Jie Ji

Math 480 Final Project

Math 480a Final Project:
Python Game: The Dungeon

My project is to create a text based game using Python. It is a single player game and contains three simple puzzles and one warm up question, which is a homework question from the course. In order to win the dungeon game, player must answer three problems correctly. First puzzle is a simple guessing game. It involves some kind of bisection method by the player. Second puzzle involves in matrix multiplication. It asks the user's input for the sum of the elements in the product of two fixed matrix. The last puzzle "the dragon realm" is based on luck. It let the user to choose an integer between 1 and 2, which represents two caves. Then the computer randomly decides which is the "good" one and then the other is the "bad" obviously. The good cave will let the player win the game and the bad cave will cost the character's life in the game.

The idea of the game originally comes from previous projects. There are a calculator module and a Sudoku game in last year's projects. The Sudoku project obviously gives me the idea to write a game using the python and the calculator module encourages me with the method it deals with user's input. Then I checked online for some instruction to write a python game, I found a book [1] that is so useful that immediately I settled my final project for creating a game. Also the classic game *MUD* (Multi-Users Dungeon) gives me the idea to write a dungeon game.

The first thing I did with the project is to decide the puzzles for the game. Since this is a math class, so I want something that at least deals with numbers. Therefore I come up with the idea of the guessing game and the matrix multiplication. Since there is no direct algorithm for matrix in Python, I used the Numpy module, which is fascinating dealing with numerical analysis. Additionally, in order to add some undeterminable variable to the game, I made the "dragon realm" puzzles.

My initial plan for the game is that it should be with graphics and animations. However, since I am really bad at drawing, I had to change up the plan by replacing the graphics and animations with simple text. Moreover, in order to make it more "game-like", I added background music to it[2], which is accomplished by the pygame module[3]. It is a really fascinating "add-ons" for python since it enables programmers to use lots of elements like graphics, sound, animations and videos in their games.

One thing I noticed that when I just double click to open the project, Python seems to eat up some words in my print() command, the it display the message for wrong answer even if I input the right answer. However, I was able to run the program perfectly in IDEL(Python GUI), which is a code editor for Python. I tried to look for solutions online, but unfortunately, nothing seems close to my situation. My best guess is that pygame module is not officially released for 64-bit windows operating system, therefore it could just be bugs for pygame. I already reported it to the the developer of pygame module.

The most beneficial thing I learned in this project is that open source software is really powerful! Can you imagine using Matlab to write a computer game? Definitely no way! Since Python (also SAGE) is open source, so anybody who wants to do some adjustments, or say add some modules to the language could easily do their things, without worries of copyrights, license, etc. With proper management and enough sponsor, open source software could really do something serious and fascinating

BTW: not really related to the project AND the course, but just something interesting about Python. A friend of mine told me that downloading Python is forbidden by the internet censoring system (A.K.A the Great Firewall, GFW) in China. That is why on the Python official website, we could see two labels on the left, one of them is “download” in English and the other is “download” in Chinese.

Here is the code of the project with comments and explanations:

```
import pygame, sys, time, random, numpy as np
from pygame.locals import *
pygame.init() #This is the command to initial all the function/command in pygame module
              #Python would give an error message without this line

#set up background music of the game
pygame.mixer.music.load('wind.wav')
pygame.mixer.music.play(-1, 0.0) #"-1" let Python know that I only play the soundtrack once
                                 # (0,0) is when I want the music to start
musicPlaying = True
WINDOWWIDTH = 1
WINDOWHEIGHT = 1
```

```
#This is where Python open up a small window to let the background music play when I open it.
windowSurface = pygame.display.set_mode((WINDOWWIDTH, WINDOWHEIGHT), 0, 32)
pygame.display.set_caption('Animation')

#take the player's name
print('What is your name, player?')
playerName = input()

#display the intro of the game and take the player's input as the player's name
def displayIntro():
    print()
    print()
    print('Once upon a time,' + playerName +', a ancient hero,')
    print('who is trapped in the dungeon by the dark force.')
    print('Two thousand years after his capture in the dungeon,')
    print('The outside world is facing great danger.')
    print('...')
    time.sleep(1)
    print('...')
    time.sleep(2)
    print('It is now the beginning of the journey of our hero,' + playerName)
    print('to save the world again, from the dark force.')
    time.sleep(3)
    print('...')
    time.sleep(2)
    print('Well, only if our hero,' + playerName +' could get out of the dungeon first,')
    print ('..........')
    time.sleep(3)
    print ('Old sayings that only the one who could complete three puzzles of numbers')
    time.sleep(1)
    print ('would be able to escape from the dungeon')
    print ('so ' + playerName + ',')
    time.sleep(2)
    print('are you ready?')
################################################################################
#first mission of the game, the guessing game, which let the player
#to guess the randomly generated integer from 1 to 20 in 8 shots.(pretty easy one)
def Guess():
    print(' ')
    print(' ')
    print('Here we are, at a ancient stone wall, you have reached a dead end')
    time.sleep(2)
    print(playerName + ', only by completing this quest will buy you the pass')
    time.sleep(2)
    print('...')
    print('to the way out.')
    time.sleep(2)

    guessTaken = 0
    number = random.randint(1,20)
    print('')
    print('')
    print('Well, ' + playerName +', you have to figure out a mystery number between 1 and 20')
    while guessTaken < 8:
```

```
    print('Take a guess.')
    guess = input()
    guess = int(guess)

    guessTaken = guessTaken + 1

    if guess < number:
        print('Too low.')

    if guess > number:
        print('Too high.')

    if guess == number:
        break

if guess == number:
    guessTaken = str(guessTaken)
    print('You got it right!! ' + playerName + 'You just get the pass')
    time.sleep(1)
    print('to the next level. Good Job!! You are close to the exit')

if guess != number:
    number = str(number)
    print('Sorry, '+playerName + 'you failed this quest')
    print('Do you want to play again?')
    playAgain = input()
    if playAgain == 'yes' or playAgain == 'y':
            Guess()
        else:
            sys.exit()
#end of the guessing game
################################################################################
################################################################################
#second level of the dungeon which is a simple matrix multiplication game
#player have to input the right answer to move on
def Multi():
    A = np.matrix('3 6 1; 5 2 9; 4 5 7')
    B = np.matrix('5;7;8')
    print(' ')
    print()
    print('Now you have reached wall, the writing on it is so fuzzy')
    time.sleep(1)
    print('but, still, you can see it is like:')
    print(' ')
    time.sleep(1)
    print('oh, it is:')
    print(A);
    time.sleep(1)
    print('oh, there is more!')
    time.sleep(1)
    print(B)
    time.sleep(1)
    print('obviously, it is puzzle, waiting for you to solve')
    print('hint: sum of the elements in A*B')
    print('you only have one shot')


    ans = input()
    if ans == '287':
        print('You Got it right!! Now move up to the final level!!')
    else:
        print('Sorry, '+playerName + 'you failed this quest')
        print('Do you want to play again?')
        playAgain2 = input()
        if playAgain2 == 'yes' or playAgain2 == 'y':
            Multi()
        else:
            sys.exit()
#end of the multiplication
#################################################################################
#################################################################################
#the final stage of the dungeon game,which is based on the luck of the player
#choosing between two cave. Randomly one is the cave to win the game
#and another kills the player. The player could choose to replay the game or quit.

#intro of the dragon game
def dragon():
    print()
    print()
    print('Now you have come to the dragon realms. Dragons are the wardens of the dungeon')
    time.sleep(1)
    print('In front of you are two caves')
    time.sleep(1)
    print('one of them would point you the way out of the dungeon')
    time.sleep(2)
    print('the other will eat you on sight.')
    time.sleep(1)
    print('so, be careful, ' + playerName, 'it is your choice...')
    print()
#let the player choose a cave between 1 and two
def chooseCave():
    cave = ''
    while cave != '1' and cave != '2':
        print('Which cave will you go into? (1 or 2)')
        cave = input()
    return cave

#determine whether the cave chosen is good or bad
def checkCave (chosenCave):
    print('You approach the cave...')
    time.sleep(2)
    print('It is dark and spooky...')
    time.sleep(2)
    print('A large dragon suddenly appears in front of you! He opens his jaw and...')
    print()
    time.sleep(2)

    GoodCave = random.randint(1,2)
    if chosenCave == str(GoodCave):
        print('points you the way out of dungeon.')
        time.sleep(2)
        print('Congratulations! ' + playerName +', time for you to save the world out there!')
    else:
        print('gobbles you down in one bite!!' )
#end of the dragon game
################################################################################
#warm up for the game, it is a previous homework question
def warmup():
    print()
    print()
    print('first of all, a little bit warm up')
    time.sleep(1)
    print('What is the least composite number that has all')
    print('of the first 5 prime numbers as factors?')
    answer = input()
    if answer == '2310':
        print('right, now you are ready to go !!')
    else:
        print('well, the answer should be 2310. Concentration!!! ' + playerName)

#Execution the function defined above.
playAgain = 'yes'
while playAgain == 'yes' or playAgain == 'y':
    displayIntro()
    time.sleep(1)
```

```
warmup()
time.sleep(1)
Guess()
time.sleep(1)
Multi()
time.sleep(1)
dragon()
caveNumber = chooseCave()
checkCave(caveNumber)
print('Do you want to play again? (yes or no)')
playAgain = input()
```