



Universidad Autónoma de Chihuahua

Facultad de ingeniería

Carrera: Ingeniería en sistemas computacionales en hardware.

Materia: Simulación de sistemas.

Tarea: Cuarto avance simulación peluquería.

Alumno: Oscar Alejandro Serrano Pizarro

Matricula: 338820

Docente: Oscar Ramsés Ruiz Varela

Fecha: 11/05/2023

Explicación de lo agregado al programa.

1. Agregué una variedad de variables globales, que pueden ser alteradas para modificar un poco la simulación.

```
#Cosas que se pueden cambiar para alterar un poco la simulacion
FPS = 60 #ticks por segundo
VELOCIDAD = 5
SEGUNDOS_ENTRADA_MIN = 1
SEGUNDOS_ENTRADA_MAX = 3
SEGUNDOS_ATENDER_MIN = 3
SEGUNDOS_ATENDER_MAX = 9
MOSTRAR_CLIENTES_FALTANTES = True
REPETIR_AUTOMATICAMENTE = False
PRIMERO_RANDOM_TAMBIEN = False
GUARDAR_EN_SEGUNDOS = True #False guarda ticks de la simulacion
DECIMALES = 3 #Solo afecta si se mide en segundos
#Solo hay que asegurarse de que min es menor o igual a Max, sino ni va a correr.
```

Aquí podemos ver que podemos modificar fácilmente la velocidad de la simulación, de los clientes, los rangos de tiempos que duran las actividades en la peluquería, si se muestra o no una imagen que dice la cantidad de clientes faltantes, repeticiones automáticas de la simulación, si el primer cliente también es afectado por los rangos o si sale en el segundo 0, para guardar en el “.csv” utilizamos segundos o “ticks” y cuantas decimales tienen los segundos.

2. Añadí un diccionario en la clase “Cliente” para guardar los tiempos más tarde en el archivo “.csv”.

```
#El diccionario para guardar los tiempos de todos
self.tiempos = {'Tiempo de llegada': 0, 'Tiempo en formarse': 0, 'Tiempo formado': 0, 'Tiempo siendo atendido': 0, 'Tiempo de salida': 0, 'Tiempo pasado en la tienda': 0}
```

3. Alteré partes específicas del código para guardar los tiempos de cada cliente, y cuando el cliente llega a la salida de la peluquería lo guardo en el archivo “Tiempos_Peluqueria.csv” justo antes de eliminar todo lo relacionado con el cliente.

```
if GUARDAR_EN_SEGUNDOS:
    cliente.tiempos['Tiempo de llegada'] = round(self.tiempoTotal/FPS,DECIMALES)
else:
    cliente.tiempos['Tiempo de llegada'] = self.tiempoTotal
```

```

if not self.formado:
    self.tiempos['Tiempo en formarse'] = tiempo - self.tiempos['Tiempo de llegada']
    self.formado = True

self.sentado = True
self.tiempos['Tiempo formado'] = tiempo - (self.tiempos['Tiempo en formarse']+self.tiempos['Tiempo de llegada'])

cliente.atendido = True
if GUARDAR_EN_SEGUNDOS:
    tiempoTotal /= FPS
    tiempoTotal = round(tiempoTotal,DECIMALES)
    cliente.tiempos['Tiempo siendo atendido'] = tiempoTotal - (cliente.tiempos['Tiempo de llegada']+cliente.tiempos['Tiempo en formarse']+cliente.tiempos['Tiempo formado'])

if GUARDAR_EN_SEGUNDOS:
    tiempo /= FPS
    tiempo = round(tiempo,DECIMALES)
self.tiempos['Tiempo de salida'] = tiempo
self.tiempos['Tiempo pasado en la tienda'] = tiempo - self.tiempos['Tiempo de llegada']

#Cuando el cliente llega a la salida, lo borramos de la existencia, pero antes guardamos sus datos
try:
    with open('Tiempos_Peluqueria.csv', 'a', newline='') as archivocsv:
        escritor = csv.DictWriter(archivocsv, fieldnames=cliente.tiempos.keys())
        escritor.writerow(cliente.tiempos)
except IOError:
    print("La neta, no se pero ps no se abrio el .csv y ps ahora no se guardaron los datos del morro que acaba de salir")
self.clientes.remove(cliente)

```

4. La simulación y su archivo se ven de la siguiente manera.

