

รายละเอียดของชิ้นงาน

ข้อมูลที่นำมาใช้งาน

ข้อมูลทั้งหมด 50 รายการ

แสดงข้อมูลลำดับที่ 1 ถึง 100

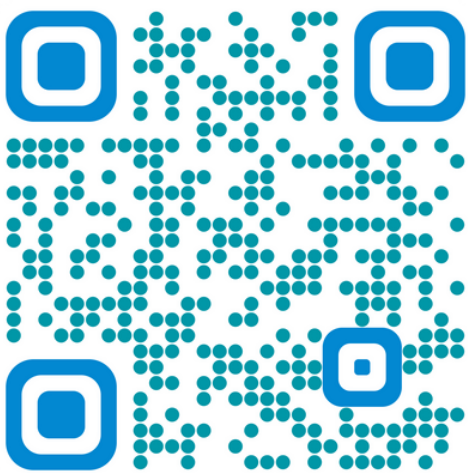
	_id	dcode	birthyear52	birthyear53	birthyear54	birthyear55	b
1	1	1001	1	2	0	1	1
2	2	1002	4163	3416	3209	3074	273
3	3	1003	506	481	484	403	327
4	4	1004	4471	4429	4754	5104	521
5	5	1005	1130	1062	868	1016	962
6	6	1006	1228	1144	1147	1124	101
7	7	1007	8732	7133	7525	7694	715
8	8	1008	2665	2675	2826	2288	206
9	9	1009	6	4	2	4	2
10	10	1010	4090	4830	5375	5082	433
11	11	1011	401	384	352	444	409
12	12	1012	4	5	4	6	2
13	13	1013	0	0	0	1	0
14	14	1014	1898	1827	1919	2026	202
15	15	1015	2994	2618	2650	2325	2113
16	16	1016	1	0	0	1	0
17	17	1017	2076	1964	1889	2097	182
18	18	1018	4292	3570	2936	2751	260
19	19	1019	3	3	2	6	1
20	20	1020	9691	9779	9735	10550	102
21	21	1021	1562	1597	1887	1682	152

ข้อมูลทรัพยากร : ข้อมูลจำนวนการเกิด – การตายของประชากรในเขตกรุงเทพมหานคร

ข้อมูลจำนวนกการเกิดและการตายของประชากรในเขตกรุงเทพมหานครโดย
ดาวน์โหลดมาในรูปแบบไฟล์CSV เป็นข้อมูลในช่วงปี 52 - 61 ประกอบด้วยการเกิดและ
การตาย ในแต่ละปีของทั้งการเกิดและตายจะมีข้อมูลย่อยลงไปของแต่ละเขตในกทม. มี
ทั้งหมด50เขต

ข้อมูลนี้มาจากเว็บไซต์

<https://data.go.th/dataset/birthdead1>



รูปแบบการวิเคราะห์

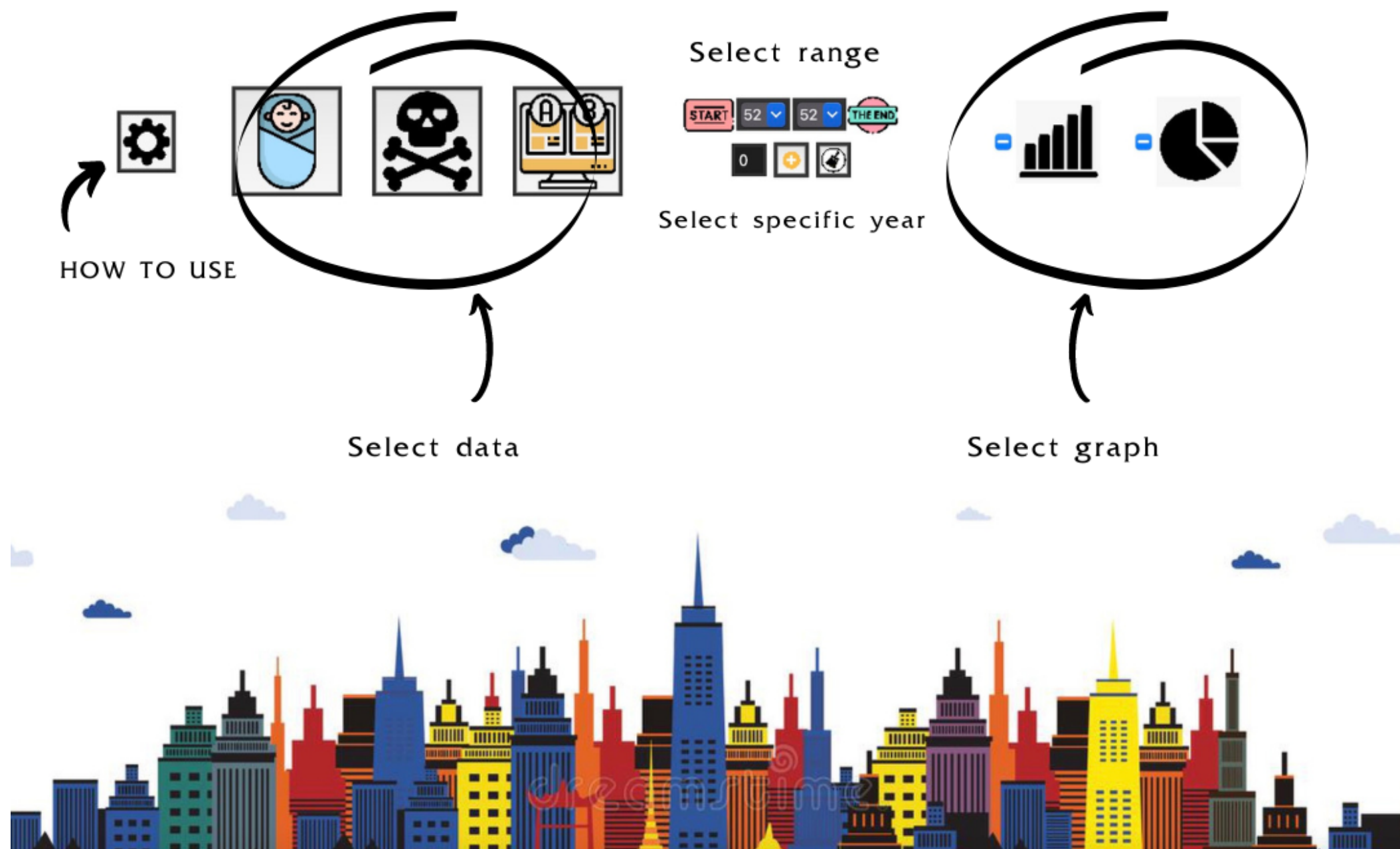
Library ที่จำเป็น

- 1.tkinter สำหรับการสร้าง GUI
- 2.matplotlib สำหรับการสร้างกราฟ
- 3.pandas สำหรับการนำเข้าไฟล์CSV
- 4.numpy สำหรับฟังก์ชัน np.array() เพื่อระบุปีของกราฟ

ขั้นตอนการทำ

- 1.นำเข้าไฟล์
 - 2.นำเข้ารูปภาพICONสำหรับหน้าGUI
 - 3.สร้างหน้าต่างโดย tkinter และตั้งค่านำจอ
 - 4.สร้างLabel ,Button , Combobox , Text box and Checklist
 - 5.ตั้งค่าWidgetทั้งหมดข้างต้น
 - 6.จัดวางWidgetตามที่ออกแบบไว้
- ,

หน้าUIและการใช้งานคร่าวๆ



การใส่ลำดับการใช้งาน

1.เลือกปีที่สนใจ

เลือกแบบช่วงปี

เช่น ปี52-54 , 53-61 เป็นต้น โดยกดตรงcombobox
กล่องซ้ายคือเริ่มที่ปีไหน กล่องขวาคือจบที่ปีไหน

เลือกเฉพาะปีที่สนใจ

เช่น ปี54 , 55 , 61 เป็นต้น โดยพิมพ์ปีที่สนใจทีละปีลงในกล่องText box
เมื่อพิมพ์ปีเสร็จให้กดเครื่องหมายบวกเพื่อเพิ่มเข้าลิสต์ปีที่สนใจ
และกดเคลียเพื่อล้างข้อมูลในลิสต์

การเลือกแบบช่วงปีจะทำงานต่อเมื่อลิสต์ปีที่สนใจไม่มีค่าภายใน

2.เลือกรูปแบบกราฟที่สนใจ

หากสนใจให้แสดงกราฟแบบใดให้กดไปที่รูปภาพกราฟนั้น
หากสนใจให้แสดงทั้งสองแบบให้กดเลือกทั้งสองอัน

3.เลือกรูปแบบข้อมูลที่สนใจ

หากสนใจให้แสดงข้อมูลแบบใดให้กดไปที่รูปภาพนั้น
โดยเครื่องหมายตารางเป็นการเปรียบเทียบระหว่างข้อมูลการเกิดและการ
ตายโดยข้อมูลสามารถแสดงได้แค่เพียงรูปแบบกราฟแท่ง

โครงสร้างโปรแกรมและการแบ่งฟังก์ชัน

นำเข้า Library ที่จำเป็น

```
GUI.py > ...
1  #import library
2  import pandas as pd
3  import matplotlib.pyplot as plt
4  from tkinter import *
5  from tkinter import ttk
6  import numpy as np
7  from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
8  import matplotlib.colors as mcolors
9
```

นำเข้าไฟล์CSV สร้างหน้าต่างUI นำเข้ารูปภาพที่ใช้งาน

```
9
10 #import CSV file
11 df = pd.read_csv("./birthdead.csv")
12
13 #making GUI window and window setting
14 root = Tk()
15 root.geometry("1200x1000")
16 root['bg'] = 'white'
17 root.title("Birth and Death in BKK")
18
19 #import png file
20 startpic = (PhotoImage(file = r"./start_icon.png")).subsample(12,12)
21 endpic = (PhotoImage(file = r"./end_icon.png")).subsample(12,12)
22 arrowpic = (PhotoImage(file = r"./arrow.png")).subsample(12,12)
23 bkk = (PhotoImage(file = r"./bkkpic.png")).subsample(1,1)
24 addpic = (PhotoImage(file = r"./add.png")).subsample(24,24)
25 clearpic = (PhotoImage(file = r"./clear.png")).subsample(95,95)
26 setpic = (PhotoImage(file = r"./setting.png")).subsample(20,20)
27 howto = (PhotoImage(file = r"./howto.png")).subsample(1,1)
28 barpic = (PhotoImage(file = r"./bar.png")).subsample(12,12)
29 piepic = (PhotoImage(file = r"./pie.png")).subsample(12,12)
30 bpic = (PhotoImage(file = r"./birth_icon.png")).subsample(6,6)
31 dpic = (PhotoImage(file = r"./death_icon.png")).subsample(6,6)
32 bdpic = (PhotoImage(file = r"./compare.png")).subsample(6,6)
33
```


ฟังก์ชัน setting() สำหรับป้อนอัปเดตภาพการใช้งานคราวๆ

```
# HOW TO USE image section
def setting():
    how = Toplevel(root)
    how.geometry("1200x1000")
    how.title("How to use")
    how['bg'] = 'white'
    Label(how,image=howto,bg="black" ).place(relx=0.5, rely=0.5, anchor=CENTER)

how_to = Button(root,image=setpic,bg = "white",command=setting).place(relx = 0.1,rely=0.2,anchor=CENTER)
```

สร้างcomboboxสำหรับการเลือกช่วงปี

```
#making combobox for select first year
Label(root,image=startpic,bg="white" ).place(relx=0.5, rely=0.17, anchor=CENTER)
start_value = IntVar()
start_value.set(52)
startt = ttk.Combobox(root,textvariable = start_value,width=2)
startt["value"] = [i for i in range(52,62)]
startt.place(relx=0.52, rely=0.17, anchor=W)

#making combobox for select end year
Label(root,image=endpic,bg="white").place(relx=0.617, rely=0.17, anchor=CENTER)
end_value = IntVar()
end_value.set(52)
endd = ttk.Combobox(textvariable = end_value,width=2)
endd["value"] = [x for x in range(52,62)]
endd.place(relx=0.56, rely=0.17, anchor=W)
```

สร้างentryรับค่าปีที่สนใจ

add_graph() สำหรับยืนยันปีและนำค่าเข้าลิสต์ graph_list

clear_graph() สำหรับเคลียร์ลิสต์

show_list() แสดงปีที่เพิ่มทั้งหมดในลิสต์

```
#for select specific year
data = IntVar()
en1 = Entry(width=2,textvariable=data,font=30)
en1.place(relx=0.517, rely=0.22, anchor=W)

graph_list = []
def add_graph():
    year_add = data.get()
    if year_add in range(52,62) and year_add not in graph_list:
        graph_list.append(year_add)
    en1.delete(0,END)
    show_list()

def clear_graph():
    print(graph_list)
    graph_list.clear()
    en1.delete(0,END)
    show_list()

def show_list():
    Label(root,text=graph_list,bg="pink",font= 30).place(relx=0.56,rely=0.27,anchor=CENTER)

btn = Button(root,image=addpic,command=add_graph).place(relx=0.547, rely=0.22, anchor=W)
btn1 = Button(root,image=clearpic,command=clear_graph).place(relx=0.577, rely=0.22, anchor=W)
```

สร้างcheckboxbuttonเพื่อเลือกการแสดงรูปแบบกราฟ

```
#graph selection
choice_bar = StringVar()
Checkbutton(root,image=barpic,variable=choice_bar,bg="white").place(relx=0.7, rely=0.2, anchor=W)
choice_pie = StringVar()
Checkbutton(root,image=piepic,variable=choice_pie,bg="white").place(relx=0.8, rely=0.2, anchor=W)
```

สร้างปุ่มเลือกกรรูปแบบข้อมูลและแสดงกราฟที่เลือกไว้

```
birth_button = Button(root,image=bpic,command=birth_selected).place(relx = 0.2,rely=0.2,anchor=CENTER)
death_button = Button(root,image=dpic,command=death_selected).place(relx = 0.3,rely=0.2,anchor=CENTER)
bd_button = Button(root,image=bdpic,command=compare_selected).place(relx = 0.4,rely=0.2,anchor=CENTER)
```

birth_button() แสดงข้อมูลการเกิดในกรุงเทพมหานครโดยแสดงออกตามรูปแบบกราฟที่เลือกไว้

death_button() แสดงข้อมูลการเกิดในกรุงเทพมหานครโดยแสดงออกตามรูปแบบกราฟที่เลือกไว้

bd_button() แสดงข้อมูลการเปรียบเทียบระหว่างการเกิดและการตายซึ่งมีพื้นรูปแบบกราฟต่าง