Anoint

Alvin Mi & Rex Liu

June 2022

1 Introduction

Chatbots are used in everyday life. You can find them everywhere: on your phone, in messaging apps, in shopping websites, etc. Chatbots make our lives easier, users can find the information they need by responding to chatbots' questions without the need for human intervention.

Chatbots have evolved significantly over the last decades. They used to be text-based and worked like an interactive FAQ, but they are limited to reply to complex and unfamiliar questions. Chatbots nowadays usually are voice-based, and use much more advanced Natural Language Processing (NLP). Chatbots can be used in all different ways. They can set a reminder for your schedule, they can find you the closest fuel station, they can tell you the current weather condition, they can also provide general customer service support. All in all, chatbots have become a big part of software technology.

The values of chatbots are huge in the current world. According to research, the global chatbot market was valued at over $190 million in 2016 (Sweezey, 2018). One of the studies found that 15% of all consumers have engaged with a company via a chatbot in the past 12 months. Therefore it's not hard to see the significance of chatbots. It is widely used in tech jobs, and could also boost the economy by reducing unnecessary expenses for companies.

In this Science Fair, we are creating our own chatbot – **Alex** (Alex is a combination of the names of the two group members). We are trying to create a better human-computer interaction experience, so we used Python to write the program (and tkinter for the GUI). *Alex* has the ability to respond to user's messages, and *Alex* is created to help the users to revise. As of right now, it can work as a calculator, can search things on *Bing* or a Chinese encyclopedia (*baike.baidu.com*), can ask quizzes about specific topics (capitals, chemistry, computer science and biology) and can calculate how many days you have lived.This is an open-source project so all codes has been uploaded to *Github.com*, link=
`https://github.com/RexCHNNZ/Anoint`

2 Logic

We can explain what is happening in the program by using a flowchart:

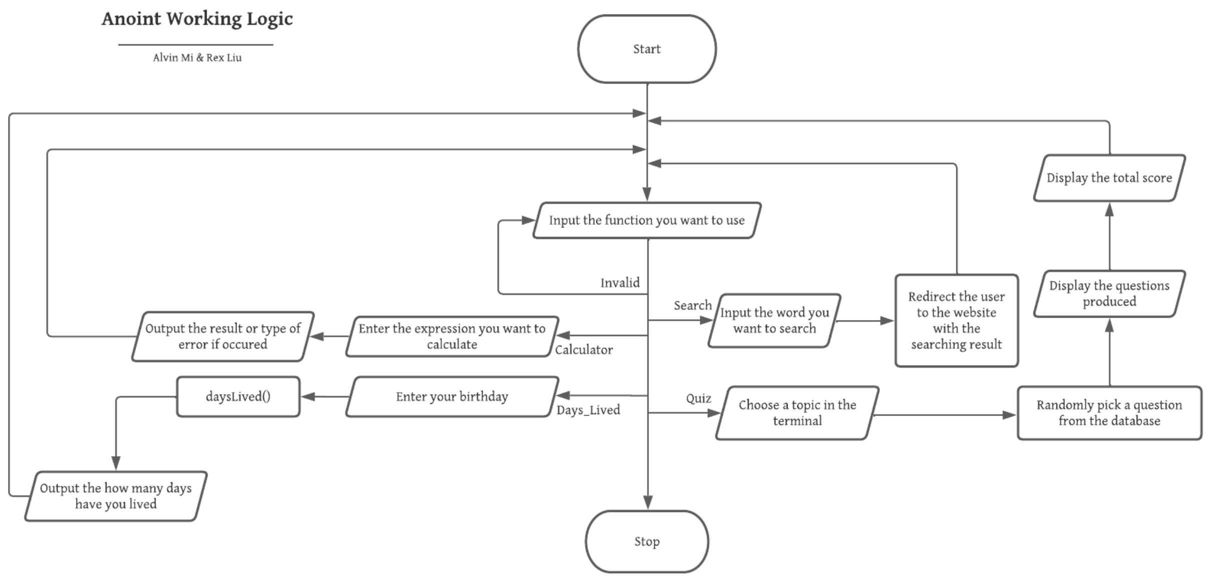

Anoint Working Logic
Alvin Mi & Rex Liu
Start
Input the function you want to use
Invalid
Search
Input the word you want to search
Redirect the user to the website with the searching result
Calculator
Enter the expression you want to calculate
Output the result or type of error if occured
Days_Lived
Enter your birthday
daysLived()
Output the how many days have you lived
Quiz
Choose a topic in the terminal
Randomly pick a question from the database
Display the questions produced
Display the total score
Stop

Figure 1: Anoint Working Logic

3 Module Library

In our program, we mainly use functions from the following modules:

- Tkinter: build the Graphical User Interface (GUI) for the user to interact with the chatbot
- Keyboard: allows user to press “Enter”to calculate the result in either Calculator Mode and Days Lived Mode
- Datetime: used in Days Lived Mode to import the current date and calculate difference between user’s birthday and today using the built-in function
- Webbrowser: used in Search/Encyclopedia Mode to open up the Uniform Resource Locator (URL) of the Bing Search/Baidu Encyclopedia webpage
- Sys: used to import functions we defined in other separate files to the main chatbot program
- Random: used to select a random greeting sentence and a random question from the list
- Json: JavaScript Object Notation, used to process the data as well as providing a better experience of reading data
- Pygame: to create the plane fight game in Python, providing easy access to the config/settings properties from all python modules, it supports normal and lazy initialization for each property.
- Traceback: provides a standard interface to extract, format and print stack traces of a python program
- OS: provides functions for interacting with the operating system

4 Quiz Database

To give out the quiz questions, we uses databases to store all available questions, which can also be edited by users as Alex’s code is open-source.
The databases are shown here.
Chemistry Database:

```
["Alex: What is the most reactive non-metal?", "fluorine"],
["Alex: What is the name of the first element in the Periodic Table?", "hydrogen"],
["Alex: What is the name of group 8 (helium, neon, argon etc.)?", "noble gas"],
["Alex: Which group is called the alkali metals?", "group 1"],
["Alex: What is the name of the gas that makes up 78 percent of air?", "nitrogen"],
["Alex: What is the boiling point of pure water in degrees Celsius?", "100"],
["Alex: What is the most reactive non-radioactive metal?", "potassium"],
["Alex: Which industrial process is used to make ammonia?", "haber process"],
["Alex: Which hydrocarbon group has the general formula CnH2n?", "alkenes"],
["Alex: Does the density of metals increase down group 1 (true of false)", "true"],
["Alex: What is the symbol of tin", "sn"],
["Alex: Which element has the symbol Pb", "lead"],
["Alex: Which carbon conducts electricity", "graphite"],
["Alex: What is the only liquid metal under room temperature", "mercury"],
["Alex: What is element 118 on the periodic table", "oganesson"],
["Alex: Which word is used to describe that metals can be drawn into wires", "ductile"],
["Alex: What is the name of KMnO4", "potassium permanganate"],
["Alex: What is the name of the ester formed from C2H5OH and CH3COOH", "ethyl ethanoate"],
["Alex: What is the equation of the catalyst used in the Contact Process", "V2O5"],
["Alex: What is the name of the catalyst used in Hydrogenation reaction", "nickel"]
```

Computer Science Database:

```
["Alex: Convert the binary number 111 to denary", "7"],
["Alex: Convert the hexadecimal number 3A to 8-bit binary", "00111010"],
["Alex: How many bits of data do serial data transmission transmit at one time?", "1"],
["Alex: Which direction is half-duplex data transmission?", "both directions but not at the same time"],
["Alex: Which touchscreen cannot be used with gloved hands?", "capacitive"],
["Alex: What is the full name of LCD?", "liquid crystal display"],
["Alex: Which technology does not use back-lighting?", "OLED"],
["Alex: Which register is used to hold results from calculations?", "acc"],
["Alex: Is address bus unidirectional or bidirectional?", "unidirectional"],
["Alex: Which shape is used to represent process in a flowchart?", "rectangle"],
["Alex: What is represented by a diamond shape in a flowchart?", "decision"],
["Alex: Which optical storage uses concentric tracks?", "dvd-ram"],
["Alex: What is the full name of DVD?", "digital versatile disk"],
["Alex: What is the sequence of registers down the FDE cycle?(use - to separate)", "pc-mar-mdr-cir"],
["Alex: True or false? Python doesn't have post-condition iteration structure", "true"],
["Alex: What is the full name of MAC(address)?", "media access control"],
["Alex: True or false? DLP technology uses micro-mirrors", "true"],
["Alex: What is the full name of URL?", "uniform resource locator"],
["Alex: Is firewall a software or a hardware or both?", "both"],
["Alex: What is the other separate layer in TLS beside handshake layer?", "record"]
```

Biology Database:

```
["Alex: What is one characteristic of living organisms that means irritability?", "sensitivity"],
["Alex: True or false? Vacuole never exists in animal cells", "false"],
["Alex: True or false? Active transport is a passive process", "false"],
["Alex: Which word is used to describe membrane that water crosses in osmosis?", "partially-permeable"],
["Alex: What is the colour for positive protein test?", "lilac"],
["Alex: What is the colour for positive starch test?", "blue-black"],
["Alex: What does DCPIP test for?", "vitamin c"],
["Alex: By which process do O2 and CO2 move between cells and capillaries?", "diffusion"],
["Alex: What is the basic unit of protein?", "amino acid"],
["Alex: Which part of a cell makes glucose?", "chloroplast"],
["Alex: Where does aerobic respiration occur in a cell? (use plural form)", "mitochondria"],
["Alex: True or false? Auxin causes cells to elongate", "true"],
["Alex: What is the shape of a DNA?", "double helix"],
["Alex: True or false? Meiosis produces haploid cells?", "true"],
["Alex: What are the bases present on the opposite strand of one with ACCTGAGC?", "tggactcg"],
["Alex: What's the process that occurs in the liver where excess amino acids are made to urea?",
"deamination"],
["Alex: Which chamber of the heart has the thickest wall?", "left ventricle"],
["Alex: What is another name for the right atrioventricular valve?", "tricuspid"],
["Alex: Where is bile stored?", "gall bladder"],
["Alex: Which substance is anaemia lacking?", "iron"],
["Alex: What is the name of the first section of the small intestine?", "duodenum"],
["Alex: Where is water absorbed in the human body?", "large intestine"],
["Alex: True or false? Nitrate ions are needed to make chlorophyll in plants", "false"],
["Alex: Which word means relaxing in the cardiac cycle: Diastole or Systole?", "diastole"],
["Alex: What happens to the external intercostal muscles when you exhale?", "relax"],
["Alex: Which part of the eye refracts light?", "cornea"],
["Alex: Which enzyme is released when blood glucose concentration is too high?", "insulin"],
["Alex: How does the zygote move down the oviduct?", "peristalsis"],
```

Physics Database:

```
["Which instrument is most suitable for measuring the thickness of a paper?", "Micrometer screw gauge"],
["What is ultrasound?", "Sound waves that are too high-pitched for humans to hear"],
["Which statement is not correct?", "Alpha-particles are used to detect cracks in metallic structures"],
["Which statement about electromagnetic waves is not correct?", "They travel at 340 m / s in air"],
["Which energy resource has the Sun as its only source of energy", "Oil"],
["A current is in a copper wire. Which particles are moving along the wire?", "Electrons"],
["Engine produces 240kJ of energy in 2minutes.What is the power of the engine?", "2kW"],
["What has happened to a negative rod if it becomes positively charged?", "Loses electrons"],
["Which statement about image of an object formed in a plane mirror is true?",
"It is the same size as the object"],
["What causes liquid level in a thermometer to rise when placed in hot water?", "The liquid expands"],
["Which quantity is weight an example of?","Force],
["What is the source of the Sun's energy?", "Nuclear fusion in the Sun's core],
["Which quantity is energy transferred in driving charge round a complete circuit?",
"Electromotive force"],
["Which property changes when temperature is measured with LiG thermometer?", "Volume"],
["What name is given to a region which the molecules are further apart?", "Rarefaction"],
["Which quantity is a vector?", "Acceleration"],
["Which energy resource does not have the Sun as the original source?", "Geothermal"],
["Which energy resource is not renewable?", "Nuclear fission"],
["Which quantity is not a vector?", "Temperature"]
```

Capital City Database:

```
["Alex: What is the capital city of India?", "new delhi"],
["Alex: What is the capital city of Canada?", "ottawa"],
["Alex: What is the capital city of Australia?", "canberra"],
["Alex: What is the capital city of Turkey?", "ankara"],
["Alex: Known as the windiest city, what is the capital city of New Zealand?", "wellington"],
["Alex: What is the capital city of Ireland?", "dublin"],
["Alex: What is the capital city of China?", "beijing"],
["Alex: What is the capital city of the United Kingdom?", "london"],
["Alex: What is the capital city of the United States of America?", "washington"],
["Alex: What is the capital city of Malaysia?", "kuala lumpur"],
["Alex: What is the capital city of France?", "paris"],
["Alex: What is the capital city of Germany?", "berlin"],
["Alex: What is the capital city of Spain?", "madrid"],
["Alex: What is the capital city of Italy?", "rome"],
["Alex: What is the capital city of South Korea?", "seoul"],
["Alex: What is the capital city of Japan?", "tokyo"],
["Alex: What is the capital city of the largest country in the world?", "moscow"],
["Alex: Cairo is the capital city of which country?", "egypt"],
["Alex: Which place in Southeast Asia is both a capital city and a country?", "singapore"],
["Alex: Which country hosts the 2022 FIFA World Cup and has Doha as its capital city?", "qatar"],
["Alex: What is the capital of the state which has a basketball team that won 2022 NBA playoffs?",
"sacramento"],
["Alex: Bern is the capital city of which European country?", "switzerland"],
["Alex: What is the capital city of Albania?", "tirana"],
["Alex: What is the capital city of Bulgaria?", "sofia"],
["Alex: What is the capital city of Fiji?", "suva"],
["Alex: What is the capital city of the country currently ranked first for FIFA men’s football?",
"brasilia"],
["Alex: What is the capital city of the country with the most significant length:width ratio?",
"santiago"],
["Alex: Which African country has 3 capital cities?", "south africa"],
["Alex: What is the capital city of the country that is to the East of the Andes Mountains?",
"buenos aires"],
["Alex: What is the capital city of Texas?", "austin"],
["Alex: What is the capital city of the country located in the very North Atlantic Ocean and is
surrounded by water?", "reykjavik"],
["Alex: Which country has the capital city Montevideo and has won the first ever FIFA World Cup?",
"uruguay"],
["Alex: What is the capital city of the country where IKEA is founded?", "stockholm"],
["Alex: Addis Ababa is the capital city of which African country?", "ethiopia"],
["Alex: What is the capital city of the country with a double-triangle shaped flag?", "kathmandu"],
["Alex: What is the capital city of Croatia?", "zagreb"],
["Alex: Which country has the lowest average elevations in the world and has Male as its capital city?",
"maldives"]
```

5 Function Implementation

Alex currently has several functions:

- Expression Calculator
- Search

- Quiz
- Calculate how many days the user has lived
- *Plane Fight* GAME

For the expression calculator, it's easy to achieve by using *eval()* function in Python. There are basically 3 common errors which might occurs:

```
errorZero = "Error: Division By Zero"
errorName = "Error: Name is not defined"
errorSyn = "Error: Syntax is invalid"
```

The main function:

```
def Calc():
    try:
        results = str(eval(expression.get()))
        ansOfCalc = "= " + results
        label.config(text = ansOfCalc)
    except ZeroDivisionError:
        label.config(text = errorZero)
    except NameError:
        label.config(text = errorName)
    except SyntaxError:
        label.config(text = errorSyn)
```

For the search function, we uses the *webbrowser.open()* function in the webbrowser module to open the URL.

```
def Search():
    def get_input1():
        webbrowser.open("https://bing.com/search?q="+entry1.get())
    def get_input2():
        webbrowser.open("https://baike.baidu.com/item/"+entry1.get())
    entry1 = Entry(win)
    entry1.grid(row = 0,column = 0,ipadx = 100)
    btn1 = Button(win, text="Search on Bing", command = get_input1).grid(row = 1, column = 0)
    btn2 = Button(win, text = "Baidu Encyclopedia", command = get_input2).grid(row = 2, column = 0)
```

In quiz mode, We use JSON to process the databases, which is very important. This part of code is the process of opening a JSON file. Then the database is shuffled to give out a random order of questions.

```
with open('quiz.json') as f:
    obj = json.load(f)
q = (obj['ques'])
options = (obj['options'])
a = (obj['ans'])
z = zip(q, options, a)
l = list(z)
random.shuffle(l)
q,options,a=zip(*l)
```

The main part of this function consists of Input part and Output part(I/O), which is a representation of human-computer interaction. We created a class *Quiz* to store the methods in the main part.

```
class Quiz:
    def __init__(self):
        self.qn = 0
        self.qno = 1
        self.quest = StringVar()
        self.ques = self.question(self.qn)
        self.opt_selected = IntVar()
        self.opts = self.radiobtns()
        self.display_options(self.qn)
        self.buttons()
```

```
            self.correct = 0

        def display_options(self, qn):
            val = 0
            self.opt_selected.set(0)
            self.ques['text'] = q[qn]
            for op in options[qn]:
                  self.opts[val]['text'] = op
                  val += 1

        def checkans(self, qn):
            if self.opt_selected.get() == a[qn]:
                 return True

        def nextbtn(self):
            if self.checkans(self.qn):
                self.correct += 1
            self.qn += 1
            self.qno += 1
            if self.qn == 10:
                self.display_result()
                root.destroy()
            else:
                self.quest.set(str(self.qno)+". "+q[self.qn])
                self.display_options(self.qn)

        def display_result(self):
            score = int(self.correct / 10 * 100)
            result = "Score: " + str(score) + "%"
            wc = 10 - self.correct
            correct = "No. of correct answers: " + str(self.correct)
            wrong = "No. of wrong answers: " + str(wc)
            mb.showinfo("Result", "\n".join([result, correct, wrong]))
```

For the days lived function, we uses the datetime module to calculate the difference between the user's birthday and today. The user's birthday is input by the user using *Entry(frame)*. The core code of this function is *__sub__()*.

```
    def DaysLived():
        frame = Frame(root)
        entryYear = Entry(frame)
        entryYear.insert(0, "Year: (please delete)")
        entryMonth = Entry(frame)
        entryMonth.insert(0, "Month: (please delete)")
        entryDay = Entry(frame)
        entryDay.insert(0, "Day: (please delete)")
        label = Label(frame)
        def CalcDays():
            birthYear = int(entryYear.get())
            birthMonth = int(entryMonth.get())
            birthDay = int(entryDay.get())
            birth = datetime.date(birthYear, birthMonth, birthDay)
            ansOfCalc = "You have lived for " + str(current.__sub__(birth).days) + " days"
            label.config(text = ansOfCalc)
        Btn = Button(root, text ="Calculate", command = CalcDays)
        keyboard.add_hotkey("enter", lambda: CalcDays())
        currentYear = int(datetime.datetime.now().strftime('%Y'))
        currentMonth = int(datetime.datetime.now().strftime('%m'))
        currentDay = int(datetime.datetime.now().strftime('%d'))
        current = datetime.date(currentYear,currentMonth,currentDay)
```

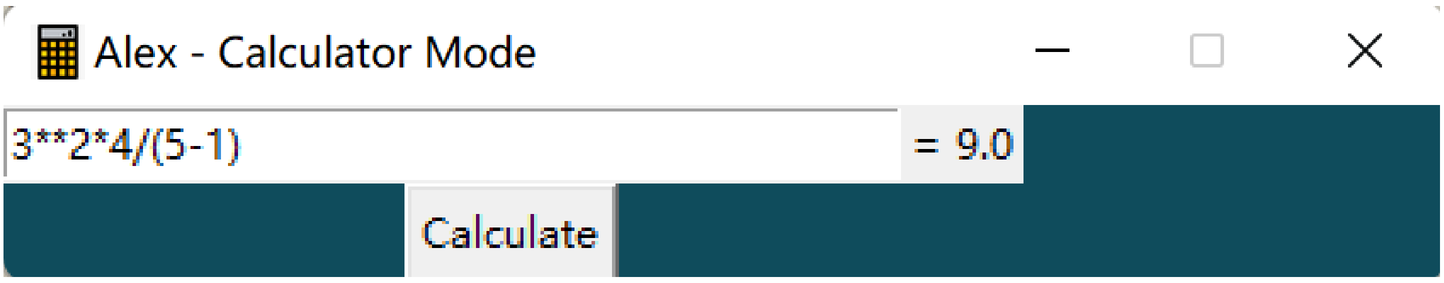
To implement the Plane Fight game, *pygame* and *settings* module is used. The main part of the program is shown.

```
while running:
    screen.fill(0)
    screen.blit(background_img,(0,0))
    score_text = score_font.render("score: %s" %str(score), True, color_white)
    screen.blit(score_text, (10, 5))
    if level == 1 and score >3000:
        level = 2
        level_up_sound.play()
        add_small_enemies(small_enemies, enemies, 2)
        add_mid_enemies(mid_enemies, enemies, 2)
        add_big_enemies(big_enemies, enemies, 1)
        inc_speed(small_enemies, 1)
    elif level ==  2 and score > 20000:
        level = 3
        level_up_sound.play()
        add_small_enemies(small_enemies, enemies, 3)
        add_mid_enemies(mid_enemies, enemies, 3)
        add_big_enemies(big_enemies, enemies, 1)
        inc_speed(small_enemies, 1)
        inc_speed(mid_enemies, 1)
    elif level ==  3 and score > 80000:
        level = 4
        level_up_sound.play()
        add_small_enemies(small_enemies, enemies, 4)
        add_mid_enemies(mid_enemies, enemies, 3)
        add_big_enemies(big_enemies, enemies, 1)
        inc_speed(small_enemies, 1)
        inc_speed(mid_enemies, 1)
        inc_speed(big_enemies, 1)
    elif level ==  4 and score > 200000:
        level = 5
        level_up_sound.play()
        add_small_enemies(small_enemies, enemies, 6)
        add_mid_enemies(mid_enemies, enemies, 4)
        add_big_enemies(big_enemies, enemies, 2)
        inc_speed(small_enemies, 1)
        inc_speed(mid_enemies, 1)
        inc_speed(big_enemies, 1)
    level_font = pygame.font.SysFont("arial", 16)
    level_text = level_font.render("level: %s" %str(level), True, color_black)
    screen.blit(level_text,(10,35))
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()
        elif event.type == MOUSEBUTTONDOWN:
            button_down_sound.play()
            if event.button == 1 and paused_rect.collidepoint(event.pos):

                paused = not paused
                if paused:
                    paused_img = resume_pressed_image
                    pygame.time.set_timer(supply_timer,0)
                    pygame.mixer.music.pause()
                    pygame.mixer.pause()
                else:
                    paused_img = pause_pressed_image
                    pygame.time.set_timer(supply_timer, 30 * 1000)
                    pygame.mixer.music.unpause()
                    pygame.mixer.unpause()
            if life_num == 0:
                if event.button == 1 and back_to_game_rect.collidepoint(event.pos):
                    gameBeigin(running)
        elif event.type == MOUSEMOTION:
            if paused_rect.collidepoint(event.pos):
                if paused:
                    paused_img = resume_pressed_image
                else:
                    paused_img = pause_pressed_image
            else:
                if paused:
                    paused_img = resume_nor_image
                else:
                    paused_img = pause_nor_image
        elif event.type == KEYDOWN:
            if event.key == K_ESCAPE:
                pygame.quit()
                sys.exit()
            if event.key == K_RETURN:
                gameBeigin(running)
            if event.key == K_SPACE:
                if bomb_num:
                    bomb_num -= 1
                    bomb_sound.play()
                    for each in enemies:
                        if each.rect.bottom > 0:
                            each.active = False
        elif event.type == supply_timer:
            if random.choice([True, False]):
                bomb_supply.reset()
            else:
                bullet_supply.reset()
        elif event.type == double_bullet_timer:
            is_double_bullet = False
            pygame.time.set_timer(double_bullet_timer, 0)
        elif event.type == invincible_time:
            player.invincible = False
            pygame.time.set_timer(invincible_time, 0)
    screen.blit(paused_img, paused_rect)
    if life_num and (not paused):
        bomb_text = bomb_font.render("x %d" % bomb_num, True, color_black)
        bomb_text_rect = bomb_text.get_rect()
        screen.blit(bomb_img, (10, screenheight - 5 - bomb_rect.height))
        screen.blit(bomb_text,(20 + bomb_rect.width, screenheight - 5 - bomb_text_rect.height))
        if life_num:
            for i in range(life_num):
                screen.blit(life_img, (screenwidth - i - 10 - (i+1)*life_rect.width, screenheight
- 5 - life_rect.height))
        key_pressed = pygame.key.get_pressed()
        if key_pressed[K_w] or key_pressed[K_UP]:
            player.moveUp()
        if key_pressed[K_s] or key_pressed[K_DOWN]:
            player.moveDown()
        if key_pressed[K_a] or key_pressed[K_LEFT]:
            player.moveLeft()
        if key_pressed[K_d] or key_pressed[K_RIGHT]:
            player.moveRight()
        if not (delay % 6):
            bullet_sound.play()
            if not is_double_bullet:
                bullets = bullet1
                bullets[bullet1_index].reset(player.rect.midtop)
                bullet1_index = (bullet1_index + 1) % bullet1_num
            else:
                bullets = bullet2
                bullets[bullet2_index].reset((player.rect.centerx - 33, player.rect.centery))
                bullets[bullet2_index+1].reset((player.rect.centerx + 30, player.rect.centery))
                bullets[bullet2_index+2].reset((player.rect.centerx, player.rect.top))
                bullet2_index = (bullet2_index + 3) % bullet2_num

        if bomb_supply.active:
            bomb_supply.move()
            screen.blit(bomb_supply.image, bomb_supply.rect)
            if pygame.sprite.collide_mask(bomb_supply, player):
                get_bomb_sound.play()
                if bomb_num < 4:
                    bomb_num += 1
                bomb_supply.active = False

        if bullet_supply.active:
            bullet_supply.move()
            screen.blit(bullet_supply.image, bullet_supply.rect)
            if pygame.sprite.collide_mask(bullet_supply, player):
                get_bullet_sound.play()
                is_double_bullet = True
                pygame.time.set_timer(double_bullet_timer, 18 * 1000)
                bullet_supply.active = False

        for b in bullets:
            if b.active:
                b.move()
                screen.blit(b.image, b.rect)
                enemies_hit = pygame.sprite.spritecollide(b, enemies,False,pygame.sprite.collide_mask)
                if enemies_hit:
                    b.active = False
                    for e in enemies_hit:
                        if e in big_enemies or e in mid_enemies:
                            e.energy -= 1
                            e.hit = True
                            if e.energy == 0:
                                e.active = False
                        else:
                            e.active = False
        enemies_down = pygame.sprite.spritecollide(player, enemies, False, pygame.sprite.collide_mask)
        if enemies_down and not player.invincible:
            player.active = False
            for e in enemies_down:
                e.active = False
        if delay == 0:
            delay = 60
        delay -= 1
        if not delay % 3:
            switch_img = not switch_img
        if player.active:
            if switch_img:
                screen.blit(player.image1, player.rect)
            else:
                screen.blit(player.image2, player.rect)
        else:
            if not (delay % 3):
                screen.blit(player.destroy_images[player_destroy_index], player.rect)
                player_destroy_index = (player_destroy_index + 1) % 4
                if player_destroy_index == 0:
                    me_killed_sound.play()
                    life_num -= 1
                    player.reset()
                    pygame.time.set_timer(invincible_time, 3*1000)
            for each in big_enemies:
                if each.active:
                    each.move()
                    if not each.hit:
                        if switch_img:
                            screen.blit(each.image1, each.rect)
                        else:
                            screen.blit(each.image2, each.rect)
                            if each.rect.bottom == -50:
                                big_enemy_flying_sound.play(-1)
                    else:
                        screen.blit(each.image_hit, each.rect)
                        each.hit = False
                    pygame.draw.line(screen,color_black,(each.rect.left,each.rect.top-5)
,(each.rect.right, each.rect.top-5),4)
                    energy_remain = each.energy/BigEnemy.energy
                    if energy_remain > 0.4:
                        energy_color = color_green
                    else:
                        energy_color = color_red
                    pygame.draw.line(screen,energy_color,(each.rect.left, each.rect.top - 5)
,(each.rect.left + each.rect.width * energy_remain, each.rect.top - 5),4)
                    if each.rect.bottom == 0:
                        big_enemy_flying_sound.play(-1)
                else:
                    big_enemy_flying_sound.stop()
                    if e3_destroy_index == 0:
                        enemy3_killed_sound.play()
                    if not (delay % 3):
                        screen.blit(each.destroy_images[e3_destroy_index],each.rect)
                        e3_destroy_index = (e3_destroy_index + 1) % 6
                        if e3_destroy_index == 0:
                            score += 3000
                            each.reset()
            for each in mid_enemies:
                if each.active:
                    each.move()
                    if not each.hit:
                        screen.blit(each.image, each.rect)
                    else:
                        screen.blit(each.image_hit, each.rect)
                        each.hit = False
                    pygame.draw.line(screen,color_black,(each.rect.left,each.rect.top - 5)
,(each.rect.right,each.rect.top-5),2)
                    energy_remain = each.energy/MidEnemy.energy
                    if energy_remain > 0.2:
                        energy_color = color_green
                    else:
                        energy_color = color_red
                    pygame.draw.line(screen,energy_color,(each.rect.left, each.rect.top - 5)
, (each.rect.left + each.rect.width * energy_remain, each.rect.top - 5), 2)

                else:
                    if e2_destroy_index == 0:
                        enemy2_killed_sound.play()
                    if not (delay % 3):
                        screen.blit(each.destroy_images[e2_destroy_index],each.rect)
                        e2_destroy_index = (e2_destroy_index + 1) % 4
                        if e2_destroy_index == 0:
                            score += 1000
                            each.reset()
            for each in small_enemies:
                if each.active:
                    each.move()
                    screen.blit(each.image, each.rect)
                else:
                    if e1_destroy_index == 0:
                        enemy1_killed_sound.play()
                    if not (delay % 3):
                        screen.blit(each.destroy_images[e1_destroy_index],each.rect)
                        e1_destroy_index = (e1_destroy_index + 1) % 4
                        if e1_destroy_index == 0:
                            score += 200
                            each.reset()
        elif life_num == 0:
            screen.blit(gameover_img, gameover_rect)
            pygame.mixer.music.stop()
            pygame.mixer.stop()
            pygame.time.set_timer(supply_timer, 0)
            if not flag_recorded:
                flag_recorded = True
                with open("score_record.txt", "r") as f:
                    record_score = int(f.read())
                if score > record_score:
                    with open("score_record.txt", "w") as f:
                        f.write(str(score))
            record_score_text = score_font.render("%d" % record_score, True, color_black)
            screen.blit(record_score_text,(150, 29))
            game_over_score_text = score_font.render("%d" % score, True, color_black)
            screen.blit(game_over_score_text,(200,305))
            screen.blit(back_to_game_image,(90,500))
            pygame.display.update()
            clock.tick(60)
```

6 Document

We have made a brief website as the document of Anoint. It also contains SPARK introduction and IGCSE Revision Guide Websites for user to study. We uses HTML and CSS to construct this *.html* document with the help of a CSS Framework *Bootstrap*.We have combined several learning website for the users so that they are able to go directly in those websites in the *Anoint* Document.

7 Output Results



```
3**2*4/(5-1)    = 9.0
```

Figure 2: Calculator Mode

This is Calculator Mode. User can enter a equation and can either click the “Calculate” button or press “Enter” on their keyboard to get the calculation result. Alex is able to make calculations involving addition, subtraction, multiplication, division and exponential. Yet it is not advanced enough to deal with factorial and square root.

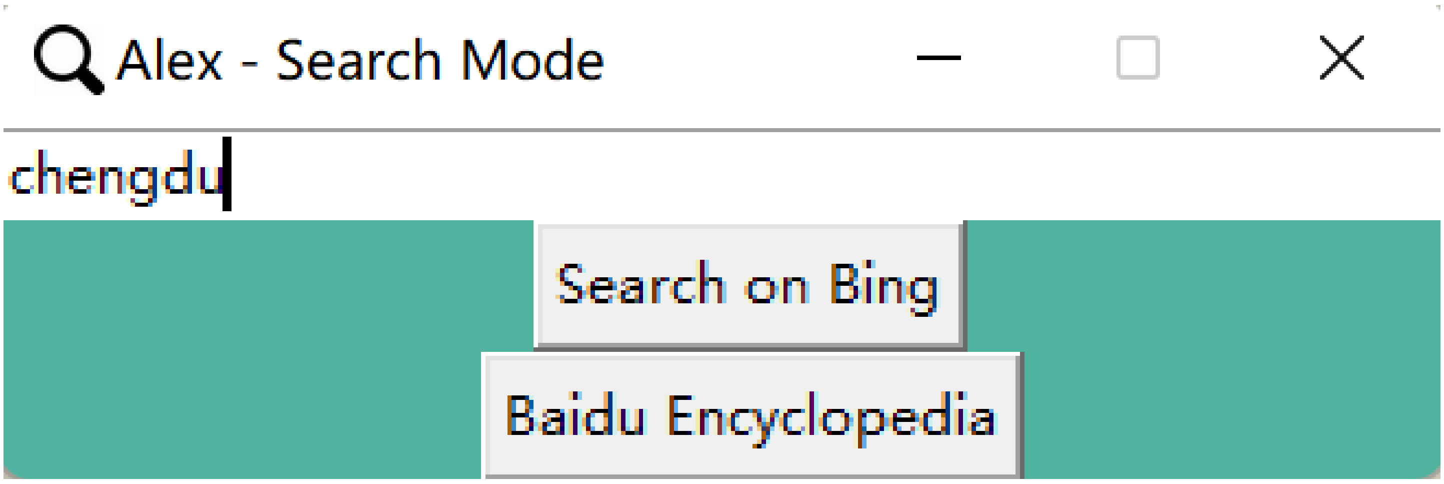
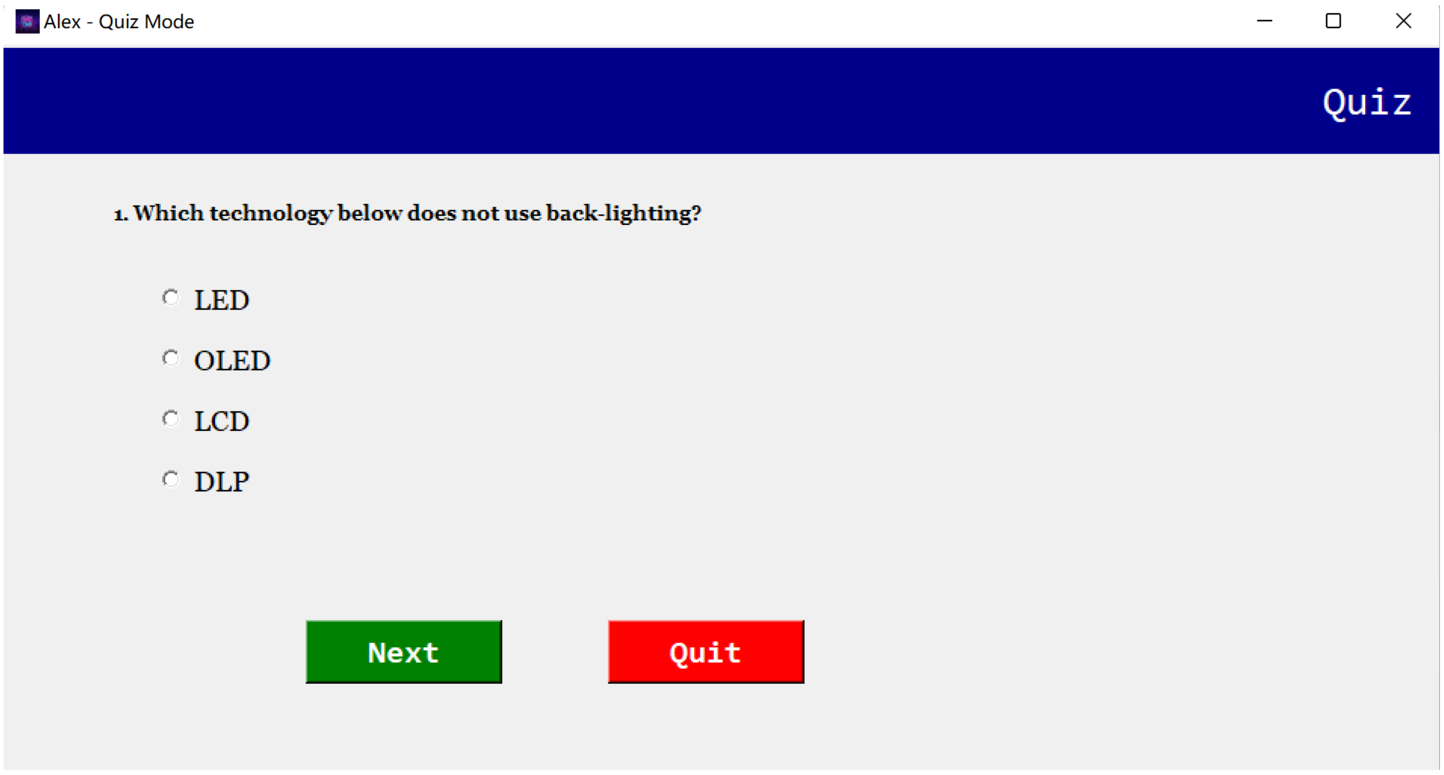


Figure 3: Search Mode

This is the Search Mode. User can enter something they want to search in the entry bar, and click one of the two buttons below: *Search on Bing* (“Enter”) and *Baidu Encyclopedia* (“Shift”). They will be redirected to a webpage with resources they need.

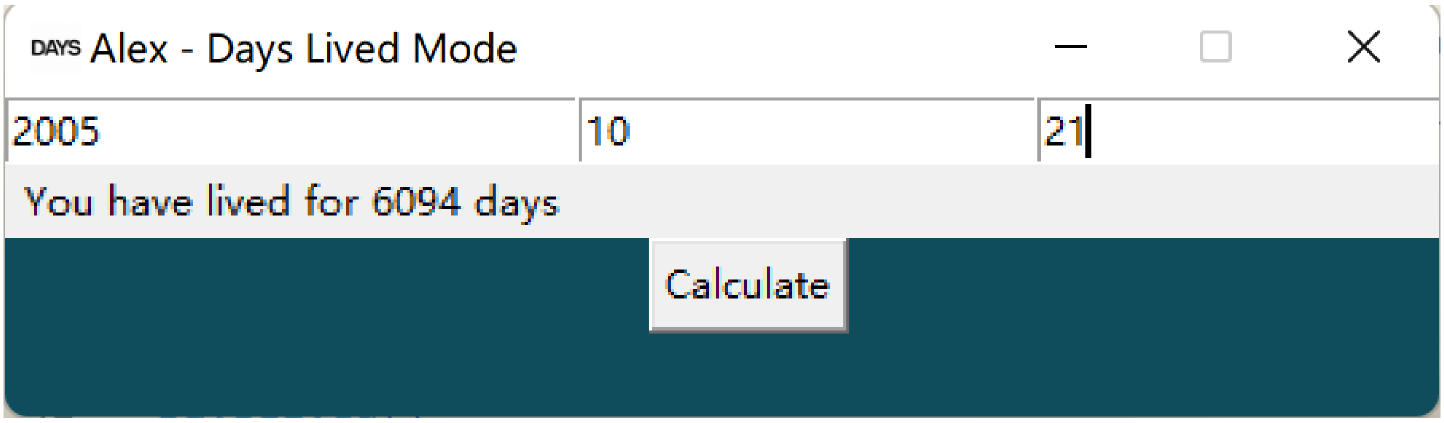


1. Which technology below does not use back-lighting?

- LED
- OLED
- LCD
- DLP

Figure 4: Quiz Mode

This is Quiz Mode. User firstly selects the topic that the quiz is about, then 10 questions about that specific topic will be displayed by Alex. User can click the answer they think it’s right, and at the end Alex will tell the user how many they got right.



You have lived for 6094 days

Figure 5: Days Lived Mode

This is Alex under Days Lived Mode. User can enter their birthday in the three entry boxes at the top, then they can either click the “Calculate” button or press “Enter” on their keyboard to get the calculation result.

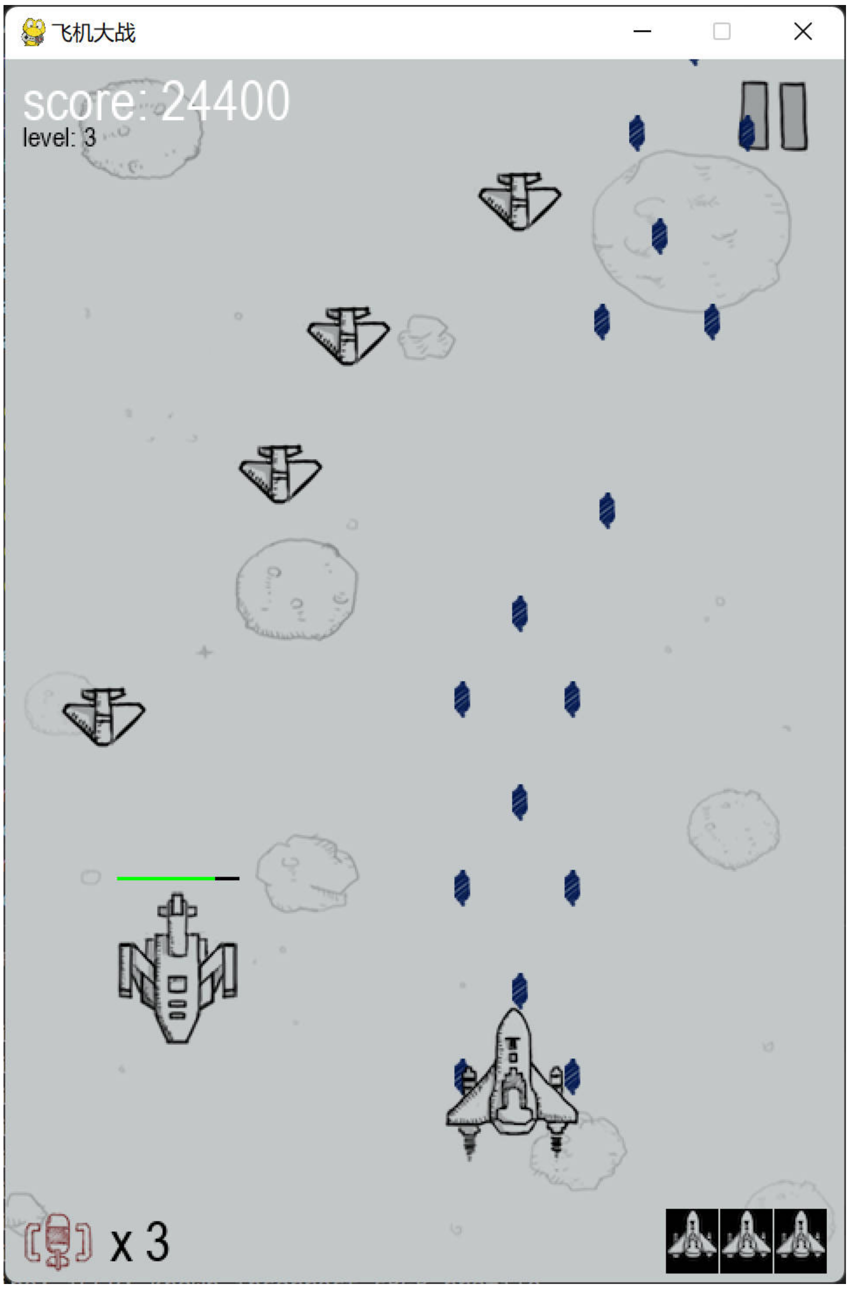


Figure 6: Plane Fight GAME Mode

This is the fifth function - Plane Fight GAME. This game is created to help the user to relax after hard working and learning.

8 Conclusion

As chatbots become an essential part in software technology, it is important to understand how they work and their values to society. By building a chatbot ourselves, we have learnt and understood more about chatbots and its community. Future programming should focus on Natural Language Processing (NLP) and Natural Language Toolkit (NLTK) to make the chatbot more advanced and more human-computer interactive. Voice chat is another field that makes chatbots easier to communicate with. Overall, we have acquired many new skills in programming, and have produced our best work.

9 References

- The Value Of Chatbots For Today's Consumers. (2018). Retrieved from: `www.forbes.com/sites/forbescommunicationscouncil/2018/02/13/the-value-of-chatbots-for-todays-consumers`