Kevyn Guadamuz Rojas

Carnet: 2017021057

Roger Valderrama

Carnet: 2017113167

Taller de Programación

Tarea Programada #3

Grupo #1

Profesor: Jeff Schmidt

Fecha de Entrega:  
Domingo 18 de Junio

2017

**Índice:**

**Introducción…………………………………… Pag #3**

**Descripción del Problema…………………….. Pag #4**

**Diagrama de Clases…………………………… Pag #5**

**Dificultades Encontrada……………………… Pag #5**

**Análisis de Resultados………………………... Pag #7**

**Bitácora de Actividades………………………. Pag #8**

**Estadísticas de Tiempo……………………….. Pag #10**

**Conclusión…………………………………….. Pag #11**

**Introducción**

Para este proyecto se pretende desarrollar un simulador de una estación de trenes del Tecnológico de Costa Rica. El simulador pretende manejar los trenes con las diferentes rutas y simular el abordaje, la salida y la llegada de los trenes

Con este proyecto se pretende aumentar los conocimientos del lenguaje que se utiliza en este curso el cuál es Python.  
Se trabajará con conceptos de Programación Orientada a Objetos y también con el concepto de Lista Doblemente Enlazadas.

En la parte de la Programación Orientada a Objetos se utilizará para manejar cada objeto como lo son los trenes, los vagones y las maquinas.

El concepto de Listas Doblemente Enlazadas se puede observar en los vagones, con el cual cada vagón pertenece a un nodo y el tren en si es la lista completa.  
El simulador debe poseer un mecanismo automático que asigne los vagones a los trenes de acuerdo a la demanda de usuarios.

La simulación de la estación de tren se trabajará con una interfaz gráfica lo cual permitirá que se adquieran y se complementen los conocimientos sobre interfaz gráfica ya se la librería que se esté utilizando, en nuestro caso se utilizará Tkinter por lo que el conocimiento ya adquirido con proyectos anteriores se complementara con los conocimientos que se van a adquirir con este nuevo proyecto.

**Descripción del Problema.**

El principal objetivo con este proyecto es desarrollar un simulador de una estación de trenes que será construida en un futuro en el Instituto Tecnológico de Costa Rica. La simulación va a presentar el funcionamiento de llegadas y salidas de trenes, pudiendo administrarse la cantidad de pasajeros, la cantidad de vagones a asignar por cada tren y otros aspectos. No se van a considerar aspectos como capacidades de la vía férrea ni tampoco seguridad en el manejo y asignación de rutas. Tampoco se van a manejar filas de pasajeros (colas) ni pagos de para adquirir los tiquetes.

El simulador debe tener una serie de botones o menús que permitan realizar las siguientes funciones:

* Iniciar simulación: lee los datos de un archivo de configuración.
* Lista de rutas por hora: para la siguiente hora muestra todos los trenes que tiene planeada una salida o llegada.
* Estimación de demanda por ruta: por medio de esta opción se genera un número aleatorio de la cantidad de personas que van a viajar en una de las rutas a manejar en la hora. Aplica solo para salidas de trenes.
* Administración de vagones: el controlador puede realizar de forma manual o automática la asignación de vagones para una ruta, de acuerdo a la demanda.
* Los vagones libres son cargados desde el archivo de configuración.
* Salida de tren: desaparece de la lista de trenes a administrar.
* Llegada de tren: se registra la llegada y se liberan los vagones.

Todas estas funciones deben mostrar elementos gráficos que permitan al controlador lo que está pasando.

El tren deberá ser implementado en programación orientada a objetos. Se deberá definir los métodos y atributos de cada objeto. Pueden existir algunos métodos que se requieran y no estén descritos en la presente definición. Pueden realizarse modificaciones al modelo de objetos enunciado, siempre y cuando las mismas se documenten.

Para mejorar la calidad y presentación de la tarea, debe investigarse el uso de algunas funciones referentes a validaciones de datos y despliegue de información. Las funciones que podrían utilizarse, entre otras son:

* Utilización de multimedia: integración de animaciones, sonidos y otros.
* Generación de números aleatorios
* Manejo de archivos de texto

Diagrama de Clases:

**Dificultades Encontradas**

**Lectura y escritura de archivos de texto:**

El primer problema que apareció en el transcurso del desarrollo del proyecto es la lectura y la escritura de archivos de texto, los cuales deben utilizarse para inicializar las instancias de cada objeto.

Para la solución de este problema se implementa el uso de archivos “.cvs” los cuales son archivos que los datos se encuentran separados por comas.   
Para la lectura de estos archivos se importa la libre CSV que trae Python y se utiliza esta librería para el manejo de los archivos “.csv”, los cuales pueden ser modificados por medio del bloc de notas

**Definición de la función que administre los vagones automáticamente:**

Al implementar la lógica de esta función nos vimos en un problema ya que no encontrábamos la forma de ver como leer el estado del vagón (si se encontraba libre u ocupado) y ver la capacidad que este poseía. Y encontrar la manera que esta función cuando la demanda de pasajeros ya fuese cubierta.

Con la lectura de los archivos “.csv” se inicializa las instancias y desde ahí se empieza a trabajar con las instancias. De manera que existe una función que reciba cuales vagones se encuentran en estado libre y crea una lista con las instancias que poseen ese estado, luego se envía a la función que realiza el enganche de los vagones de manera automática y cuando algún vagón se engancha este cambia su estado a ocupado.

**Reproducir audio:**

Para la reproducción de audio necesitábamos implementar un Thread para que el audio se reprodujera mientras se ejecutaba la ventana y así al momento de apretar el mismo botón este se detuviera.

Debidos a los problemas que se han tenido con los Threads, el audio no pudo ser implementando con los Threads pero preguntado y nos sugirieron usar la librería PyGame para ejecutar el audio, en efecto se utiliza la librería dicha anteriormente y reproduce los audios sin ningún problema en cuanto al reproducirlos juntamente cuando se ejecuta la ventana.

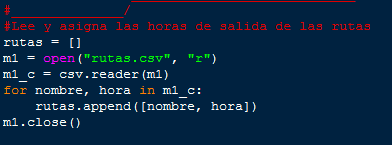
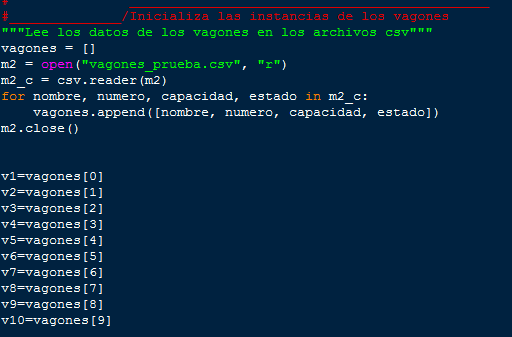
**Soldar el circuito en la Tarjeta Perforada:**

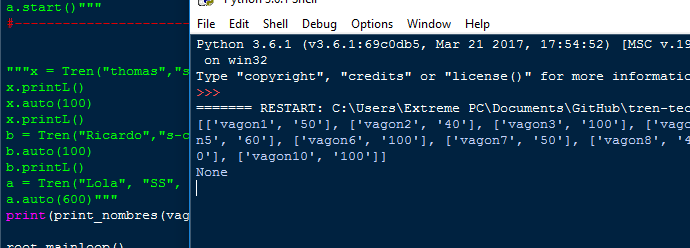
Debido a la inexperiencia con el trabajo con circuitos, la parte de soldar el circuito a la tarjeta perforada nos complicó un poco el desarrollo del proyecto, sin embargo, con la ayuda de la investigación se logra un resultado muy bueno con respecto a la soldadura.

**Análisis de Resultados**

**Lectura y escritura de archivos de texto:**

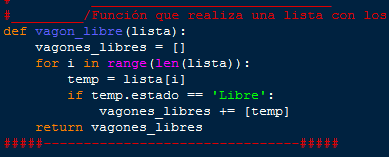
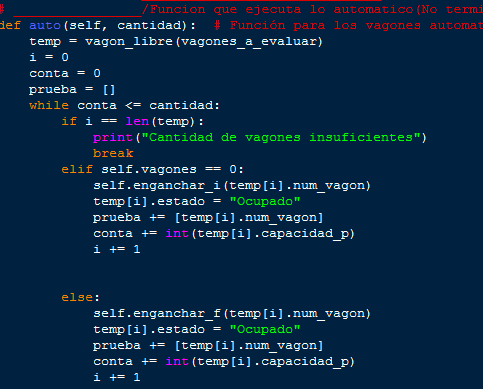
Se importa el módulo CSV que permite manejar los archivos de texto de una manera más sencilla y estos contienen los datos para que sean inicializadas las instancias.



**Definición de la función que administre los vagones automáticamente:**

Se implementa dos funciones para que sirva correctamente este método. Las cuales son “vagon\_libre”, lo que realiza esta función es que revisa el estado de los vagones y retorna una lista con todos los vagones que tienen el estado “Libre” y la otra función “auto” realiza el enganche de los vagones hasta que la demanda de los usuarios sea cubierta.

**Reproducir audio:**

Para la reproducción del audio se implementa la librería PyGame permitiendo así que ejecuten PyGame y Tkinter al mismo instante. Tambien en temas de utilización de recursos, PyGame permite ahorrar mucho espacio en el disco duro donde este es almacenado

**Soldar el circuito:**

Debido a la inexperiencia, como se comentó anteriormente, se complicó la parte de soldadura en la tarjeta. Pero con la ayuda de diferentes fuentes de información logramos soldar el circuito correctamente.

**Bitácora de Actividades**

Viernes 9 de Junio: Implementación de POO.  
Se inicia con la definición de cada uno de los métodos de las diferentes clases. Kevyn Guadamuz Rojas (3 horas)

Miércoles 14 de Junio:

Se empieza a trabajar con una interfaz de prueba.  
Se empieza a trabajar con una interfaz de prueba para verificar que ningún método afectara el funcionamiento de ambas partes.  
Kevyn Guadamuz Rojas (2 horas).

Implementación de Archivos CSV.  
Se investiga como leer y escribir de una manera más sencilla archivos de texto y se comienza a implementar una manera de leer los archivos.  
Kevyn Guadamuz Rojas (3 horas).

Se empieza a trabajar en la función automático.  
Se empieza a trabajar con la lógica de la función que realiza de manera automática el enganche de los vagones, pero no funciona de manera correcta ya que no se implementa la inicialización de cada instancia de la clase Vagón.  
Kevyn Guadamuz Rojas (4 horas).

Jueves 15 de Junio:

Se investiga otra forma de manejar archivos “.csv”  
Se implementa otra manera de manejar los archivos “.csv” de una forma más sencilla la cual es implementando el módulo CSV para el manejo de estos archivos.   
Kevyn Guadamuz Rojas (2 horas)

Viernes 16 de Junio:

Inicialización de Instancias de la clase Vagón.  
Se implementa como inicializar cada instancia de Vagón desde el archivo de texto.  
Kevyn Guadamuz Rojas (1 hora)

Función “vagon\_libre”.  
Se mejora la función vagón libre para que esta trabaje con las instancias que ya se crean desde el archivo “.csv”  
Kevyn Guadamuz Rojas (2 horas)

Función “auto”.  
Se trabaja en mejorar la función auto para que esta trabaje de la misma forma con las instancias con el estado libre provenientes de la función “vagon\_libre” y que este enganche de manera correcta y cambie el estado del vagón a “Ocupado”.  
Kevyn Guadamuz Rojas (3 horas)

Asignación de Variables en las rutas.  
Se implementa la manera de que la ruta sea leída desde el “.csv” para que pueda ser mostrada en la interfaz  
Kevyn Guadamuz Rojas (30 minutos)

Sábado 17 de Junio:

Documentación Externa.  
Se inicia con la documentación externa del proyecto.  
Kevyn Guadamuz Rojas (2 horas)

Instancias de la clase Tren y Maquina.

Se empieza a juntar la parte lógica con la interfaz

**Estadística de Tiempos**

|  |  |  |  |
| --- | --- | --- | --- |
| **FUNCION** | **Kevyn Guadamuz** | **Integrante 2** | **TOTAL** |
| Análisis de requerimientos | 13 horas | xx horas | xx horas |
| Diseño | 10 horas | xx horas | xx horas |
| Investigación de funciones | 5 horas | xx horas | xx horas |
| Programación | 20 horas | xx horas | xx horas |
| Circuito e interfaz | 21 horas | xx horas | xx horas |
| Documentación interna | 2 horas | xx horas | xx horas |
| Pruebas | 2 horas | xx horas | xx horas |
| Elaboración documento | 5 horas | xx horas | xx horas |
| **TOTAL** | **78 horas**  **\*\*Nota: Se hace demasiado debido a que algunas horas están repetidas e otros aspectos \*\*\*** | **xx horas** | **xx horas** |

**Conclusión**

Kevyn Guadamuz:

Con este proyecto aprendí mucho sobre las animaciones, conocí más Tkinter. Amplié mi conocimiento en el desarrollo de circuitos y la programación de un Arduino, antes de este proyecto, difícilmente sabía que era y gracias a este proyecto se logró trabajar con uno.

En el trascurso del proyecto aparecieron retos que se lograron superar de una manera correcta, siempre con la mentalidad de aprender para así lograr en un futuro desempeñarse de la mejor manera y aprovechar todo ese conocimiento que estamos obteniendo hoy y ponerlo en práctica en nuestros lugares de trabajo.

Se conoció más de la parte de electrónica combinada con la parte de la informática. El desarrollo de software y hardware son las bases de nuestra carrera y por proyectos como este son puestas en prácticas las dos facetas. La programación y la construcción de circuitos.