แปลภาพฉัน ด้วยเสียงเธอ

I told this story for you

01

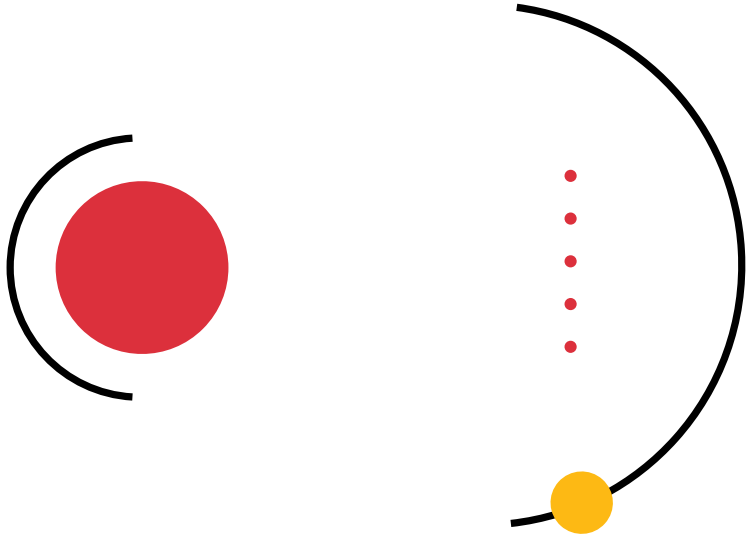
02

03

04

วัตถุประสงค์

เพื่อสร้างหนังสือภาพนิทานเสียง
ในรูปแบบของหนังสืออิเล็กทรอนิกส์
ด้วยหลักการของการออกแบบเพื่อคนทั้งมวล



01

02

03

04

ขอบเขตการศึกษา

01

ชุดข้อมูลนำเข้า (input) ต้องเป็นหนังสือภาพนิทานและข้อความภาษาอังกฤษที่อยู่ในรูปแบบของไฟล์ pdf เท่านั้น

02

ดำเนินงานโดยใช้งานไลบรารี่จากโปรแกรมภาษาไพธรอน (Python)

03

ออกแบบและสร้างหนังสืออิเล็กทรอนิกส์ประเภทสื่อประสมที่เสนอระหว่างสื่อภาพนิ่งกับสื่อประเภทเสียงภาษาอังกฤษเท่านั้น

04

ประโยชน์ที่คาดว่าจะได้รับ

หนังสือภาพนิทานเสียงแบบอิเล็กทรอนิกส์สามารถใช้เป็นสื่อการเรียนรู้สำหรับทุกคนที่สนใจได้

หนังสือภาพนิทานเสียงแบบอิเล็กทรอนิกส์ช่วยส่งเสริมทักษะในด้านการอ่าน การฟังและการสื่อสารโดยใช้ภาษาอังกฤษได้

01

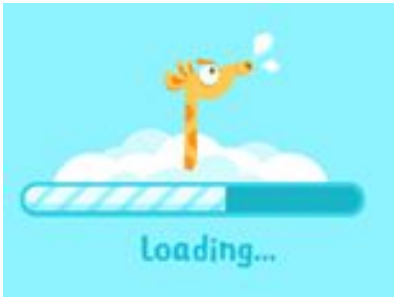
02

03

04

The input–process–output (IPO) model

input



process

output

01

02

03

04

Operation

01 PDF TO IMAGES

02 EXTRACT TEXT FROM IMAGES

03 TEXT TO SPEECH

04 GUI

01

PDF TO IMAGES

- pdf2image
- PIL(pillow)

```python
def pdf_to_image(bookfile,foldername):
    folder = foldername
    os.makedirs(folder, exist_ok=True)
    images = convert_from_path(bookfile)
    for i, image in enumerate(images):
        fname = 'image'+str(i)+'.png'
        completeName = os.path.join(folder, fname)
        image.save(completeName, "PNG")
    return folder
```

02

Extract text in image

- pytesseract
- PIL(pillow)

```python
def get_text_from_im(folder):
    path_to_tesseract = r"C:\Program Files (x86)\Tesseract-OCR\tesseract.exe"
    condition = True
    i = 0
    while condition:
        from os import path
        if path.exists(f"{folder}\image{i}.png"):
            try:
                image_path = f"{folder}\image{i}.png"
                img = Image.open(image_path)
                pytesseract.tesseract_cmd = path_to_tesseract
                text = pytesseract.image_to_string(img)
                text_to_speech(text,i, folder)
            except:

                folder_name = f'{folder}\sound'.format(folder)
                fname = f"sound{i}.mp3"
                completeName = os.path.join(folder_name, fname)
                path = r"my/path/to/file.txt"
                assert os.path.isfile(completeName)
                with open(completeName, 'w') as fp:
                    pass
                print(f"save {folder}\sound\sound{i}.mp3".format(folder))
        else:
            condition = False
        i+=1
    return text
```

Dylan the young, blue
dragon lived deep in the Rockstone
Mountains of Scotland. Everyday the
playful, young dragon woke up early
in the morning and flew all around
the mountains and valleys looking for
something to do.

01

02

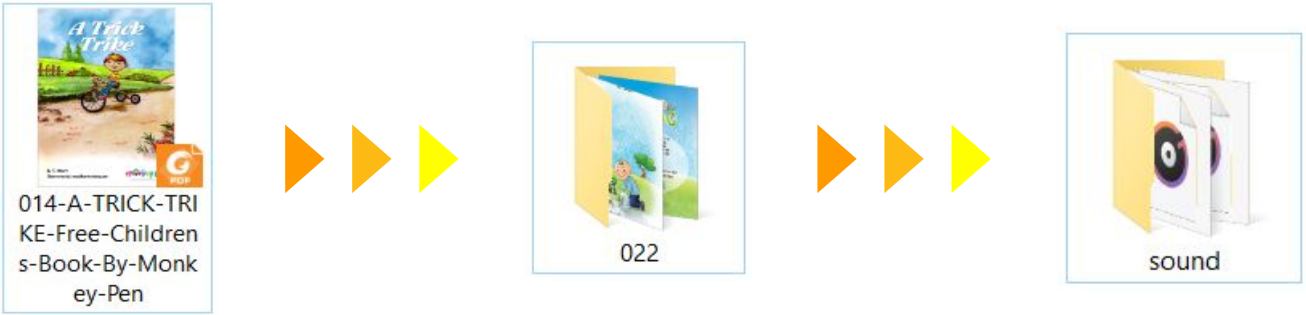
03

04

TEXT TO SPEECH

- gTTS (google text-to-speech)

```
def text_to_speech(text, i, folder):
    folder_name = f'{folder}\sound'.format(folder)
    os.makedirs(folder_name, exist_ok=True)
    language = 'en'
    myobj = gTTS(text=text, lang=language, slow=False)
    fname = f"sound{i}.mp3"
    completeName = os.path.join(folder_name, fname)
    myobj.save(completeName)
    print(f"save {folder}\sound\sound{i}.mp3".format(folder))
    return
```



01

02

03

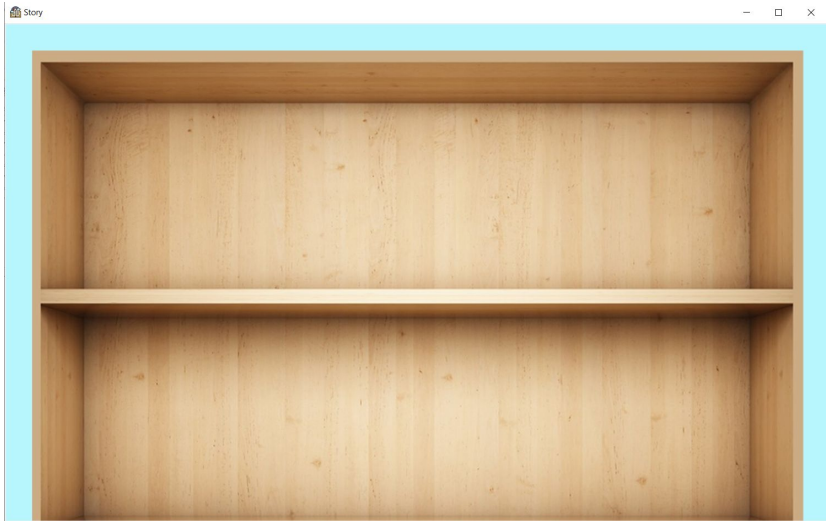
04

04

GUI

- Tkinter
- PIL(pillow)
- time
- pygame
- mutagen.mp3

Background




```
root = Tk()
root.title("Story")
root.iconbitmap("fairytale.ico")
root['background']='#b8f6fd'

canv = Canvas(root, width=1190, height=720, bg='#b8f6fd')
canv.grid(row=0, column=3)

img = ImageTk.PhotoImage(Image.open("bs3.png"))   # PIL solution
canv.create_image(40, 40, anchor=NW, image=img)
```

01

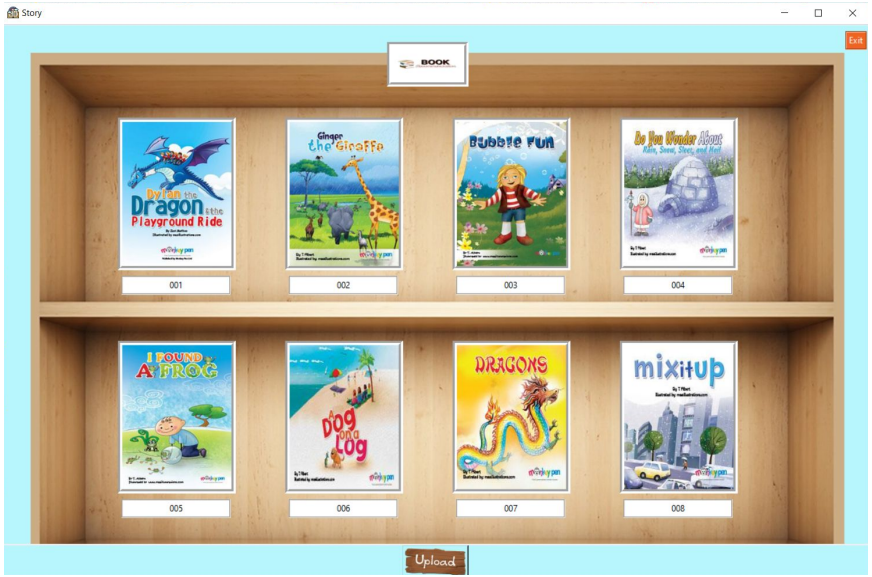
02

03

04

Book cover on shelf

```
ph1 = Image.open('011.jpg')
ph1 = ph1.resize((150,200),Image.ANTIALIAS)
my_ph1 = ImageTk.PhotoImage(ph1)

my_ph11 = Label(root,borderwidth=6, relief="groove",image = my_ph1)
my_ph11.place(x=160,y=130)

ph2 = Image.open('002.jpg')
ph2 = ph2.resize((150,200),Image.ANTIALIAS)
my_ph2 = ImageTk.PhotoImage(ph2)

my_ph22 = Label(root,borderwidth=6, relief="groove",image = my_ph2)
my_ph22.place(x=390,y=130)
```



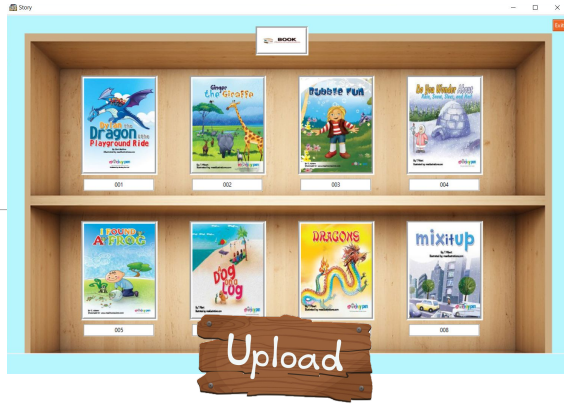

```
def cilck1(k):
    global folder
    folder = list_folder[k]
    return folder
```

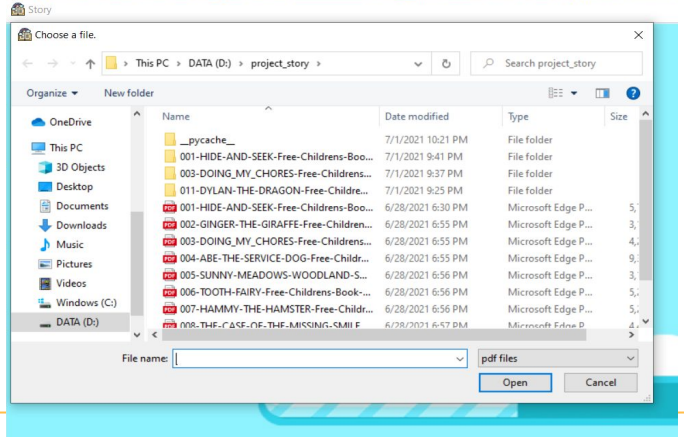
```
list_folder = ['011-DYLAN-THE-DRAGON-Free-Childrens-Book-By-Monkey-Pen','002-GINGER-THE-GIRAFFE-Free-Childrens-Book-By-Monkey-Pen',
'019-BUBBLE-FUN-Free-Childrens-Book-By-Monkey-Pen','021','022','023','025','016']
folder = ''

button1 = Button(root, width=20, text='001',relief="groove",bg='White', command= lambda :[display(cilck1(0))]).place(x=165,y=350)

button2 = Button(root, width=20, text='002',relief="groove",bg='White', command=lambda : [display(cilck1(1))]).place(x=395,y=350)
```

01 02 03 04

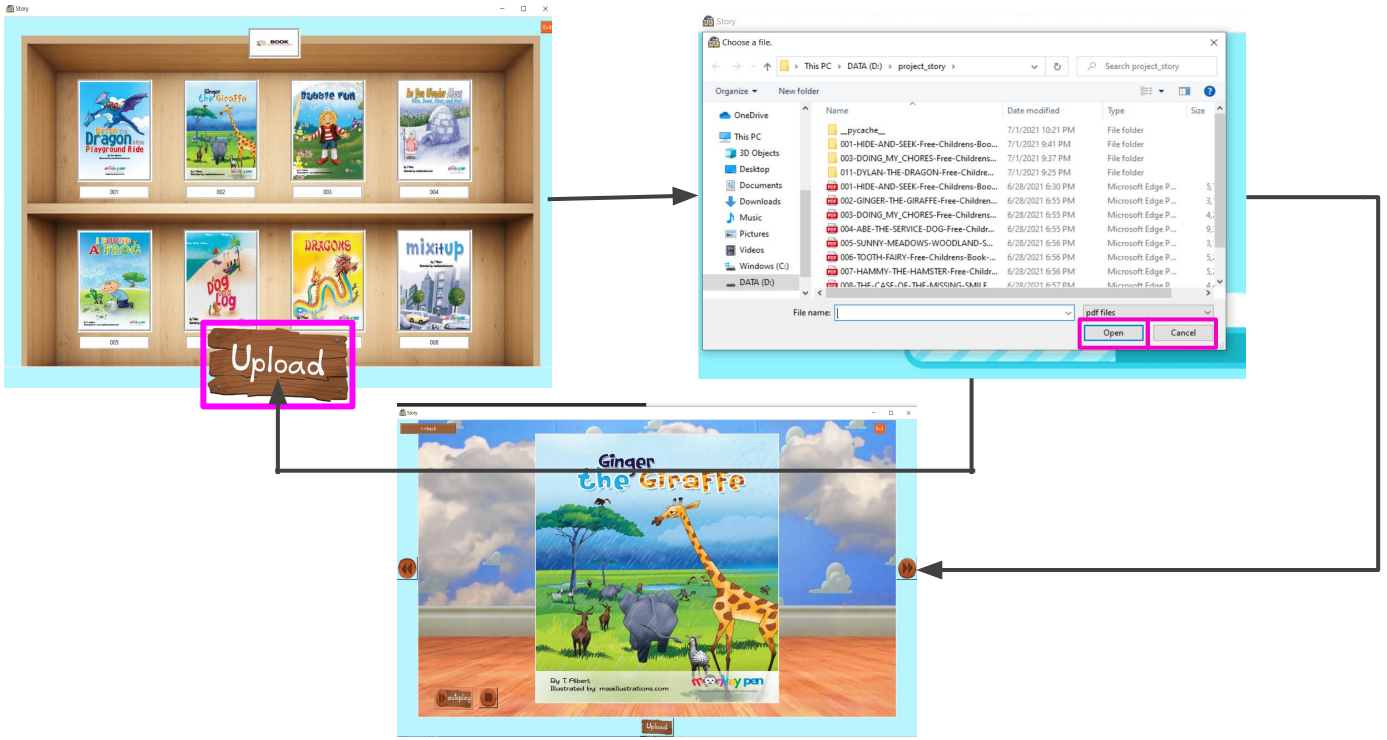
Function for upload file

```
def OpenFile():
    global folder
    global run
    name = filedialog.askopenfilenames(initialdir="",
                           filetypes =(("pdf files","*.pdf"),("All Files","*.*")),
                           title = "Choose a file."
                           )
    print(name)
    if name == '':
        clear_frame()
        display(folder)
        button_upload = Button ( root, text='Upload',image= my_img3 ,bg='#b8f6fd', fg='White',
                                 command=lambda: [stop_autoplay(),loading_img(),OpenFile()
                                                  ,runingg(),complete(),display(folder)]).grid(row=1, column=3)
        run = False

    else:
        fol_name = name[0].split('/')[-1].rsplit('.')[0]
        story_name = name[0].split('/')[-1]
        folder_name = m.pdf_to_image(name[0], fol_name)
        text = m.get_text_from_im(folder_name)
        folder = folder_name
        print(folder)
    return folder
```





Function for upload file



01

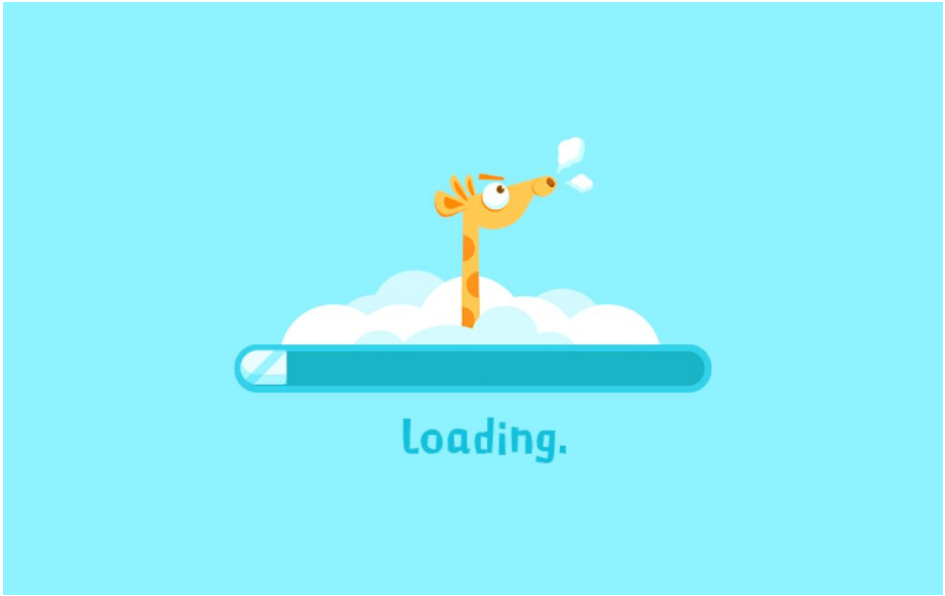
02

03

04

Pop up loading

```
def clear_frame():
   for widgets in root.winfo_children():
      widgets.destroy()

def loading_img():
    clear_frame()
    lbl = pg.ImageLabel(root)
    lbl.grid(row = 0, column=3)
    lbl.load('loading3.gif')
    lbl.place_forget()
    return
```



01

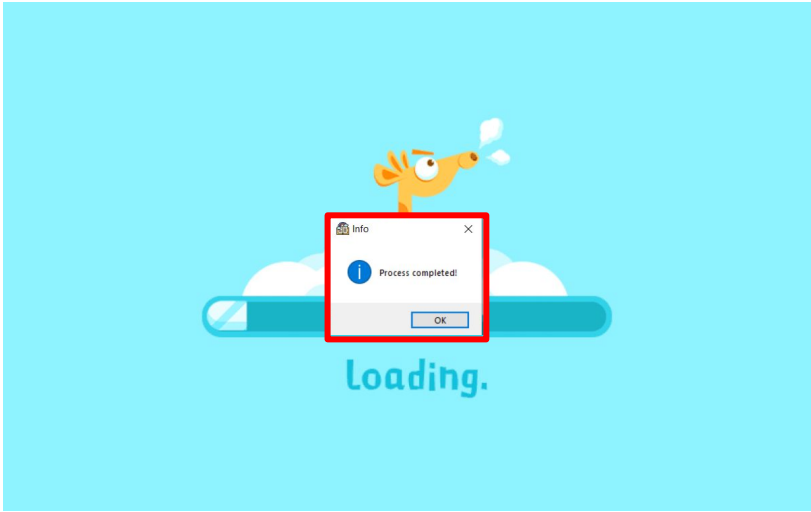
02

03

04

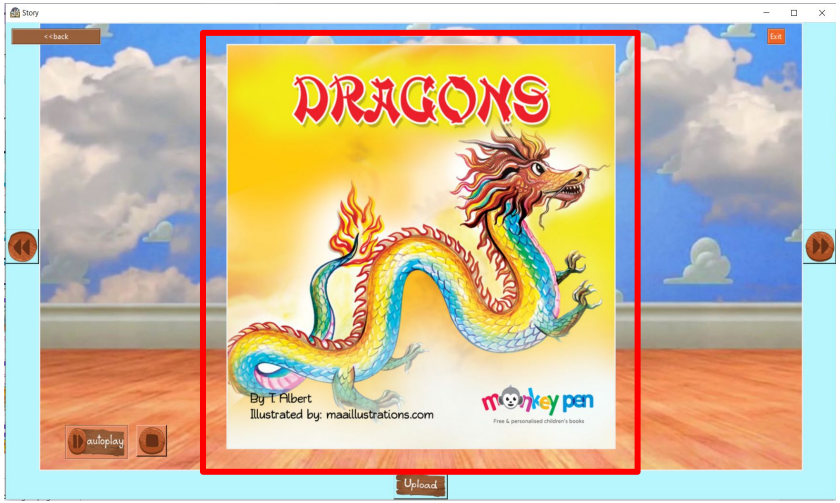
Process upload file

```
def complete():
    global run
    if run:
        messagebox.showinfo('Info', 'Process completed!')
    return

def runingg():
    global run
    if run:
        time.sleep(1)
    return

def show():
    global run
    if run:
        myLabel = Label(root, text=clicked.get()).place(x=250, y=30)
```



Get images

```
def open_image(folder):
    imger_list = []
    condition = True
    var = {}
    var2 = {}
    i = 0
    while condition:
        if path.exists(f"{folder}//image{i}.png"):
            image_path = f"{folder}//image{i}.png"
            var["my_img{0}".format(i)] = Image.open(image_path)
            imger_list.append(resize_image(var["my_img{0}".format(i)], i ,var2))
        else:
            condition = False
        i+=1
    return imger_list
```



Get sound

```
def open_sound(folder):
    sound_list = []
    condition = True
    i = 0
    while condition:
        if path.exists(f"{folder}//sound//sound{i}.mp3"):
            image_path = f"{folder}//sound//sound{i}.mp3"
            sound_list.append(image_path)
        else:
            condition = False
        i+=1
    return sound_list

def resize_image(image, i, var2 ):
    var2["new_pic{0}".format(i)] = ImageTk.PhotoImage(image.resize((650, 650), Image.ANTIALIAS))
    return var2["new_pic{0}".format(i)]
```

Function play sound

```
def playfirst(sound_list):
    mixer.init()
    mixer.music.load(sound_list[0])
    mixer.music.play()

def play_forward(image_number,sound_list):
    mixer.init()
    mixer.music.load(sound_list[image_number-1])
    mixer.music.play()
```

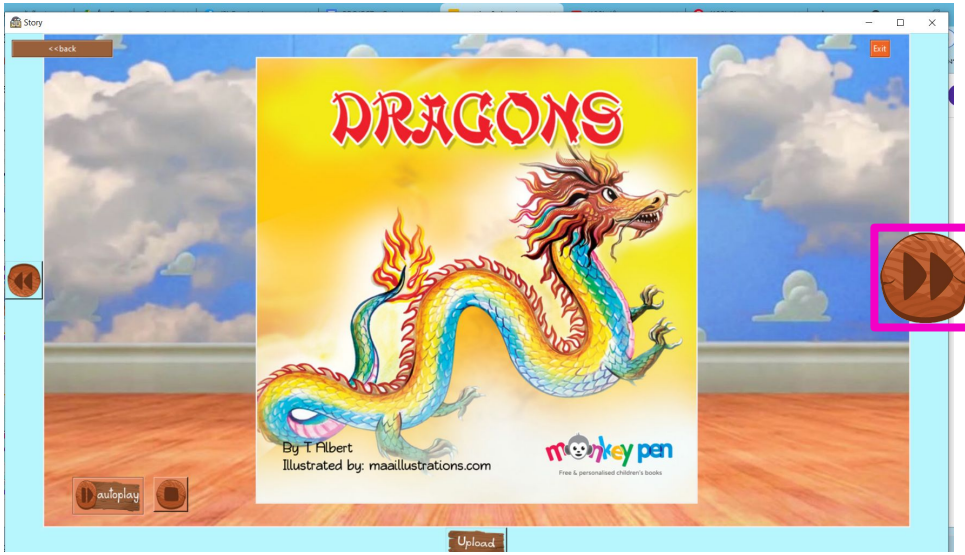
01
02
03
04

Display


```
def display(folder):
    global imger_list
    global sound_list
    global run

    if run:
        clear_frame()
        print(folder)

        imger_list = open_image(folder)

        lbl = pg.ImageLabel(root)
        lbl.grid(row = 0, column=3)
        lbl.load('bk4.jpg')

        my_lable = Label(image =imger_list[0])
        my_lable.grid(row = 0, column = 3)
        sound_list = open_sound(folder)
        my_lable.bind("<Button-1>", playfirst(sound_list))
        button_back = Button(root, text = "<<" ,image= my_img1 ,bg='#b8f6fd', command= lambda:[back(1),play_forward(1,sound_list)]).grid(row=0, column=1)
        button_upload = Button ( root, text='Upload',image= my_img3 ,bg='#b8f6fd', fg='White', command=lambda: [stop_autoplay(),loading_img(),OpenFile(),runingg(),complete(),display(folder)
        button_forward = Button(root, text=">>", image= my_img2 ,bg='#b8f6fd',command=lambda: [forward(2,imger_list),play_forward(2,sound_list)]).grid(row=0, column=4)
        button_autoplay = Button(root, text= 'autoplay',image= my_img4 , bg='#e5885d',relief="groove", command = lambda: [start_autoplay(),root.after(0, autoplay)]).place(x=100,y=650)
        button_stopplay = Button(root, text= 'stop play',bg='#e5885d', image= my_img5 ,command = lambda: stop_autoplay()).place(x=220,y=650)
        exit_button = Button(root, text="Exit", bg = '#f06524',fg='White',relief="groove",  command= lambda: [stop_autoplay(),root.destroy()]).place(x =1280, y=10)
        button_backward = Button(root, width=20, text='<<back', bg='#98603b',relief="groove",fg='White', command=lambda : [stop_autoplay(),prevPage()]).place(x=10,y=10)
    return
```

Function forward button

```
def playfirst(sound_list):
    mixer.init()
    mixer.music.load(sound_list[0])
    mixer.music.play()

def play_forward(image_number,sound_list):
    mixer.init()
    mixer.music.load(sound_list[image_number-1])
    mixer.music.play()
```




```
def forward(image_number,imger_list):
    global my_lable
    global button_forward
    global button_back

    # my_lable.grid_forget()
    my_lable = Label(image= imger_list[image_number-1])

    button_forward = Button(root, text = ">>",image= my_img2 ,bg='#b8f6fd',command = lambda: [forward(image_number+1,imger_list),play_forward(image_number+1,sound_list)])
    button_back = Button(root, text = "<<",image= my_img1 ,bg='#b8f6fd',command = lambda: [forward(image_number-1,imger_list),play_forward(image_number-1,sound_list)])

    if image_number == len(imger_list):
        button_forward = Button(root, text=">>",image= my_img2 ,bg='#b8f6fd',state = DISABLED)


    my_lable.grid(row = 0, column = 3)
    button_back.grid(row=0, column=1)
    button_forward.grid(row=0, column=4)
```

Function back button


```
def back(image_number):
    global my_lable
    global button_forward
    global button_back
    # my_lable.grid_forget()
    my_lable = Label(image= imger_list[image_number-1])
    button_forward = Button(root, text = ">>",image= my_img2 ,bg='#b8f6fd',command = lambda: [forward(image_number+1,imger_list),play_forward(image_number+1,sound_list)])
    button_back = Button(root, text = "<<",image= my_img1 ,bg='#b8f6fd',command = lambda: [forward(image_number-1,imger_list),play_forward(image_number-1,sound_list)])

    if image_number == 0:
        button_back = Button(root, text=">>",image= my_img1 ,bg='#b8f6fd',state = DISABLED)

    my_lable.grid(row = 0, column = 3)
    button_back.grid(row=0, column=1)
    button_forward.grid(row=0, column=4)
```

Autoplay and Pause button


```
def play_auto(s):
    mixer.init()
    mixer.music.load(s)
    mixer.music.play()
```

```
def stop_sound():
    mixer.music.stop()
```

```
i = 0
def autoplay():
    global i
    if running:
        try:
            my_lable = Label(root, image=imger_list[i])
            my_lable.grid(row = 0, column = 3)
            my_lable.bind("<Button-1>", play_auto(sound_list[i]))
            audio = MP3(f"{folder}//sound//sound{i}.mp3")
            time = (round(audio.info.length)*1000)+2000
            root.after(time, autoplay)
        except:
            root.after(5000, autoplay)
        i+=1


def start_autoplay():
    global running
    running = True
```

```
def stop_autoplay():
    global running
    running = False
    stop_sound()
```

01 02 03 04

DEMO

01

02

03

04

การแบ่งหน้าที่

		รับหน้าที่	
620710405	ณัฐธิดา ลาภธนชัย	ประมวลผลภาพ เสียงและตัวอักษร, Presentation	รายงาน, DEMO video,
620710407	เพชรัตน์ สุขอุบล		รายงาน
620710408	สุธาทิพย์ แย้มกลิ่น		ทำหน้าBook recommendation(Gui)
620710485	ชลธิชา ยาศรี		ทำหน้าแสดงผลนิทาน(Gui)
620710745	อาทิตยา ชมทอง		รายงาน

Thank You!!

Presented by

620710405 นางสาวณัฐธิดา ลาภธนชัย

620710407 นางสาวเพชรัตน์ สุขอุบล

620710408 นางสาวสุธาทิพย์ แย้มกลิ่น

620710485 นางสาวชลธิชา ยาศรี

620710745 นางสาวอาทิตยา ชมทอง

01

02

03

04