# Tables

December 21, 2021

@author: Marcos Tulio Fermin Lopez

[ ]:
```
import pygame
import Data_Manager
```

This module generates the tables and displays them in a Pygame window.

[ ]:
```
def show_AWT_Table():
    pygame.init()
    data = Data_Manager.get_data()

    # window properties
    WIDTH = 1400
    HEIGHT = 922
    WHITE_COLOR = (255, 255, 255)
    screen = pygame.display.set_mode((WIDTH, HEIGHT))
    # frame rate
    Clock = pygame.time.Clock()

    # convert table to desired size and remove bg
    tab1 = pygame.image.load('tables/table_1.PNG')
    white = (255, 255, 255)
    tab1.set_colorkey(white)
    tab1.convert_alpha()
    tab1 = pygame.transform.smoothscale(tab1, (600, 150))
    tab2 = pygame.image.load('tables/table_2.PNG')
    white = (255, 255, 255)
    tab2.set_colorkey(white)
    tab2.convert_alpha()
    tab2 = pygame.transform.smoothscale(tab2, (600, 150))
    title_font = pygame.font.SysFont('timesnewroman', 35)
    table_font = pygame.font.SysFont('timesnewroman', 25)
    table_font2 = pygame.font.SysFont('timesnewroman', 20)
    Title_surf1 = title_font.render("Average Waiting Time", True, (0, 0, 0))
    Title_surf2 = title_font.render("Cars Serviced", True, (0, 0, 0))
    Title_surf3 = title_font.render("Average Waiting Time (%)", True, (0, 0, 0))
    Title_surf4 = title_font.render("Cars Serviced (%)", True, (0, 0, 0))
```

```
    # texts
    time = table_font.render("Time (Sec)", True, (0, 0, 0))
    carsServed = table_font.render("Cars Served", True, (0, 0, 0))
    antenna = table_font.render("Antenna", True, (0, 0, 0))
    camera = table_font.render("Camera", True, (0, 0, 0))
    pir = table_font.render("PIR", True, (0, 0, 0))
    antennaT = table_font.render(f"{str(data['Antenna']['AWT'])}", True, (0, 0,␣
↪0))
    cameraT = table_font.render(f"{str(data['Camera']['AWT'])}", True, (0, 0,␣
↪0))
    pirT = table_font.render(f"{str(data['PIR']['AWT'])}", True, (0, 0, 0))
    antennaC = table_font.render(f"{str(data['Antenna']['carsServiced'])}",␣
↪True, (0, 0, 0))
    cameraC = table_font.render(f"{str(data['Camera']['carsServiced'])}", True,␣
↪(0, 0, 0))
    pirC = table_font.render(f"{str(data['PIR']['carsServiced'])}", True, (0,␣
↪0, 0))
    antVsCam = table_font2.render("Antenna Vs Camera", True, (0, 0, 0))
    antVsPir = table_font2.render("Antenna Vs PIR", True, (0, 0, 0))
    Efficiency = table_font.render("Efficiency", True, (0, 0, 0))

    # ****AWT Eff****

    # camera and pir
    awtAnt = (int(data['Antenna']['AWT']))
    awtCam = (int(data['Camera']['AWT']))
    awtPir = (int(data['PIR']['AWT']))
    efficiencyVsCamAWTstr = str(round(((abs(awtAnt - awtCam))/(awtCam) * 100),␣
↪2))
    efficiencyVsPirAWTstr = str(round(((abs(awtPir - awtAnt)/(awtPir)) * 100),␣
↪2))
    effCamAntAwt = table_font.render(efficiencyVsCamAWTstr, True, (0, 0, 0))
    effPirAntAwt = table_font.render(efficiencyVsPirAWTstr, True, (0, 0, 0))

    # ****Cars Serviced Eff****

    # camera and pir
    carsAnt = (int(data['Antenna']['carsServiced']))
    carsCam = (int(data['Camera']['carsServiced']))
    carsPir = (int(data['PIR']['carsServiced']))
    efficiencyVsCamCARSstr = str(round(((abs(carsAnt - carsCam)/(carsCam)) *␣
↪100), 2))
    efficiencyVsPirCARSstr = str(round(((abs(carsAnt - carsPir)/(carsPir)) *␣
↪100), 2))
    effCamAntCars = table_font.render(efficiencyVsCamCARSstr, True, (0, 0, 0))
    effPirAntCars = table_font.render(efficiencyVsPirCARSstr, True, (0, 0, 0))
```

```
    run = True
    # game starts
    while run:
        # Display screen
        screen.fill((WHITE_COLOR))
        # Display table
        screen.blit(tab1, (70, 200))
        screen.blit(tab1, (730, 200))
        screen.blit(Title_surf1, (200, 150))
        screen.blit(Title_surf2, (930, 150))
        screen.blit(tab2, (70, 650))
        screen.blit(tab2, (730, 650))
        screen.blit(Title_surf3, (200, 600))
        screen.blit(Title_surf4, (930, 600))
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                run = False
            if event.type == pygame.KEYDOWN:
                if event.key == pygame.K_ESCAPE:
                    run = False

        # print(pygame.mouse.get_pos())
        '''*******Display table texts*****'''

        # ------------AWT----------------
        screen.blit(antenna, (252, 230))
        screen.blit(camera, (405, 230))
        screen.blit(pir, (570, 230))

        # -------Cars Serviced------------
        screen.blit(antenna, (910, 230))
        screen.blit(camera, (1067, 230))
        screen.blit(pir, (1230, 230))
        '''************Data*************'''

        # ----------AWT------------------
        screen.blit(antennaT, (270, 294))
        screen.blit(cameraT, (410, 294))
        screen.blit(pirT, (570, 294))
        screen.blit(time, (93, 294))

        # -------Cars Serviced-------------
        screen.blit(antennaC, (950, 294))
        screen.blit(cameraC, (1090, 294))
        screen.blit(pirC, (1240, 294))
        screen.blit(carsServed, (750, 294))
```

```
        # -------Efficiency-----------------
        screen.blit(antVsCam, (284, 687))
        screen.blit(antVsPir, (495, 687))
        screen.blit(effCamAntAwt, (346, 748))
        screen.blit(effPirAntAwt, (555, 748))
        screen.blit(Efficiency, (120, 748))
        screen.blit(antVsCam, (950, 687))
        screen.blit(antVsPir, (1155, 687))
        screen.blit(effCamAntCars, (1000, 748))
        screen.blit(effPirAntCars, (1210, 748))
        screen.blit(Efficiency, (782, 748))
        pygame.display.flip()
        Clock.tick(10)
        pygame.display.set_caption(
            "Marcos Fermin's Dynamic Traffic Lights Simulator - EE Capstone␣
↪Project - Fall 2021")
```

[ ]:
```
if __name__ == '__main__':
    show_AWT_Table()
    show_AWT_Table()
```