

CS159-Programming Report for ‘Cupid VS TOFEL’

Group name: 4.3 ATLAS

Members: 王智慧、朱瑞雯、钱心悦

Teacher: Bao Yang

13/6/2020

目录

1 背景	3
1.1 选题原因	3
1.2 项目介绍	3
1.3 Library 介绍	5
1.3.1 Pygame	5
1.3.2 OpenCV	5
1.3.3 PIL	5
1.3.4 Pyaudio	5
1.3.5 Baidu-aip	6
2 代码解释	6
2.1 游戏主界面	6
2.1.1 游戏主界面	6
2.1.2 子弹	8
2.1.3 小怪兽	9
2.2 使用人脸识别操纵飞机（小天使）	10
2.3 生成敌人（托福小怪兽）	10
2.3.1 随机生成托福单词	10
2.3.2 生成托福小怪兽图片	11
2.4 通过语音触发子弹（爱心）	12
2.4.1 语音输入	12
2.4.2 语音识别	12
2.4.3 判断是否与单词相同	13
3 局限与改进	14
3.1 局限性	14
3.2 优化设计	15
4 小结	15
5 描述各自的工作	15
6 参考文献	16
7 附录	16

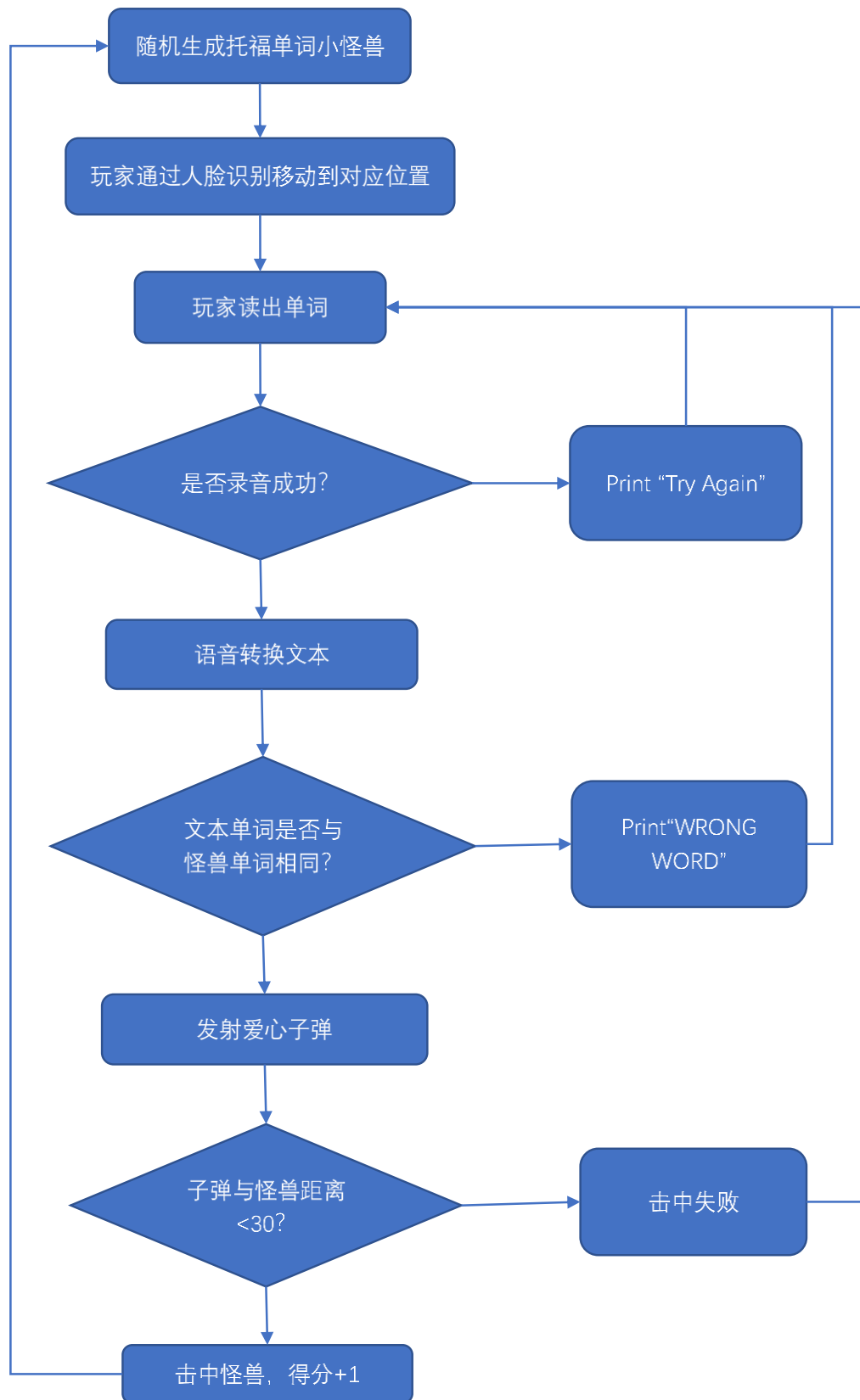
1 背景

1.1 选题原因

随着出国留学深造的人数增加且趋于低龄化,托福等语言水平考试逐渐普及。目前留学机构所开设的学习课程更偏重于教授考试技巧,然而语言的学习离不开单词的积累。除了大量阅读做题积累词汇量,我们希望能够寻找一种寓教于乐的方式,使考生们,尤其是年龄较小的考生,更为轻松地识记这些较难的托福词汇。

1.2 项目介绍

该项目改编自经典小游戏飞机大战,玩家操纵小天使移动,发射爱心击中带有托福单词的小怪兽,用爱心治愈它们。游戏使用人脸识别技术与语音识别模块来达成。当游戏进行时,玩家需要指定人脸通过物体追踪来操纵小天使作为移动的对象。同时,玩家需要准确读出英文单词才能够发射爱心击中小怪兽并治愈它。若玩家一次性得到 5 分,则认为玩家获胜。



项目流程图

1.3 Library 介绍

1.3.1 Pygame

Pygame 是一个用于 SDL 多媒体库的 Python 包装模块。它包含允许您使用 SDLs 支持来播放光盘的 python 函数和类，音频和视频输出，键盘，鼠标和操纵杆输入。在这个程序中，我们使用 pygame 来制作游戏的主界面。

1.3.2 OpenCV

开源计算机视觉库是在 BSD 许可下发布的它对学术和商业用途都是免费的。OpenCV2 是为高效的计算机计算和实时应用而设计的。在这个程序中，我们使用 openCV2 来进行人脸识别与追踪，从而达成操纵小天使的作用。

我们使用 CascadeClassifier 函数来制定人脸识别器。CascadeClassifier 为 OpenCV 中 cv namespace 下用来做目标检测的级联分类器的一个类。该类中封装的目标检测机制，简而言之是滑动窗口机制+级联分类器的方式。

1.3.3 PIL

Python 映像库(缩写为 PIL) (在新版本中称为 Pillow) 是 Python 编程语言的一个免费、开放源码的附加库，为 Python 解释器添加了图像处理功能。这个库支持许多文件格式，并提供强大的图像处理和图形功能。

1.3.4 Pyaudio

Pyaudio 主要有以下一些功能：提取特征——关于时域信号和频域信号的提取；训练并且使用分类器——监督学习，并用已有的训练集来进行训练；语音分割功能——按固定大小分割、静音检测、语音聚类、语音缩略图等进行分割；内容关系可视化——将语音信号映射到一个回归值。在本项目中，暂时只使用到了 Pyaudio 的语音录入功能。

1.3.5 Baidu-aip

百度语音识别库，可以将语音识别为文字，能够用于语音交互、语音内容分析、智能硬件、呼叫智能中心客服等等。通过调用接口，能够将人类语音中的词汇转换为计算机可读的输入，如按键、二进制编码或者字符序列等等。在本项目中，通过调用接口，将语音输入通过文件，转化为文本输出。

2 代码解释

2.1 游戏主界面

2.1.1 游戏主界面

首先，我们使用 pygame 搭建了游戏的主界面，我们使用 pygame 的内置函数确定了游戏主界面的大小、标题、图标、背景图片。

```
#初始化界面
pygame.init()
screen = pygame.display.set_mode((800,600)) #屏幕大小
pygame.display.set_caption('Cupid VS TOFEL')
icon = pygame.image.load('ufo.png') #icon
pygame.display.set_icon(icon)
bgImg = pygame.image.load('1.png') #背景图片
```

我们使用 pygame 内置函数对游戏玩家操纵的 Cupid 和游戏得分显示进行了初始化，同时我们创造了 prevention () 函数，防止 Cupid 出界。

```
#Aircraft
playerImg = pygame.image.load('Cupid_now.png')
playerX = 400
playerY = 480

#score
score = 0
font1 = pygame.font.Font('freesansbold.ttf', 32)
```

```

def prevention():
    global playerX
    #防止飞机出界
    if playerX > 850:
        playerX = 850
    if playerX < 0 :
        player = 0

```

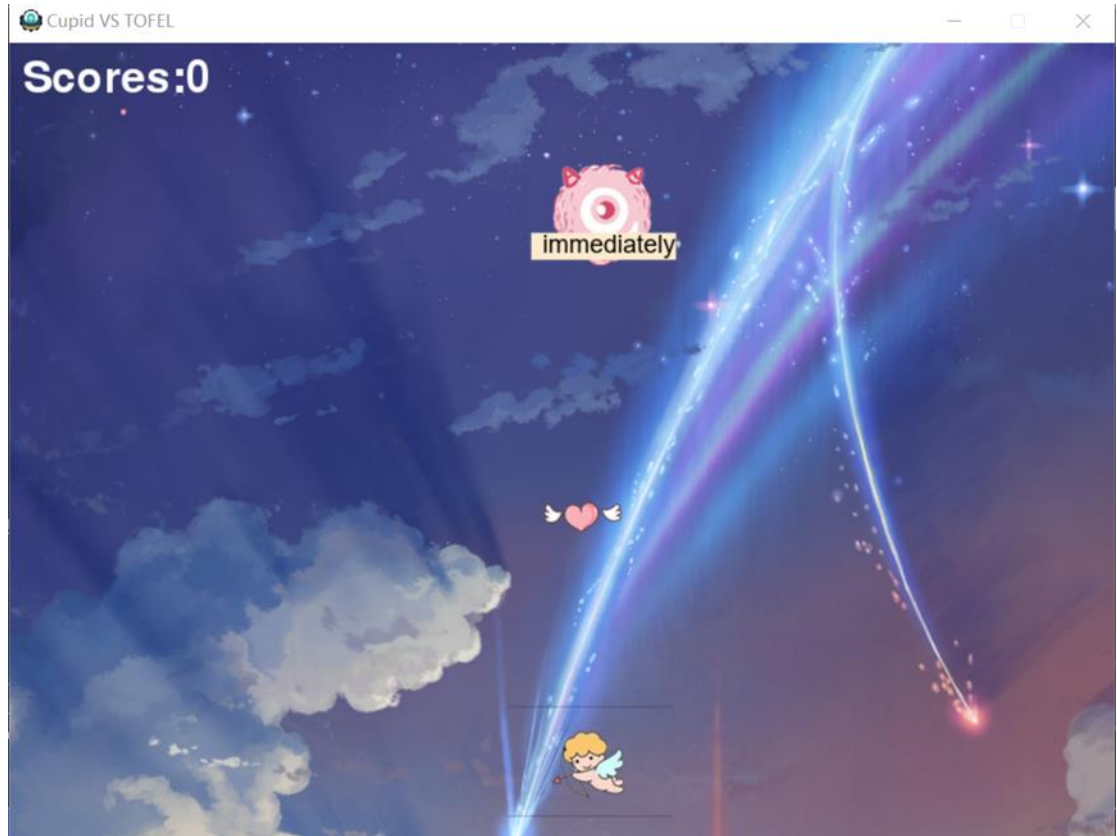
我们定义了判断游戏胜负 `check_is_win()` 与 `check_is_over()` 的函数。当玩家获得 10 分，我们使用 `screen.blit()` 函数显示 "You Win!"，并同时终止敌人移动；当敌人越过边界时，我们判定游戏失败，并使用 `screen.blit()` 函数显示 "Game over!"

```

is_over = False
font2 = pygame.font.Font('freesansbold.ttf', 64)
def check_is_win():
    if score == 10:
        text = "You Win"
        render = font2.render(text, True, (255, 0, 0))
        screen.blit(render, (200, 250))
        for e in enemies:
            e.step = 0
def check_is_over():
    if is_over:
        text = "Game Over!"
        render = font2.render(text, True, (255, 0, 0))
        screen.blit(render, (200, 250))
        bullets

```

到此，我们制作了游戏的主界面与主元素！



2.1.2 子弹

接下来我们创造了子弹类（即爱心），为了设定爱心击中怪兽，我们定义了 `distance()` 函数来衡量距离，当爱心与 TOFEL 怪兽的距离小于 30 时，我们认为爱心击中了 TOFEL 怪兽，并计得分+1，同时，怪兽被击中后会重新生成带有不同单词的新怪兽。

```
def distance(bx, by, ex, ey):  
    a = bx - ex  
    b = by - ey  
    return math.sqrt(a*a + b*b)
```



```

class Bullet():
    def __init__(self):
        self.img = pygame.image.load('heart_now.png')
        self.x = playerX + 16
        self.y = playerY - 10
        self.step = 10

    def hit(self):
        global score
        for e in enemies:
            if (distance(self.x, self.y, e.x, e.y) < 40):
                bullets.remove(self)
                e.reset() #秒投胎
                score += 1

bullets = []

def show_bullets():
    for b in bullets:
        screen.blit(b.img, (b.x, b.y))
        b.hit()
        b.y -= b.step

    if b.y < 0:
        bullets.remove(b)

```

2.1.3 小怪兽

我们创造了怪兽类，定义了他的移动与重置，并使用 random 函数使得他出现的位置随机。

```

class Enemy():
    def __init__(self):
        self.img = pygame.image.load("monster_now.png")
        self.x = random.randint(100, 700)
        self.y = 0
        self.step = random.randint(1, 2)

    def reset(self):
        TOFEL_word_enemy = TOFEL_words[random.randint(0, 571)]
        image = Image.open('monster1.png')
        image = image.resize((120, 106), Image.ANTIALIAS)
        draw = ImageDraw.Draw(image)
        myfont = ImageFont.truetype('arial.ttf', 18, encoding="arial")
        draw.text((16, 65), TOFEL_word_enemy, font = myfont, fill = fillcolor)
        image.save("monster_now" + ".png")
        self.img = pygame.image.load("monster_now.png")
        self.x = random.randint(100, 700)
        self.y = 0

enemies = []
for i in range(number_of_enemies):
    enemies.append(Enemy())

```

2.2 使用人脸识别操纵飞机（小天使）

首先，我们使用 openCV 中的 `cv2.VideoCapture` 函数读取玩家的内置摄像头。

```
frameWidth = 550
frameHeight = 350
cap = cv2.VideoCapture(0)
cap.set(3, frameWidth)
cap.set(4, frameHeight)
cap.set(10, 150)
```

随后，我们定义了人脸识别器，通过读取“haarcascade_frontalface_default.xml”文件，人脸识别器可以成功识别人脸。然后，我们通过读取玩家视频数据，将图片通过 `cv.COLOR_BGR2GRAY()` 将图片转为灰度图并使用人脸识别器进行识别，在人脸处标出方框，通过方框的位置制定飞机的 x 坐标，进行横向移动。

```
#人脸识别
faceCascade= cv2.CascadeClassifier("haarcascade_frontalface_default.xml")

——*success, img = cap.read()
——*img = cv2.flip(img, 1)
——*imgGray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
——*faces = faceCascade.detectMultiScale(imgGray, 1.1, 4)
——*for (x, y, w, h) in faces:
——*——*cv2.rectangle(img, (x, y), (x+w, y+h), (255, 0, 0), 2)
——*——*playerX = x + w//2 #还需根据画布再做调整
——*——*playerY = 480
```

2.3 生成敌人（托福小怪兽）

2.3.1 随机生成托福单词

```
f = open("TOFEL words.txt", "r")
text = f.read()
TOFEL_words = text.split()
TOFEL_word_enemy = TOFEL_words[random.randint(0,571)]
```

自网络摘录托福核心词汇 570 个并保存 txt 文件，命名为“TOFEL words.txt”。打开文件制成名称为“TOFEL_words”的 list，再使用 `random.randint` 随机取得一个单词。

2.3.2 生成托福小怪兽图片

```
image = Image.open('monster.png')
draw = ImageDraw.Draw(image)
draw.rectangle((40,350,578,450), fill = (255,235,205))
image.save("monster1" + ".png")
```

```
image = Image.open('monster1.png')
image = image.resize((120,106),Image.ANTIALIAS)
draw = ImageDraw.Draw(image)
myfont=ImageFont.truetype('arial.ttf',18,encoding="arial")
fillcolor = ImageColor.getrgb("black")
draw.text((16,65), TOFEL_word_enemy, font = myfont, fill = fillcolor)
image.save("monster_now" + ".png")
```

使用 PIL 包中 Image, ImageDraw, ImageFont, ImageColor 模块来实现图片处理。draw.rectangle 指定位置与颜色给图片添加一个杏色的框，使单词更加醒目。Image.resize 指定改变后图像的大小，ANTIALIAS 指生成图片质量最高；ImageFont.truetype 指定字体为“arial”且大小为 18；ImageColor.getrgb 直接指定文字颜色为黑色。Draw.text 表示在图片上添加文字，括号中四个变量依次为文字位置、文字内容、字体设置、文字颜色。最后 image.save 保存为新的图片文件。

2.4 通过语音触发子弹（爱心）

2.4.1 语音输入

```
def check_words():  
    # 录音  
    CHUNK = 1024  
    FORMAT = pyaudio.paInt16  
    CHANNELS = 1  
    RATE = 16000  
    RECORD_SECONDS = 3  
    WAVE_OUTPUT_FILENAME = "TFWORD.pcm"  
  
    p = pyaudio.PyAudio()  
  
    stream = p.open(format=FORMAT,  
                    channels=CHANNELS,  
                    rate=RATE,  
                    input=True,  
                    frames_per_buffer=CHUNK)  
  
    frames = []  
  
    for i in range(0, int(RATE / CHUNK * RECORD_SECONDS)):  
        data = stream.read(CHUNK)  
        frames.append(data)  
  
    stream.stop_stream()  
    stream.close()  
    p.terminate()  
  
    wf = wave.open(WAVE_OUTPUT_FILENAME, 'wb')  
    wf.setnchannels(CHANNELS)  
    wf.setsampwidth(p.get_sample_size(FORMAT))  
    wf.setframerate(RATE)  
    wf.writeframes(b''.join(frames))  
    wf.close()
```

语音部分的程序主体思路为：从用户端获得语音输入——将语音转化为文本计入一个变量中——将该变量与选中的单词作对比——相同则发射子弹；不同则提示单词阅读错误；识别失败则提示用户再次输入。

首先，从用户端获得语音输入，在这里采用的方法是：通过 pyaudio 获得用于的语音，设置好 16bit 位深、单声道与 16k 采样率，录音时长为 3s，最后将文件命名保存。最后，通过 open 函数开始录制语音，按单位写入文件，并将文件保存。

2.4.2 语音识别

```
# 读取文件  
def get_file_content(filePath):  
    with open(filePath, 'rb') as fp:  
        return fp.read()
```

```

APP_ID = '20280341'
API_KEY = 'G9s6lo8AhulnIec8Vyg077K0'
SECRET_KEY = '6PFdo8H9d0jtUZS8oLgDo15XwNYybRGU'
client = AipSpeech(APP_ID, API_KEY, SECRET_KEY)

# 识别本地文件
a = client.asr(get_file_content('TFWORD.pcm'), 'pcm', 16000, {
    'dev_pid': 1737,
})

```

通过调取百度 AIP 接口，读取语音文件的内容。并使用 asr 将文件内容识别为文字进行输出，参数分别为文件名与文件类型，采样频率，目标语言，“1737”即为将语音转为无标点的英语输出。

2.4.3 判断是否与单词相同

```

#输出单词
word_half = a.get('result')
if type(word_half) == type(None):
    print("TRY AGAIN!")
else:
    word = ''.join(word_half)
    print(word)
—— #判断单词是否正确
    if word == TOFEL_word_enemy:
        bullets.append(Bullet())
    else:
        print("WRONG WORD!")

```

由于默认返回是字典的形式，在这里单独返回的结果取出，并从列表形式转化为字符串的形式以便于与随机选中的 TOEFL 单词进行比较。首先考虑单词录入失败的问题，当单词录入失败，比如未收集到语音，或语音过于嘈杂无法识别时，返回的字典中不会出现 'result' 结果，即列表为空，此时设置让用户再次录入；当单词录入成功时，即比较单词与选定单词是否相同，若相同则发射子弹，若不相同则提示用户单词发音错误。

2.5 游戏主循环

游戏主循环中，我们调用了先前定义好的函数，并且通过键盘操纵实现了退出和启用语音识别的功能。

```

——»#游戏主循环

while 1:
——»screen.blit(bgImg, (0,0))
——»for event in pygame.event.get():
——»——»if event.type == pygame.QUIT:sys.exit()

——»——»#键盘按下
——»——»if event.type == pygame.KEYDOWN:
——»——»——»if (event.key == pygame.K_SPACE):
——»——»——»——»check_words()
——»——»——»if (event.key == pygame.K_ESCAPE):
——»——»——»——»pygame.quit()
——»——»——»——»quit()
——»——»if (event.type == pygame.QUIT):
——»——»——»pygame.quit()
——»——»——»quit()

——»success, img = cap.read()
——»img = cv2.flip(img, 1)
——»imgGray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
——»faces = faceCascade.detectMultiScale(imgGray, 1.1, 4)
——»for (x, y, w, h) in faces:
——»——»cv2.rectangle(img, (x, y), (x+w, y+h), (255, 0, 0), 2)
——»——»playerX = x + w//2 #还需根据画布再做调整
——»——»playerY = 480

——»show_score()
——»cv2.imshow("Player",img)
——»if cv2.waitKey(1) & 0xFF == ord('q'):
——»——»break
——»screen.blit(playerImg, (playerX, playerY))

——»prevention()
——»show_enemy()
——»show_bullets()
——»check_is_over()
——»check_is_win()

——»pygame.display.update()

```

3 局限与改进

3.1 局限性

人脸识别的精度不够，必须在较整洁的背景下进行游戏，否则容易产生误读，且识别对象不够广泛。

语音输入前需要按空格键提示输入。物体追踪移动范围较大，可能导致距离设备较远不方便按空格键的问题。

语音输入、语音识别有一定时间间隔，较为影响游戏体验。

3.2 优化设计

提高人脸识别准确性。设想一是采用更多的人脸样本进行训练，防止误读、漏读的发生。设想二是采用物体追踪的技术实现多对象识别。

更换其他较为便捷的方式提示输入。设想一是使用激光笔等蓝牙连接设备远程连接电脑，给游戏腾出移动空间。设想二是指定动作提示输入，当玩家做出指定的肢体动作，经摄像头检测后自动识别，语音输入程序开始。

缩短语音输入、语音识别的时间间隔。更换其他更为可靠的语音识别包或加强电脑硬件设备。

4 小结

项目从生活出发，结合当下大热的托福考试单词与经典小游戏飞机大战，创造新的故事情境——小天使治愈托福小怪兽，实现经典再创造。项目总体达到预期要求的水平，在细节与游戏精度等方面略有不足。由于时间仓促，我们也未来得及掌握 Github 网站的使用方法，但我们决定，结课后再对这个项目进行优化升级，届时打算上传 Github 网站与大家共同交流改进。

同时非常感谢鲍杨老师不厌其烦的耐心指导！

5 描述各自的工作

王智慧：使用 pygame 搭建了游戏主界面、初始化了游戏的主要元素，如 Player、Bullet、Enemy 等类、使用 opencv 实现人脸识别以操纵玩家、撰写报告游戏主

界面、使用人脸识别操纵飞机、游戏主循环部分；录制 demo 视频。

朱瑞雯：尝试 google、bing、CMUsphinx、Azure 之后，最终通过 Baidu-aip 实现语音输入、语音识别功能、将识别单词与选中单词比较的功能；撰写报告语音识别模块部分；录制 demo 视频

钱心悦：制作敌人图片；游戏背景等图像处理来美化游戏界面；撰写报告概要、局限改进、小结部分；剪辑 demo 视频

6 参考文献

[1]<https://www.bilibili.com/video/BV16K411W7x9?t=4614>

[2]<https://www.bilibili.com/video/BV1i741137rM/?p=23&t=234>

[3]<https://ai.baidu.com/>

7 附录

```
import time, sys, random, pygame, math, os
from PIL import Image, ImageDraw, ImageFont, ImageColor
import cv2
from aip import AipSpeech
import pyaudio
import wave
```

```
t = time.gmtime()
```

```
#初始化界面
```

```
pygame.init()
```

```
screen = pygame.display.set_mode((800,600))#屏幕大小
```

```
pygame.display.set_caption('Cupid VS TOFEL')
```

```
icon = pygame.image.load('ufo.png') #icon
```

```
pygame.display.set_icon(icon)
```

```
bgImg = pygame.image.load('1.png')#背景图片
```



```

#人脸识别
faceCascade= cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
frameWidth = 550
frameHeight = 350
cap = cv2.VideoCapture(0)
cap.set(3, frameWidth)
cap.set(4, frameHeight)
cap.set(10, 150)

#Aircraft
playerImg = pygame.image.load('Cupid_now.png')
playerX = 400
playerY = 500

#score
score = 0
font1 = pygame.font.Font('freesansbold.ttf', 32)

def show_score():
    text = f'Scores: {score}'
    score_render = font1.render(text, True, (255,255,255))
    screen.blit(score_render, (10,10))

is_over = False
font2 = pygame.font.Font('freesansbold.ttf',64)
def check_is_win():
    if score == 10:
        text = "You Win"
        render = font2.render(text, True, (255,0,0))
        screen.blit(render,(200,250))
        for e in enemies:
            e.step = 0
def check_is_over():
    if is_over:
        text = "Game Over!"
        render = font2.render(text, True, (255,0,0))
        screen.blit(render,(200,250))
        bullets
font3 = pygame.font.Font('freesansbold.ttf',32)

# 读取文件
def get_file_content(filePath):
    with open(filePath, 'rb') as fp:

```

```

        return fp.read()

def check_words():
    #录音
    CHUNK = 1024
    FORMAT = pyaudio.paInt16
    CHANNELS = 1
    RATE = 16000
    RECORD_SECONDS = 3
    WAVE_OUTPUT_FILENAME = "TFWORD.pcm"

    p = pyaudio.PyAudio()

    stream = p.open(format=FORMAT,
                    channels=CHANNELS,
                    rate=RATE,
                    input=True,
                    frames_per_buffer=CHUNK)

    frames = []

    for i in range(0, int(RATE / CHUNK * RECORD_SECONDS)):
        data = stream.read(CHUNK)
        frames.append(data)

    stream.stop_stream()
    stream.close()
    p.terminate()

    wf = wave.open(WAVE_OUTPUT_FILENAME, 'wb')
    wf.setnchannels(CHANNELS)
    wf.setsampwidth(p.get_sample_size(FORMAT))
    wf.setframerate(RATE)
    wf.writeframes(b''.join(frames))
    wf.close()

    APP_ID = '20280341'
    API_KEY = 'G9s6lo8AhulnIec8Vyg077K0'
    SECRET_KEY = '6PFdo8H9dOjtUZS8oLgDo15XwNYybRGU'
    client = AipSpeech(APP_ID, API_KEY, SECRET_KEY)

    # 识别本地文件
    a = client.asr(get_file_content('TFWORD.pcm'), 'pcm', 16000, {
        'dev_pid': 1737,

```

```

    })

    #输出单词
    word_half = a.get('result')
    if type(word_half) == type(None):
        print("TRY AGAIN!")
        text1_word = "TRY AGAIN!"
        word1_render = font1.render(text1_word,True,(255,255,255))
        screen.blit(word1_render, (10,50))
    else:
        word = ".join(word_half)
        print(word)
        text2_word = "The word you said:" + word
        word2_render = font1.render(text2_word, True, (255,255,255))
        screen.blit(word2_render, (10,50))
    #判断单词是否正确
    if word == TOFEL_word_enemy:
        bullets.append(Bullet())
    else:
        print("WRONG WORD!")
        print(TOFEL_word_enemy)
        text3_word = "WRONG WORD!"
        word3_render = font1.render(text3_word, True, (255,255,255))
        screen.blit(word3_render, (10,90))

    #添加敌人
    number_of_enemies = 1

    f = open("TOFEL words.txt","r")
    text = f.read()
    TOFEL_words = text.split()
    TOFEL_word_enemy = TOFEL_words[random.randint(0,571)]

    image = Image.open('monster1.png')
    image = image.resize((120,106),Image.ANTIALIAS)
    draw = ImageDraw.Draw(image)
    myfont=ImageFont.truetype('arial.ttf',18,encoding="arial")
    fillcolor = ImageColor.getrgb("black")
    draw.text((16,65), TOFEL_word_enemy, font = myfont, fill = fillcolor)
    image.save("monster_now" + ".png")

    class Enemy():
        global TOFEL_word_enemy
        def __init__(self):

```

```

        self.img = pygame.image.load("monster_now.png")
        self.x = random.randint(150,650)
        self.y = 0
        self.step = random.randint(1, 2)
    def reset(self):
        TOFEL_word_enemy = TOFEL_words[random.randint(0,571)]
        image = Image.open('monster1.png')
        image = image.resize((120,106),Image.ANTIALIAS)
        draw = ImageDraw.Draw(image)
        myfont = ImageFont.truetype('arial.ttf',18,encoding="arial")
        draw.text((16,65), TOFEL_word_enemy, font = myfont, fill = fillcolor)
        image.save("monster_now" + ".png")
        self.img = pygame.image.load("monster_now.png")
        self.x = random.randint(100,700)
        self.y = 0

enemies = []
for i in range(number_of_enemies):
    enemies.append(Enemy())

def distance(bx, by, ex, ey):
    a = bx - ex
    b = by - ey
    return math.sqrt(a*a + b*b)

class Bullet():
    def __init__(self):
        self.img = pygame.image.load('heart_now.png')
        self.x = playerX + 16
        self.y = playerY - 10
        self.step = 10

    def hit(self):
        global score
        for e in enemies:
            if (distance(self.x, self.y, e.x, e.y) < 40):
                bullets.remove(self)
                e.reset()#秒投胎
                score += 1

bullets = []

def show_bullets():
    for b in bullets:

```

```

        screen.blit(b.img,(b.x,b.y))
        b.hit()
        b.y -= b.step

    if b.y < 0:
        bullets.remove(b)

def show_enemy():
    global is_over
    for e in enemies:
        screen.blit(e.img, (e.x,e.y))
        e.y += e.step
        if e.y > 450:
            is_over = True
            enemies.clear()

def prevention():
    global playerX
    #防止飞机出界
    if playerX > 850:
        playerX = 850
    if playerX < 0 :
        player = 0

    #游戏主循环
while 1:
    screen.blit(bgImg,(0,0))
    for event in pygame.event.get():
        if event.type == pygame.QUIT:sys.exit()

        #键盘按下
        if event.type == pygame.KEYDOWN:
            if (event.key == pygame.K_SPACE):
                check_words()
            if (event.key == pygame.K_ESCAPE):
                pygame.quit()
                quit()
        if (event.type == pygame.QUIT):
            pygame.quit()
            quit()

    success, img = cap.read()
    img = cv2.flip(img, 1)
    imgGray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

```

```

faces = faceCascade.detectMultiScale(imgGray, 1.1, 4)
for (x, y, w, h) in faces:
    cv2.rectangle(img, (x, y), (x+w, y+h), (255, 0, 0), 2)
    playerX = x + w//2 #还需根据画布再做调整
    playerY = 480

show_score()
cv2.imshow("Player",img)
if cv2.waitKey(1) & 0xFF == ord('q'):
    break
screen.blit(playerImg, (playerX, playerY))

prevention()
show_enemy()
show_bullets()
check_is_over()
check_is_win()

pygame.display.update()

```