

Proyecto Programado

Tabla de contenido

Contenido

Tabla de contenido	1
Introduccion:.....	2
Descripción del Problema:	3
Dificultades Encontradas:.....	4
Análisis de resultados:	5
Bitácora de Actividades:	6
Estadística de tiempos:	8
Conclusión Personal:	9

Introduccion

El transporte por medio de trenes actualmente en nuestro país no es un transporte muy seguro, eficaz de buena calidad pues tienen pocos trenes y por tanto no cubren con la demanda por tanto los buses salen solamente en algunos en cierto horario mañanero para la salida y de regreso a otro horario y esto beneficia solo a cierta población; Los trenes son trenes viejos en no muy buenas condiciones, los sistemas de control de las estaciones son poco automatizadas y se comunican por llamadas por motivo que existen muchos casos de choques y/o vuelcos de vagones.

Los sistemas de control de los trenes desde sus inicios fueron variando poco. Algunos se han dejado de utilizar porque causaban accidentes y daban paso a sistemas más innovadores y actualizados; esto no sucede de forma automática en todos los países.

- **BLOQUEO TELEFONICO.**

Los jefes de circulación dan paso a los trenes por la vía telefónica y accionan las señales de modo totalmente manual. En Costa Rica hasta hace poco se utilizaba este sistema

- **SEÑALES AUTOMATICAS.**

Unas balizas interconectadas detectan los pasos de los trenes y activan las señales de paso o parada de modo automático. Francia fue uno de los primeros países en instalarlo y que más tiempo ha durando

- **SEÑALIZACIONES SONORAS.**

El tren tenia pegados a los lados unos pitos/petardos/etc. que sonaban al llegar a su destino, o cuando pasaba cerca de otro tren. Este sistema se siguió usando mucho hasta que se convirtió en los famosos pitos accionados por los maquinistas

- **SISTEMAS DE REDES.**

Son trenes conectados entre sí por una red que los acompaña por todo su recorrido, esto se registra en una central y por medio de procesos sincronizados asigna la salida y llegada de los trenes

Descripción del Problema:

El proyecto programado consiste en crear una simulación de estación de trenes en el TEC que se dirija a otras sedes del TEC en las 7 provincias a horas aproximadas. Además, haga la simulación de viaje y llegada a la otra sede.

Todo visto desde la óptica del operador que lo controla por lo tanto tiene que poderse vagones todo esto de forma manual. En la forma automática los.

El controlador puede hacer de forma manual:

- Controlar las salidas.
- Controlar las llegadas de los vagones
- La asignar de la máquina y los vagones al tren
- Le asigna un vagón en la posición inicial
- Le asigna un vagón en la posición final
- Le asigna un valor en la posición de en medio
- Pude quitar vagones

El sistema logra de forma automática:

- Las asignaciones de vagones son de forma aleatoria a su demanda
- La salida de los trenes de acuerdo a su hora de salida
- La llegada de los trenes a la sede prevista de acuerdo a su hora de llegada
- Todos estos datos son asignados previamente en el txt de configuración

Toda la estructura del proyecto debe realizarse con un lenguaje de programación llamado Python 3.6 y la interfaz gráfica debe realizarse con la librería tkinter, pygame, PIL y tkinter para el desarrollo de la GUI.

Dificultades Encontradas:

Durante el planteamiento del problema, fue la decisión de crear una animación estilo plana o una animación isométrica para la interfaz.

El segundo problema aun el planteamiento del programa es la comprensión de lo proyecto pues estaba un poco confuso.

El tercer problema fue encontrar la mejor forma de cargar imágenes transparentes y la instalación la librería PIL, pues esta tiene diversos requerimientos iniciales para su instalación y que no tenga conflicto con otras librerías existentes.

El tercer problema, el que más tiempo me llevo fue realizar la complementación de la GUI con la lógica de las listas doblemente enlazadas de forma eficiente y sin que se cayera ninguna función realizada durante la animación en automático y manual.

El cuarto problema fue la asignación de vagones al inicio y al medio, pues teníamos que combinar las posiciones de los archivos, decirles que se movieran de acuerdo a su tag.

El cuarto problema, fue la actualización del label que tenía que mostrar los datos cada vez que se generara una salida de un tren

El quinto problema fue la utilización de sonidos .wav el peso era excesivo por tanto no se realizó.

El sexto problema fue trabajar con los canvas pues estos se le podían asignar más de un tag a la vez

Análisis de resultados:

✚ Para la asignación de vagones al final (primera vez):

```
Exception in Tkinter callback
Traceback (most recent call last):
  File "C:\Users\Brenes\AppData\Local\Programs\Python\Python36-32\lib\tkinter\__init__.py", line 1699, in
__call__
    return self.func(*args)
  File "progra_final.py", line 410, in accion_enganchar_final
    if self.trenes[j].__len__() <=6:
NameError: name 'self' is not defined
```

✚ Para la asignación de vagones al final (ultima vez):

```
Tren[0]= [1,2,3,4]
Asignación al final
Tren[0] =[1,2,3,4,5]
```

✚ Pruebas con Label (primera vez):

En la imagen 2 Anexos

✚ Intentos de usar librería para las imagenes:

```
from PIL import Image
imagen = Image.open("prueba.jpg")
imagen.show()
RESTART: C:\Users\Brenes\Dropbox\TEC\intro progra\Ejemplo Tkinter pruebas\prueba_pillow.py
Traceback (most recent call last):
  File "C:\Users\Brenes\Dropbox\TEC\intro progra\Ejemplo Tkinter \pruebas\prueba_pillow.py", line 1, in
<module>
    from PIL import Image
  File "C:\Users\Brenes\AppData\Local\Programs\Python\Python36-32\lib\site-packages\PIL\Image.py", line 56,
in <module>
    from . import _imaging as core
ImportError: DLL load failed: No se encontrar el process especificado.

from apng import APNG
t1 = 200
t2 = 1000
atletas = [('atleta-01.png', t1),
           ('atleta-02.png', t1),
           ('atleta-03.png', t1),
           ('atleta-04.png', t1),
           ('atleta-05.png', t2)]

anima = APNG()
for archivo, delay in atletas:
    anima.append(archivo, delay=delay)
anima.save('animatleta2.png')
File "C:\Users\Brenes\AppData\Local\Programs\Python\Python36-32\lib\site-packages\apng\__init__.py", line
69, in chunks
    PIL.Image.open(png).save(f2, "PNG", optimize=True)
NameError: name 'PIL' is not defined
```

Bitácora de Actividades:

Fecha	Actividad	Responsable	Duración
4 de junio	Tutorials de pygame	Bertha	2 horas
5 de junio	Declaración de objetos con sus métodos y atributos Además de iniciar documentación externa	Bertha Brenes	5 horas
7 de junio	Iniciar los diseños de la estación	Bertha Brenes	8 horas
8 de junio	Iniciar los diseños de los vagones y los trenes	Bertha Brenes	5 o 6 horas
9 de abril	Empiezo a trabajar con Roger como pareja en la nueva interfaz	Bertha Brenes Roger Mora	4 horas
10 de abril	Trabajar un poco la lógica de la listas	Bertha Brenes	5 horas
10 de Abril	Diseño de la estación vieja	Roger Mora	6 horas
11 de abril	Terminar las listas Iniciar a trabajar en la Interfaz usando pygame solamente	Bertha Brenes	8 horas
12 de abril	Diseño de tren y vagón viejo	Roger Mora	6 horas
13 de Abril	Interfaz usando pygame solamente	Bertha Brenes	3 horas
14 de abril	Migración de interfaz de pygame a tkinter	Bertha Brenes	3 horas
15 de abril	Instalar y trabajar con PIL Desarrollo de interfaz usando PIL, tkinter y pygame	Bertha Brenes Roger Mora	12 horas
15 de Abril	Diseño de la estación actual	Roger Mora	8 horas
16 de abril	Desarrollo de documentación Reiniciar la interfaz e iniciar de nuevo pero con animaciones transparentes	Bertha Brenes Roger Mora	8 horas

17 de abril	Trabajo en la acoplar la interfaz con las listas doblemente enlazadas y quedando funcional	Roger Mora Bertha Brenes	18 horas
18 de abril	Encontrar los errores y solucionarlos Terminar detalles de interfaz e intentar quitar código basura Finalizar documentación	Bertha Brenes Roger Mora	15 horas

Estadística de tiempos:

Función	Tiempo
Análisis de requerimientos	30 horas
Diseño de aplicación	13-16 horas
Investigación de funciones	24 horas
Programación	50 horas
Documentación Interna	3 hora
Pruebas	16 horas
Elaboración de documento	2 horas
Total	138 horas

Imagen1:

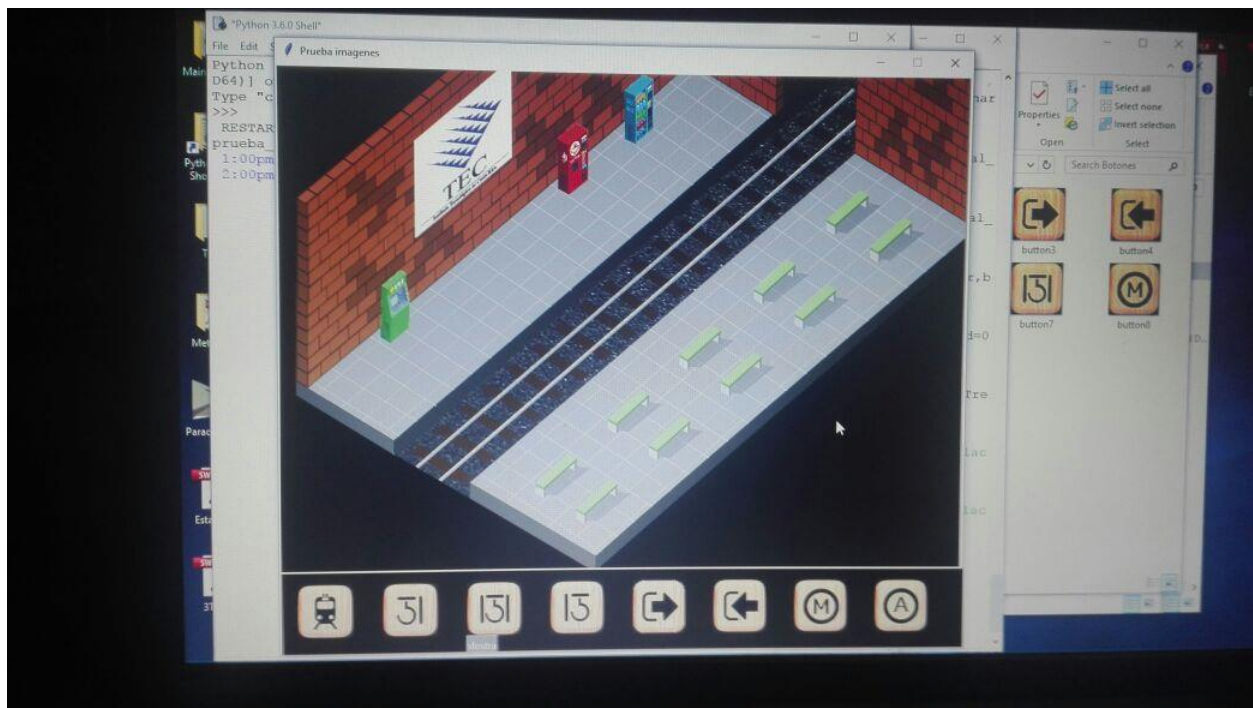
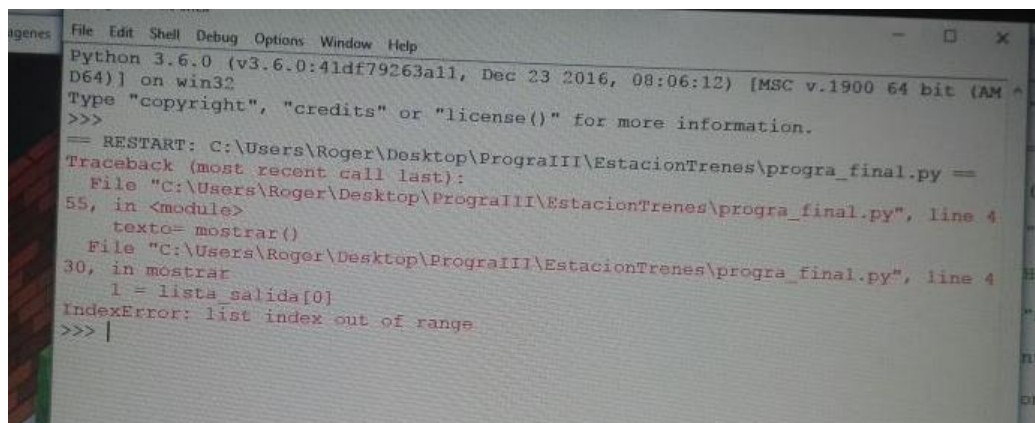


imagen 2



Conclusión Personal:

Bertha:

Me costó bastante comprender en si lo que quería el profe del proyecto porque la redacción del archivo estaba bastante confusa y un poco mal redactada. Pero aprendí bastante en la utilización de tkinter, PIL y pygame

Además de los grandes usos que se le pueden dar a los canvas con solo crear uno, la progra fue super interesante porque termine de entender el funcionamiento de las listas doblemente enlazadas de forma más visual. Me hubiera gustado agregarle más cosas, pero tiempo y espacio de memoria no se logró.

Roger:

En este proyecto aprendí bastante del manejo de clases, logré finalmente comprenderlas. Además, varias funcionalidades de Tkinter, Pygame e incluso el manejo de PIL. Fue un poco difícil trabajar utilizando Git, ya que aún no logro comprenderlo bien, tuve varios errores, pero aprendí un poco más a usarlo. Entendí lo importante que puede ser esta herramienta para trabajar en equipo, también sitios web como codeshare.io

Bibliografía

Ntwari, D. (2015). Resize images using Python. [online] Opensource.com. Available at: <https://opensource.com/life/15/2/resize-images-python> [Accessed 14 Jun. 2017].

Ferropedia.es. (2015). Sistemas de control y protección de trenes - Ferropedia. [online] Available at: http://www.ferropedia.es/wiki/Sistemas_de_control_y_protección_de_trenes [Accessed 12 Jun. 2017].

Shipman, J. (2013). 8.6. Methods on Canvas widgets. [online] Infohost.nmt.edu. Available at: <http://infohost.nmt.edu/tcc/help/pubs/tkinter/web/canvas-methods.html> [Accessed 14 Jun. 2017].

Campos, O. (2011). Multiprocesamiento en Python: Threads a fondo, introducción. [online] Genbetadev. Available at: <https://www.genbetadev.com/python/multiprocesamiento-en-python-threads-a-fondo-intro> [Accessed 14 Jun. 2017].

Shinners, P. (n.d.). Pygame Tutorials - Setting Display Modes — Pygame v1.9.2 documentation. [online] Pygame.org. Available at: <http://www.pygame.org/docs/tut/DisplayModes.html> [Accessed 13 Jun. 2017].

Game Development with Python 3. (2014). [video] https://www.youtube.com/watch?v=16LkXpZ4mKU&index=8&list=PLQVvva0QuDdLkP8MrOXLe_rKuf6r80KO.