaderboard, el otro es guardado durante cualquier punto de la partida para poder recargar la misma más tarde.

## Bitácora de actividades

Fecha de reunión	Descripción de Actividades
13/6/2021	Distribución de tareas e investigación de códigos y elementos para programar
15/6/2021	Se creó el servidor a utilizar
18/6/2021	Se realizó el código de las puntuaciones.
20/6/2021	Se realizó para comentar los avances
21/6/2021	Se creó el código del juego y el menú, también se implementó el servidor al código
23/6/2021	Se realizó los detalles finales al proyecto
24/6/2021	Documentación externa del proyecto.

### Estadística de tiempos

Análisis de requerimientos	3 horas
Diseño de la aplicación y diagrama de clases	3 horas
Investigación de funciones	5 horas
Programación	20 horas
Documentación interna	1 horas
Pruebas	6 horas
Elaboración documento	2 horas
<b>TOTAL</b>	<b>40 horas</b>

## **Conclusión**

Al finalizar este proyecto logramos entender un poco mejor el funcionamiento de los servidores, esto gracias a los sockets, también logramos entender mejor cómo guardar datos en archivos de texto. Otra cosa que nos queda de este proyecto es la gran cantidad de usos que se le pueden dar a la librería pygame, ya sea crea un juego simple como puede ser “Snake” o una infinidad de juegos más elaborados.