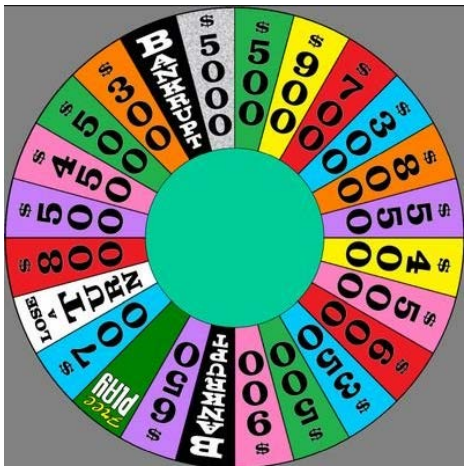
MA1008 Introduction to Computational Thinking
Mini Project, Weeks 10 – 13
Submission Deadline: 23:59, Friday 17 April 2020

Introduction

You are to write a program to play the game "Wheel of Fortune". If you are not familiar with the game, please watch this YouTube video: https://www.youtube.com/watch?v=OeX1rJ9Abr0. The game you are to get your program to play is contained in the time segment starting from 3 minutes 16 seconds into the video up till 5 minutes. Basically, it contains a puzzle which is a phrase containing a number of words, which are made up of letters. At the start, all the letters are set to blank. The players are to fill in the blanks by guessing one letter at a time, until a player could complete the phrase correctly. Before guessing, a player needs to spin a wheel, shown in the figure below.



The wheel is divided into 24 sectors. Each sector contains either

1. a number, or
2. the word "Bankrupt", or
3. the phrase "Lose a turn" or
4. the phrase "Free play".

Rules and Actions of the Game

A player spins the wheel which will stop at a sector. Here are the rules and actions upon a spin:

1. If the spin stops at a number, the player is to either call out a consonant or buy a vowel at $250, within ten seconds.
    a. If the player calls out a consonant,
        i. If the consonant exists in the phrase, the player is awarded the dollar value of the number multiplied by the number of occurrences of the consonant in the phrase, and the consonant is filled into the phrase. The player is then allowed one of three actions:
            a. Spin the wheel again, which means we return to Step 1.
            b. Buy a vowel by calling out the vowel desired. One of three things can happen:
                i. Each vowel costs $250 and is to be deducted from the player's total so far. If the player's total is below $250, the player cannot buy and therefore must spin again.
                ii. The vowel exists, and is filled into the appropriate blank(s) in the puzzle. $250 is deducted from the player's total regardless of the number of occurrences of the vowel. The player then can buy another vowel (return to Step 1.a.i.b) or spin the wheel again (return to Step 1).
                iii. The vowel doesn't exist or has already appeared. The player loses the turn. $250 is deducted.
            c. Solve the puzzle.
        ii. If the letter does not exist, the player loses the turn.
    b. If the player buys a vowel, then the play moves to 1.a.i.b above.
2. If the spin stops at "Bankrupt", the player loses all the money won so far and loses the turn.
3. If the spin stops on "Lose a turn", the player loses the turn but keeps the money won so far.

4. If the spin lands on “Free play”, the player is to call out a letter, either consonant or vowel.
   a. If the letter exists, it is filled into the phrase. No money is added. Then the player returns to spinning the wheel, buying a vowel or solving the puzzle.
   b. If the letter doesn’t exist, the player proceeds to spinning the wheel, buying a vowel or solving the puzzle without losing a turn.
5. Upon successfully filling in a letter, the player is offered one chance only to complete the phrase within 10 seconds.
   a. If this is successful, the player keeps the money earned so far, and the game ends. All the other players get nothing.
   b. If this is not successful, then the player loses the turn.
6. Once the current player loses the turn, the next player spins the wheel, and we return to Step 1 for this new player.

A flow chart of the game is given at the end of this handout.

What you need to do

1. Write a program to play the game according to the rules above with three players. You need to have a means to identify the players.
2. You will need to provide a display with the following contents:
   a. A picture of the wheel with the sectors, each marked correctly with a number or phrase.
   b. A means of simulating spinning and stopping the wheel. You need to be able to control how long the wheel spins, with a certain degree of random variation.
   c. A means for indicating which sector a spin stops at.
   d. At the start, display blanks that indicate the letters to be filled in.
   e. As the game progresses, fill in the letters in the right position as a player calls them out.
   f. Display the score of each player as the game progresses.
3. When a player successfully completes the phrase, award the winning amount to the player and terminate the game.

Resources

Here are two resources you need to use:

1. The phrases that you can use in the program are stored in a text file called **WofFPhrases.txt**, provided together with this handout in the Mini Project link in the Assignments folder. Each phrase is contained in one line in the file. Your program should automatically and randomly select a phrase from the file and offer that as the puzzle to be solved. So if the phrase selected is “Wheel of Fortune”, then your display should show a pattern of boxes such as this:

   □□□□□ □□ □□□□□□□

   As the game progresses, the boxes are filled in with letters until a player guesses the complete phrase correctly. You may add more phrases to the file if you wish.

2. The displays will need to be done with graphics. For the graphics, you need to use the graphics library provided with Python, called **Turtle**. Turtle capabilities and functions are described fully in Section 24.1 of the Python documentation, which you can access via the IDLE interactive shell by clicking Help > Python Docs and then search for “Turtle”. Turtle provides the fundamental functions required for drawing, such as plotting points, lines and shapes, and controlling their attributes such as location, colour, line thickness, and more. It can display text also.

   You should study the different Turtle functions that are required for displaying your graphics entities. There are many Turtle functions, you only need to learn the ones you require. It is therefore necessary that you work out what you need beforehand.

   There is only one set of rules for playing the game, as given above. But there are different levels of sophistication in your display board. This is up to you to design. One basic requirement is that the display must be clear, so that the user can see exactly the states of play in the game.

What you need to submit

1. The Python source file of your program. Remember to put your name and group number as a comment in the first line of your code.
2. A Word file providing instructions on how to run your program and play the game. The file also should contain a section highlighting any special features you wish your tutor to take note of, such as how you organise the data, specific algorithms central to the working of your program, and features in your displays and the user interface. Keep this file short. Excluding diagrams, we expect a file of no more than two or three pages. Again, make sure that you have your name at the top of this Word file.

When to submit

The deadline for the submission is at **17:00 on Friday, 24 April 2020**.

How to submit

The submission is to be made online in the course site under the Mini Project link in the Assignment folder, the same place where you fetch the project handout. It should be clear how to submit when you are there. You must name your file with your surname and initial. So, if you surname is Ang and your initials are B. C., your filename should be AngBC.py.

What we look for in the grading

We will be looking for these things in the grading:

1. The quality of your algorithm, i.e. how well your algorithm works and whether it takes care of different possibilities that might occur.
2. The quality of your code. Given that this is a take home project with a long time span, we expect your code to run. Any code that does not run will suffer a large grade reduction. Also, remember that your code needs to be easily readable and understandable, so use meaningful names for the variables and add comments appropriately. You should handle any possible errors (such as in user inputs) gracefully.
3. The quality of your outputs. Your display should be clear and neat and it should be intuitive for a player to comprehend it and play the game. While we do not insist on a pretty display (as the judgement on that can be subjective), untidy ones will be penalised.

Epilogue

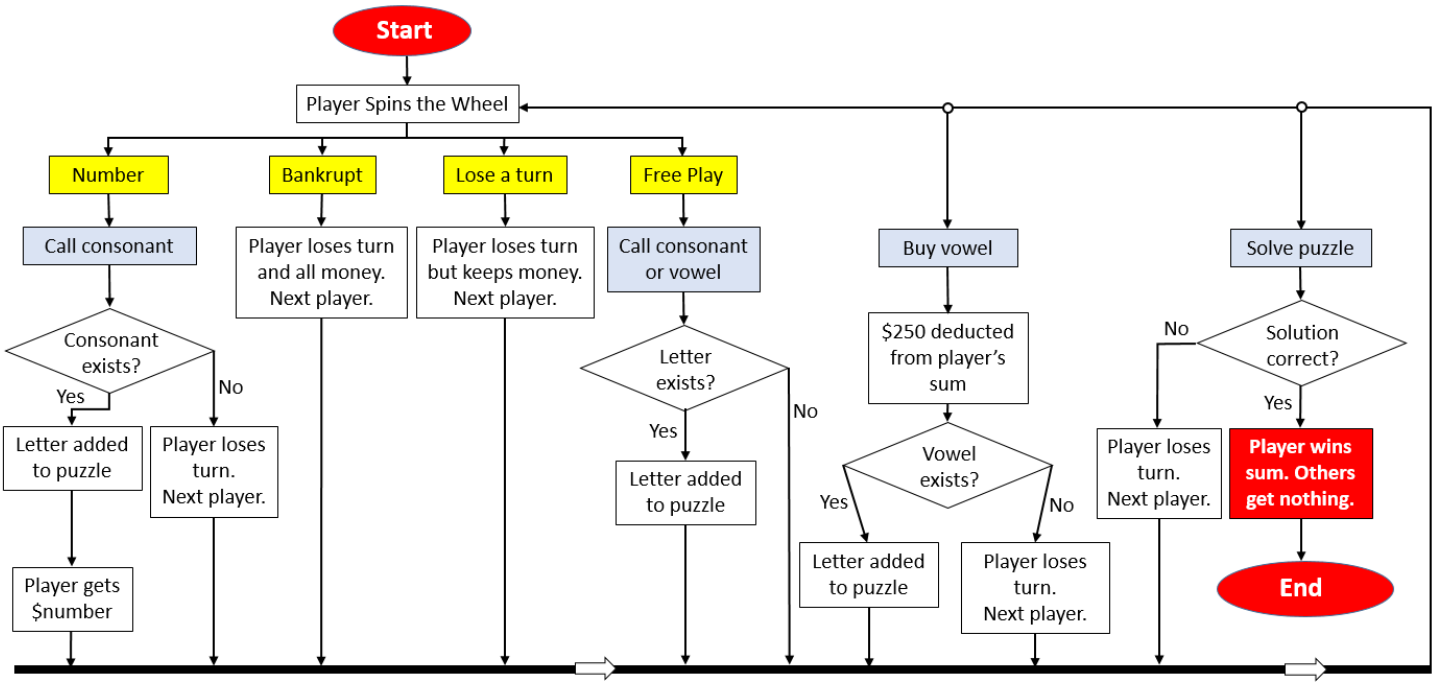
This is a take-home project for which you are given four long weeks. On average, if you spend 3 - 6 hours per week on the project, it is about right.

Start right away. Do not wait till the last few days; you will not have enough time, because there are new things to learn beyond what has been covered in class. Develop the algorithm first, step by step, then the code. Work out what you need to do, and what facilities (i.e. functions) you need. You have to study the facilities Turtle provides; build on them and exploit them. But you don't need to learn everything about Turtle because you don't need them all.

We know that there are Wheel of Fortune programs on the Internet, and we know how they work. If you model your program on anyone of them, you are required to cite the source in your report and explain what additional "value" you have added to your program. Otherwise, you will be deemed to have plagiarised and will fail the project.

You are welcome to consult and seek clarifications from your tutor, refer to books and the Internet, and even discuss with your peers. After all, this is how we all learn. But the construction of the program must be your very own. Any programs deemed to be copies of each other will be given a fail grade, regardless of who does the copying. We expect and trust that you will do the work honestly. Upon your honour.

Flow chart of the Game



Notes:

1. Your program starts at the "Start" box, and its flow should follow the directions of the arrows.
2. Red boxes are for the start and the end. Yellow boxes are for the outcomes of a spin. Grey boxes are for decisions by the player.
3. Note the box which says "Next player". It means that there is a change to a new player.