

Tarea programada # 2:

Documentación

Taller de programación

Instituto Tecnológico de Costa Rica

Estudiantes: Andrés Uriza Lazo – 2021466844  
Jose Pablo Esquetini Fallas - 2021035767

Profesor: Jeff Schmidt Peralta

Fecha: 18 de junio de 2021

## **Tabla de contenidos:**

<b>Introducción:</b> .....	3
<b>Reglas de grupo y roles:</b> .....	3
<b>Descripción del problema:</b> .....	4
<b>Modelo de objetos:</b> .....	10
diagramas de clase y diagramas de secuencia.....	10
<b>Dificultades encontradas:</b> .....	12
<b>Bitácora de actividades:</b> .....	14
Estadística de tiempos: .....	15
Conclusión: de cada uno.....	16
Video .....	17
<b>Fuentes consultadas:</b> .....	21

## **Introducción:**

En el siguiente documento se explicará como fue nuestro proceso al colaborar para realizar el proyecto. El documento incluye información sobre cómo nos comunicamos y organizamos nuestros roles, al igual que las duraciones para cada una de nuestras tareas seguidas de una representación grafica de las estadísticas de los tiempos. También información sobre las dificultades que se nos presentaron y qué hicimos para superarlas, y un análisis de los resultados obtenidos del código elaborado.

Complementarios al código en sí se incluyen los diagramas de clases y de secuencia correspondientes a cada objeto utilizado, al igual que las referencias bibliográficas que fueron utilizadas para la lógica del código.

## **Reglas de grupo y roles:**

No especificamos reglas explícitas, más que tener buena comunicación y cumplir en lo que habíamos acordado, pero no se presentaron problemas a la hora de colaborar en el proyecto.

Se definieron los roles para Andrés como el encargado de escoger la temática del juego junto con Jose Pablo, de realizar la esquemática del menú, el movimiento del jugador, y la definición básica de los objetos de ambos; también la implementación del algoritmo Quicksort y la implementación de los sonidos y música para el juego. Al igual que la documentación externa.

Para Jose Pablo se definieron los roles como el encargado de escoger la temática del juego junto con Andrés, de refinar las partes de interfaz implementadas por Andrés, la

implementación de los objetos para cada instancia de nivel, al igual que los canvas utilizados por cada nivel, el menú de about, y la lógica de colisión de los obstáculos con el jugador, al igual que la lógica del movimiento de los obstáculos. En la parte de documentación externa, se encargó de realizar el video.

### **Descripción del problema:**

Se ocupaba realizar un juego que tiene como temática el controlar a un ninja que se encuentra en una jungla y esquiva cuchillas circulares que son lanzadas hacia el jugador desde algún borde aleatorio de la pantalla. Consta de 3 niveles, el más fácil siendo el primero y el más difícil siendo el tercero, y cada uno tiene modificaciones en el ritmo de aparición de obstáculos para aumentar la dificultad. El juego debe además registrar los puntajes obtenidos y mostrar en orden descendiente los 10 mejores obtenidos.

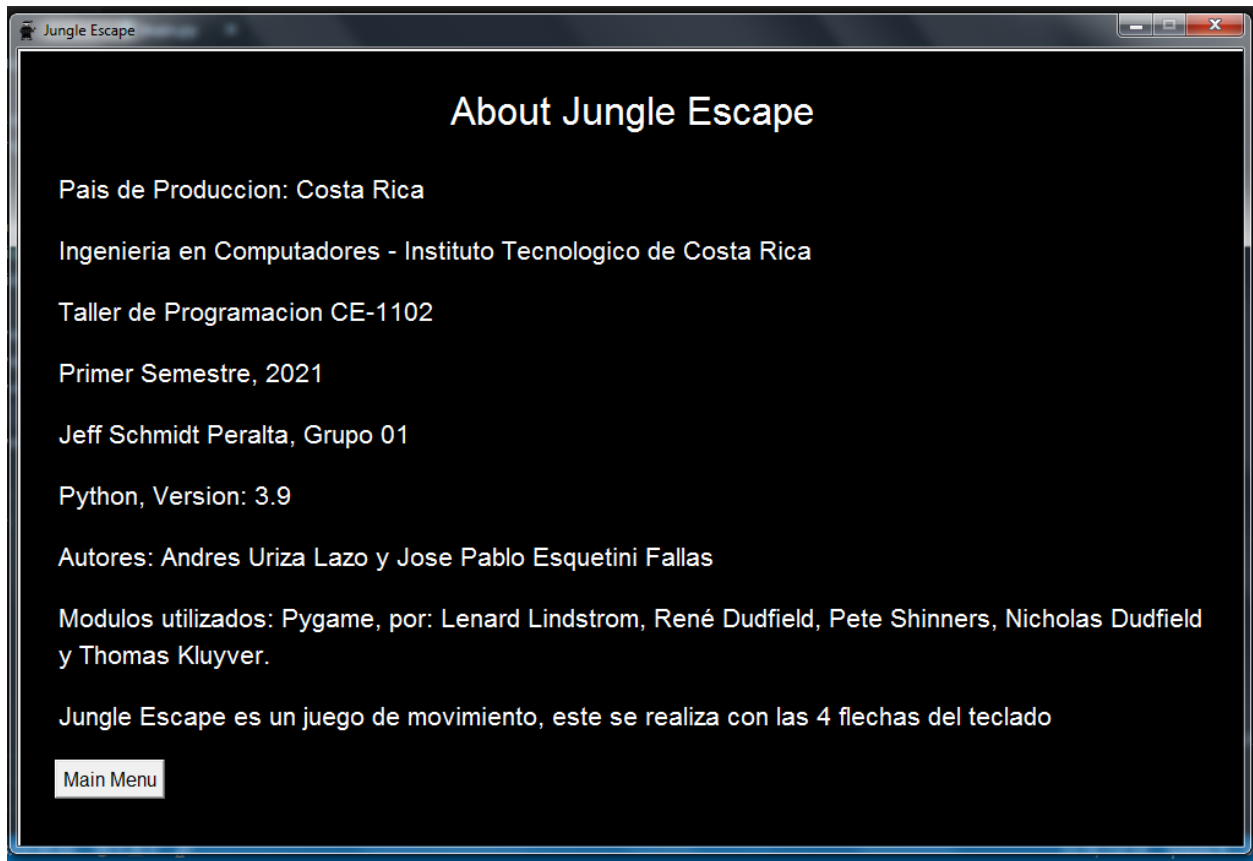
### **Análisis de resultados:**

Todo lo especificado en las indicaciones del proyecto se pudieron cumplir en el código del juego y este es funcional en su totalidad. A continuación, se presentan capturas de pantalla de las diferentes pantallas del juego.

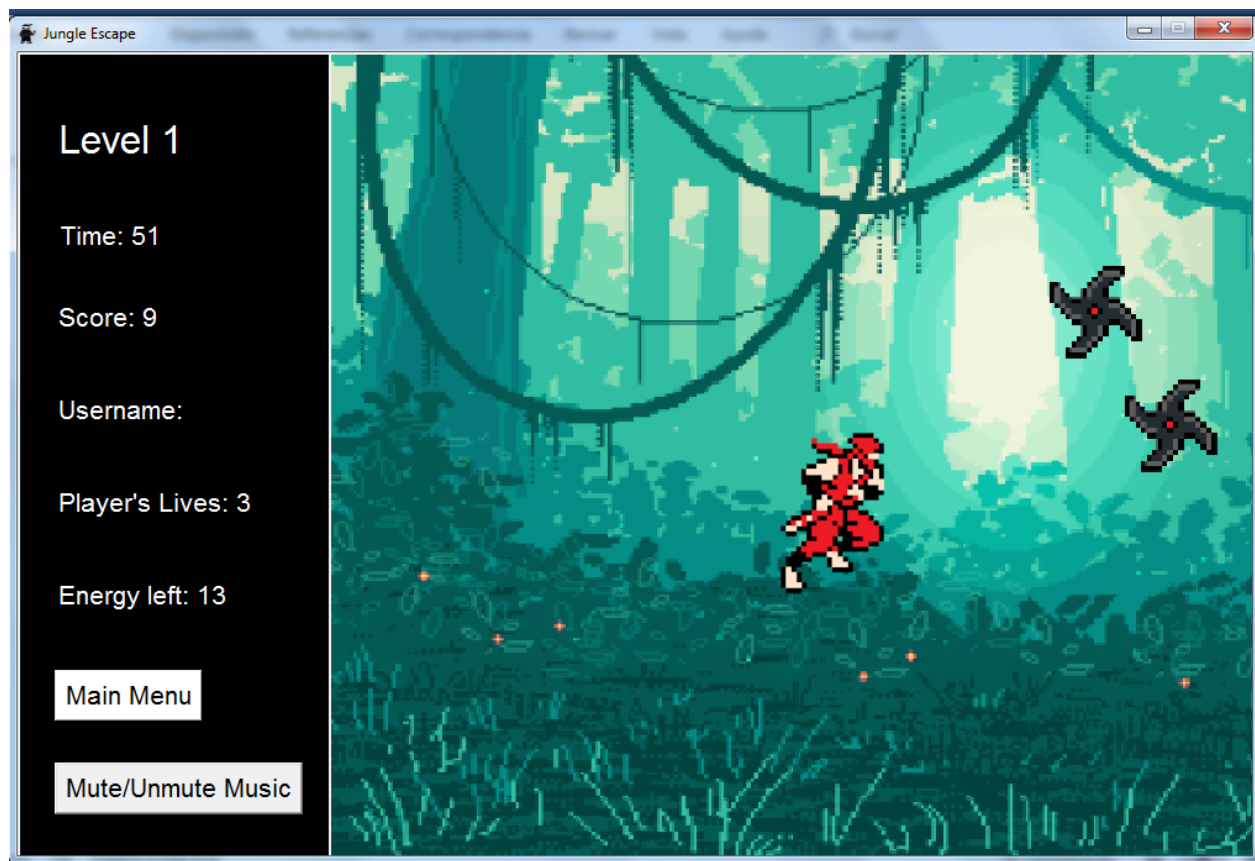
Menú principal:

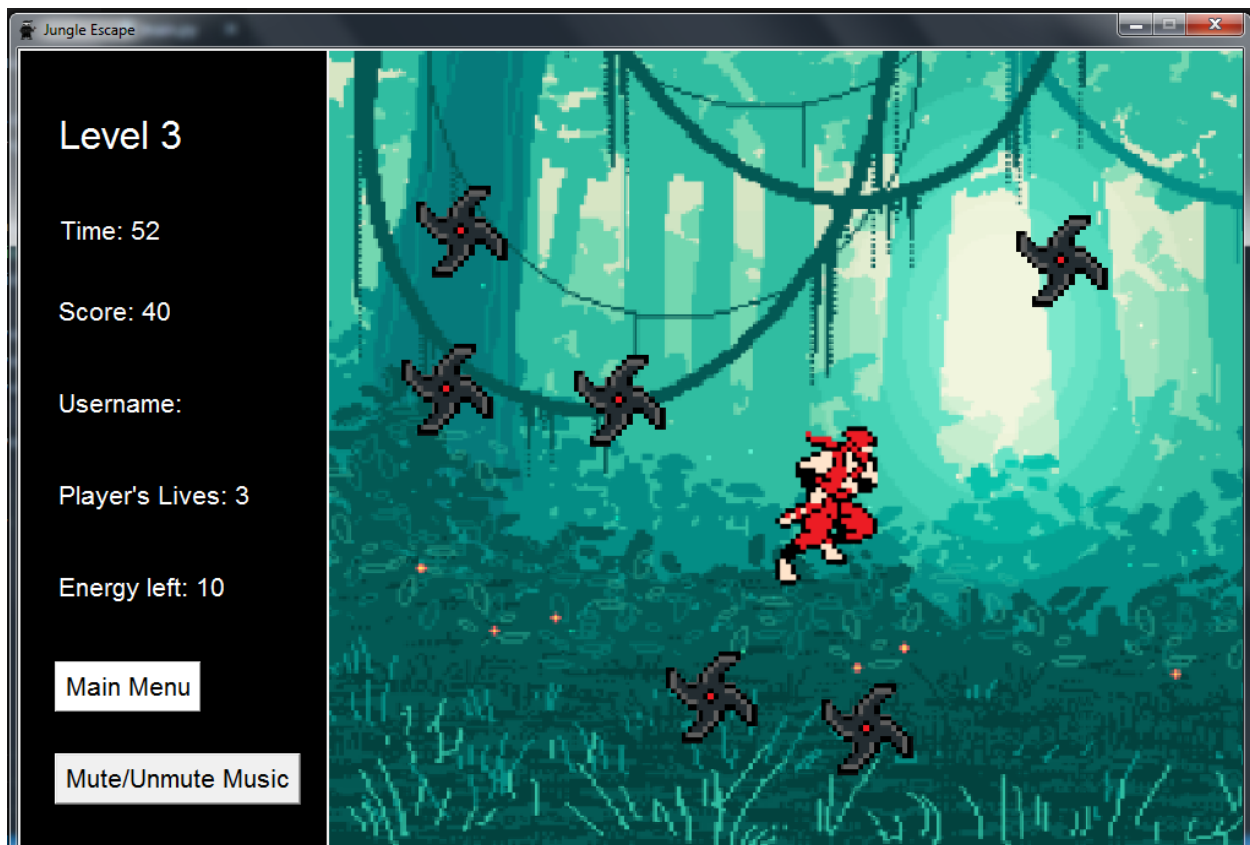
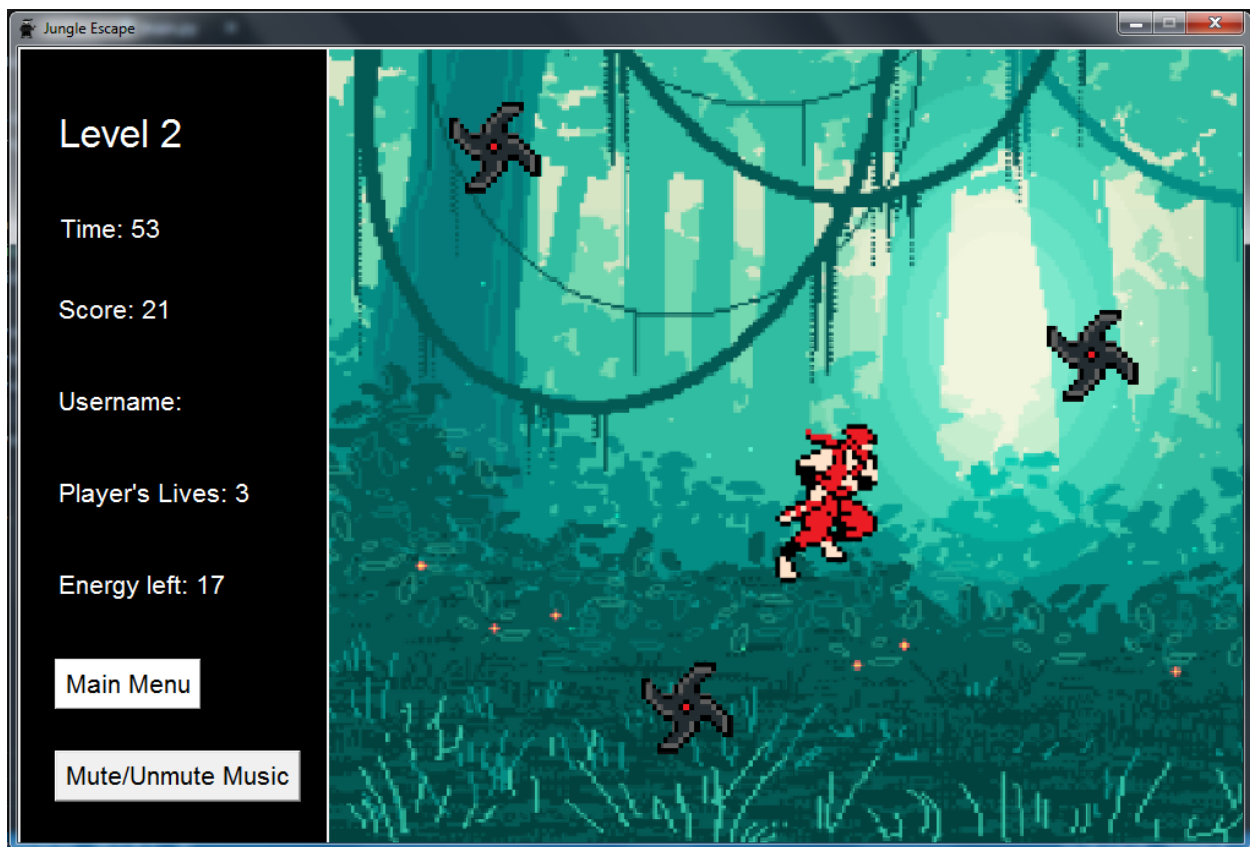


Pantalla de información complementaria (about):



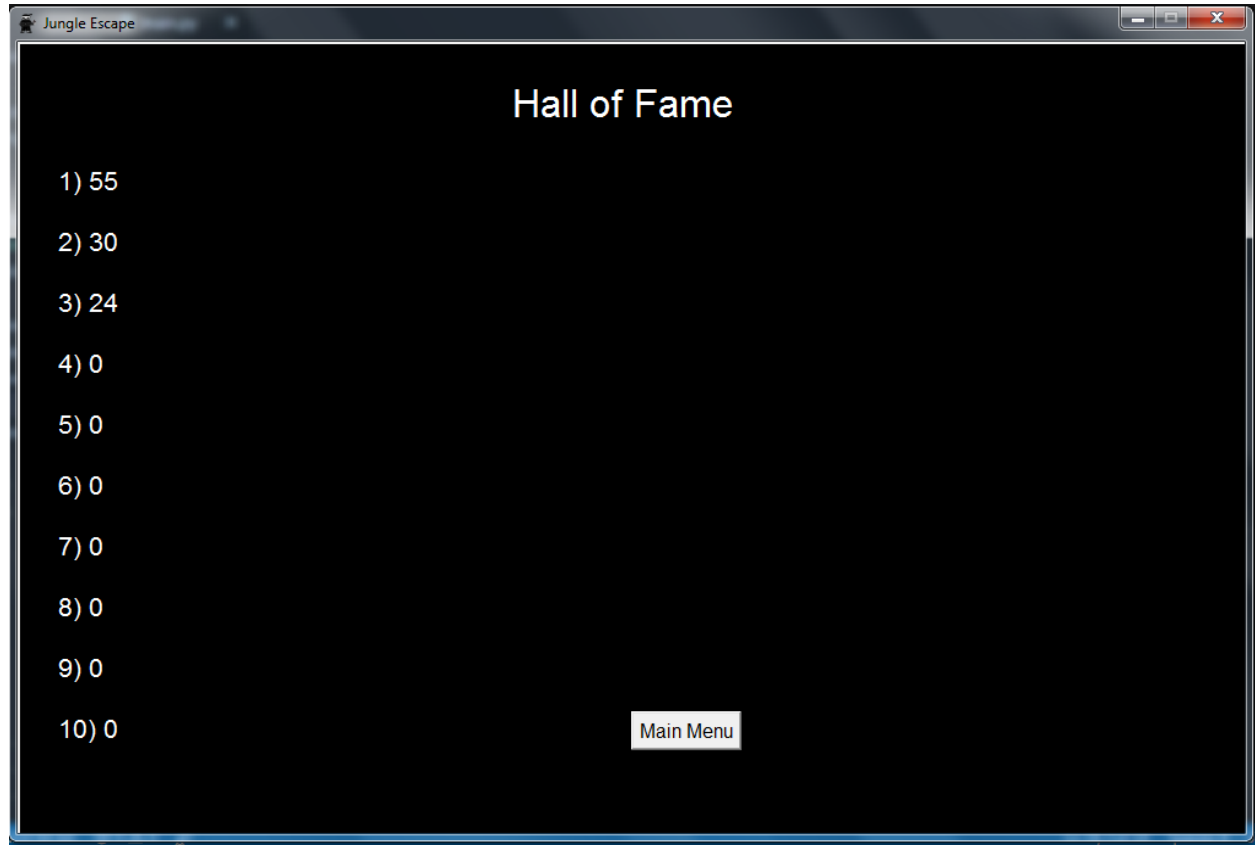
Niveles del juego:





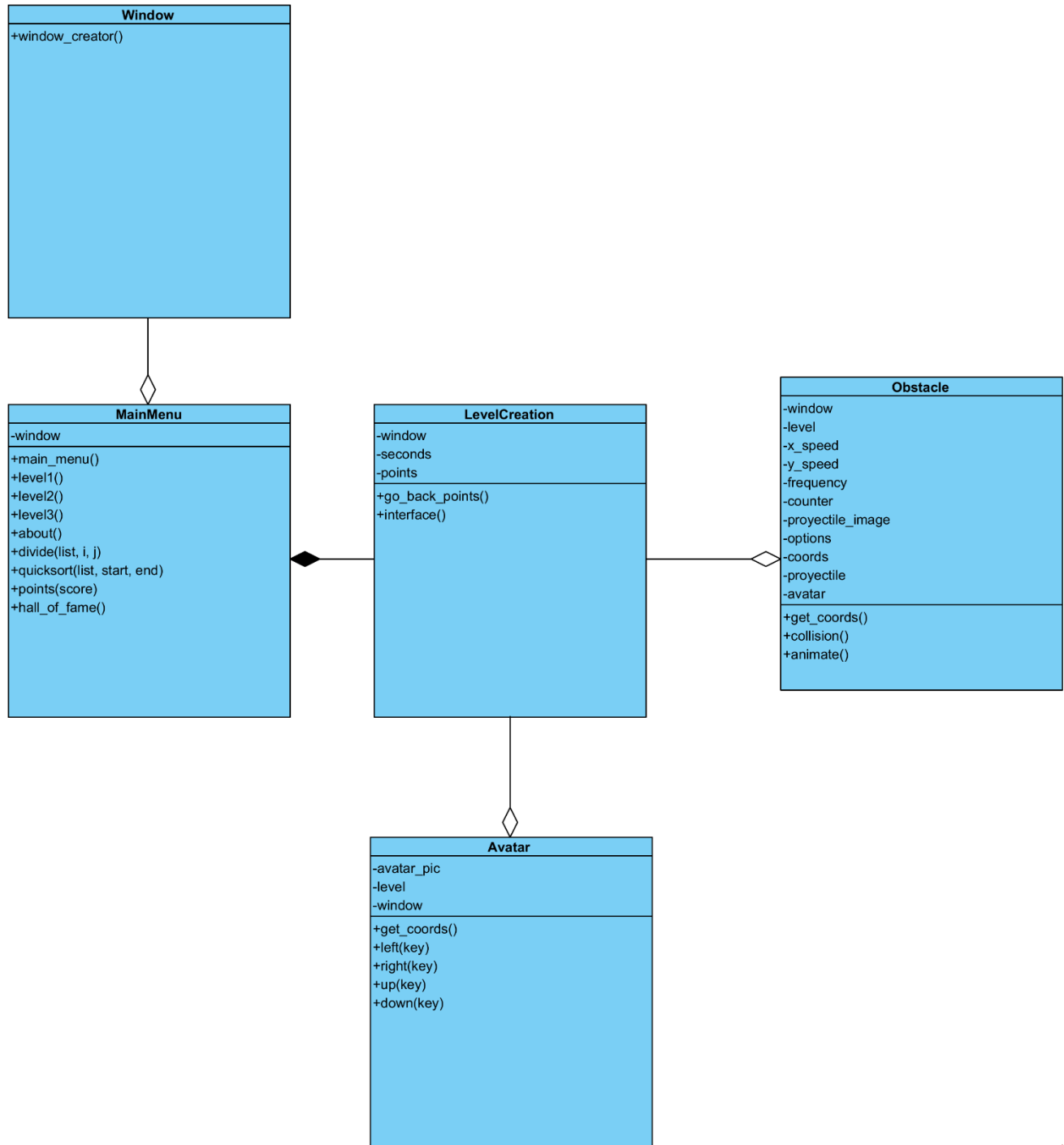


## Puntajes

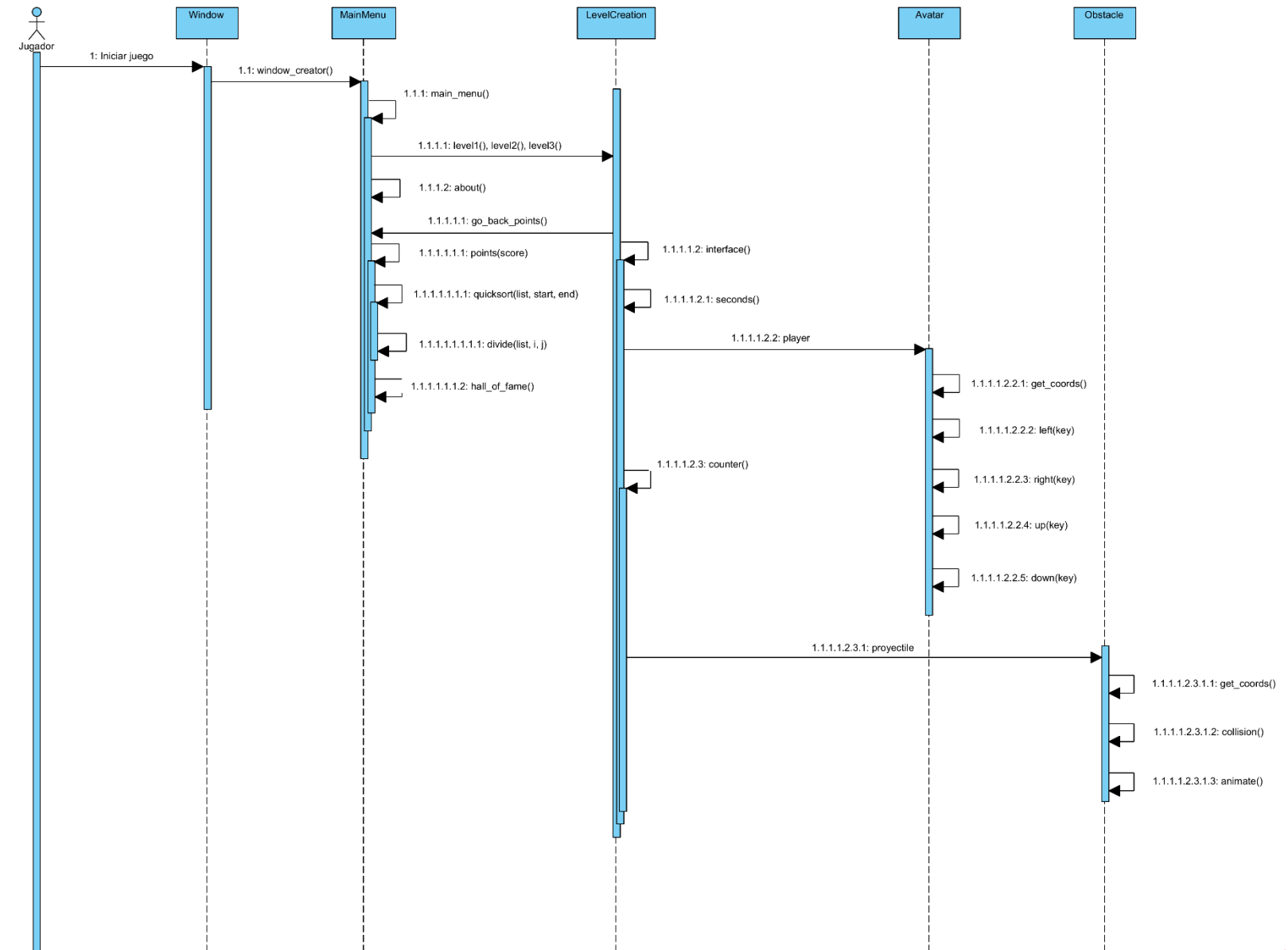


## Modelo de objetos:

- Diagrama de clases:



- Diagrama de secuencias:



## **Dificultades encontradas:**

### **1. Quicksort:**

Lograr que el algoritmo Quicksort funcionara correctamente tomó mucho tiempo, a pesar de que nos basamos en las instrucciones del algoritmo que venían detallados en imágenes, se logro primero hacer que se ordenaran los elementos de la lista menores y mayores del pivot, sin embargo no encontrábamos manera de organizar estos subgrupos, al final encontramos un código en internet que ejemplificaba el uso del Quicksort como dos funciones, una que generaba el orden con respecto al pivot, la cual ya habíamos realizado y otra que nos ayudó mucho que utilizamos como referencia que era iterativa y que iba realizando llamadas a la función de Quicksort para ordenar la lista en su totalidad.

### **2. Colisión de obstáculos y jugador:**

A pesar de que en el proyecto anterior ya se había trabajado con un sistema de colisiones, similar al de este nuevo proyecto, este nuevo proyecto presento sus dificultades, entre las cuales principalmente se debieron al modelo de objetos, ya que anteriormente todo se trabajo sin objetos, pero en este era parte del proyecto utilizar POO, y esto hizo que nos tuviéramos que adaptar a la forma de trabajar, principalmente a la manera de transmitir información entre objetos, y a poder utilizar esa información a tiempo real.

### 3. Imágenes de jugador y obstáculos no aparecían en la pantalla:

Originalmente definimos el canvas del primer nivel con su barra lateral de información, pero al agregar las imágenes del jugador y de los obstáculos, estos no aparecían, debido a que estos estaban cargando debajo de la imagen del fondo, es decir, se encontraban ahí, pero estaban ocultos. La solución fue definir canvas separados, uno para el nivel, otro para el fondo que contiene las imágenes en forma de etiquetas del fondo, el jugador y los obstáculos y otro para la barra de información lateral con sus etiquetas específicas.

### 4. Nivel seguía funcionando a pesar de estar en el nivel principal:

Tras establecer cada nivel, nos dimos cuenta de que, a pesar de cambiar de escena, los elementos de los niveles anteriores seguían funcionando, como por ejemplo aun los proyectiles colisionaban, y le causaban daño al jugador, así como producir sonidos, por lo cual tuvimos que buscar una forma de poder terminar cada nivel, y que las demás instrucciones de terminaran con el mismo. Finalmente se logró, pero fue una gran dificultad.

## **Bitácora de actividades:**

Andrés:

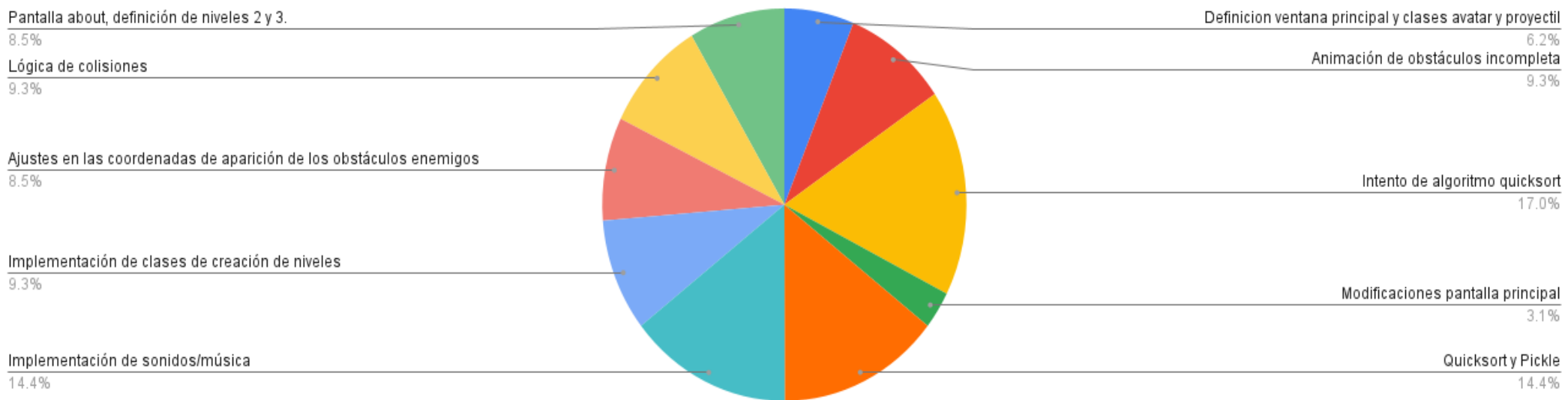
- 2 hrs - Definición ventana principal y clases avatar y proyectil
- 3 hrs - Animación de obstáculos incompleta
- 5 hrs 30 mins - Intento de algoritmo Quicksort
- 1 hr - Modificaciones pantalla principal, continuación Quicksort
- 4 hrs 39 mins - Implementación de Quicksort con lectura de archivo .txt con Pickle
- 4 hrs 40 mins - Implementación de sonidos/música Pygame, definición incompleta de objetos de avatar y obstáculos

Jose Pablo:

- 3hr - Acomodo general del código e implementación de clases de creación de niveles, así como cambio de niveles
- 2 hrs 45 mins - Ajustes en las coordenadas de aparición de los obstáculos enemigos
- 3 hrs - Desaparición de los obstáculos al colisionar, y colisiones ya programadas
- 2 hrs 45 mins - Pantalla about, definición de comportamiento de niveles 2 y 3.
- 5 horas – Implementacion de sonidos personalizados para cada nivel, así como los botones para cancelar la reproducción de sonidos
- 1 hora, 30 mins – Arreglo de detalles finales en el código, previo a la grabación del video

Total: 32 horas y 19 minutos

## Estadística de tiempos:



## **Conclusiones:**

- Andrés:

En conclusión, como en el proyecto pasado también trabajamos en un juego, me pareció más sencillo realizar este, a pesar de que igual consumió bastante tiempo, por lo menos no existía el problema de tener que aprender a hacer un juego desde cero, este proyecto sirvió para poner a prueba lo que aprendimos realizando el proyecto pasado y así mejorar algunas cosas en términos de organización al trabajar y optimización del código.

Utilizar objetos que fue la principal diferencia de este proyecto, fue algo tedioso al inicio para adaptarse, pero realmente logré apreciar al final la verdadera razón por la cual se han vuelto un estándar en la programación y es como ponerse a organizar archivos en carpetas, tal vez uno prefiere tenerlo todo suelto y ya conocer donde se encuentra a memoria, pero entre más archivos uno tenga se comienza a desorganizar todo y lo mismo pasa con los objetos, permite organizar todas las funciones de un programa y que tengan sus propios atributos sin tener que recurrir a variables globales y sin que funciones y variables que no tengan mucha correlación se mezclen entre sí. También aprendí que colaborar con alguien más en el ámbito de la programación es esencial, a pesar de que cada uno tenga sus propios estilos de programar, realmente alivia mucha carga el delegar tareas para cada uno.

- Jose Pablo:

De manera general este proyecto me pareció mas sencillo que el anterior, ya que en muchos aspectos estos eran muy similares, sin embargo, lo que mas me tomo tiempo, fue la implementación de POO, ya que, si lo habíamos utilizado, ya en esta escala, fue un reto completamente nuevo, y puso a prueba todo lo aprendido durante este semestre.



El otro gran reto al cual siento que nos enfrentamos, y es de lo que mas se aprendió durante este segundo proyecto, fue el trabajo en equipo en un mismo código, al ser dos personas las cuales programan de manera muy distinta, poder mantener un código ordenado, y poder llevar las distintas tareas del proyecto de manera correcta, fue un gran reto, del cual se aprendió bastante, y es de lo que yo mas le rescato a este segundo proyecto.

**Video:**

Link: <https://youtu.be/f54bl1347fE>

## **Evaluación experimental del tiempo de ejecución del algoritmo quicksort:**

Los resultados obtenidos de cada una de las 15 ejecuciones fueron respectivamente:

0.211 segundos

0.218 segundos

0.173 segundos

0.217 segundos

0.163 segundos

0.142 segundos

0.180 segundos

0.163 segundos

0.178 segundos

0.185 segundos

0.172 segundos

0.172 segundos

0.186 segundos

0.178 segundos

0.175 segundos

Por lo tanto, se obtuvo un promedio de 0.181 segundos de ejecución del algoritmo Quicksort en una lista de 10000 elementos.

## **Ensayo sobre ética, equidad y trabajo en equipo:**

Por más que sea tentador y algunas veces más sencillo optar por la opción de cometer plagio en un trabajo ya sea como un acto desesperado o simplemente una muestra de desinterés en lo que actualmente se está trabajando, sin importar la causa, el acto es moralmente incorrecto, ya que se está engañando a la persona que va a evaluar el trabajo, aprovechándose de un posible descuido a la hora del proceso de revisión, con la esperanza de que uno se puede salir con la suya, solamente esto si se comete plagio con el consentimiento de aquel que está siendo plagiado, si sucede sin que el autor original no se dé cuenta entonces se esta engañando al evaluador además de que también se le está robando a aquel que dedicó su esfuerzo en hacer el trabajo original.

Usualmente una consecuencia inmediata del acto del plagio sería que se dé cuenta el evaluador y proceda a imponer una sanción lo suficientemente grave tal que perjudique a la persona para que tenga consecuencias tales que les sirvan de lección, ya que de lo contrario entonces el costo de cometerlo no sería mucho y seguiría existiendo un incentivo para el mismo, sin embargo, algo que siempre aplica sin importar si el plagio es descubierto o no, es que la persona que lo comete se está engañando principalmente a sí misma ya que no esta aprendiendo nada del curso, por lo que si alguien pasara todo un grado académico cometiendo plagio, su título también seria “robado”, ya que no se lo ganó por sus propios méritos.

Para lograr un ambiente de trabajo equitativo en nuestro grupo al hacer este trabajo, fue muy importante la comunicación para así tener una idea de que quedaba pendiente por realizar, al inicio cada uno comenzó por trabajar en algo específico, pero en el transcurso si surgía un problema entonces veíamos como lo solucionábamos entre los dos, al igual que si alguno ya había trabajado por muchas horas, se distribuían las tareas a la otra persona con el objetivo de que en total trabajáramos una cantidad similar de horas, o

como cada uno tenía su ritmo, por lo menos de un peso similar, que al final así lo logramos.

## **Fuentes consultadas:**

Para saber cómo implementar el algoritmo Quicksort:

<https://stackabuse.com/quicksort-in-python>

Para saber cómo utilizar Pickle:

<https://www.codegrepper.com/code-examples/python/read+a+text+file+in+python+and+dump+the+contents>

Para saber cómo utilizar sonidos con Pygame:

<https://www.youtube.com/watch?v=pcdB2s2y4Qc&t>