UFCFXK-30-3: Digital Systems Project

Student name: Joshua Hore

Project Title: Python Trivia Game

Abstract:

My planned project is a game that teaches people how to code in Python by choosing and outputting questions from a set. Players will have to see how high a score they can get within the time limit.

Aims and objectives:

I wish to design a functional game that makes use of an algorithm to make learning a subject that I find difficult easier for future learners.

e.g. The project objectives are:

- To create an interactive game that makes learning Python easier and more fun.
- To create a game with a working timer and point system to encourage players to gain higher scores
- To design and make use of an algorithm to generate questions
- To design an interface that's easy to understand and has an appealing design.

Research:

Gamification is defined as being when game-like. Examples of these include mechanics, such as rules rewards and levelling. And dynamics, such as co-operation and competition, which relate to emotion. The reason why companies may use gamification include increasing engagement and making tool usage easier. This is similar to why I have chosen to create this game.

Sources that I have looked at have given real-world examples of Gamification. These include Nike+, which set challenges, rewards for completing them, and leaderboards to make workouts more fun and motivate people to do better. Another example would be Google Forms,, which used processes such as creating reward badges or interactive submission systems to benefit not just the user, but entire groups of employees or students. One final example would be HP's Utility cup, which allowed people to compete based on their knowledge of cyber security. This shows that gamification has been used in ways that relate to computing knowledge before.

I tried looking up some tips to try to increase the appeal of my trivia game. Some of the tips that I found included making sure that the correct answer is not always in the same place, so that no-one thinks that they can win just by constantly pressing one button and make wrong answers plausible and keep questions relevant and not to difficult, but there could still be a few questions that really try to get people to think. I'm not sure how many of these tips I will add to my own work, but they have at least given me ideas from people who have had ideas similar to my own.

Key requirements:

Functional Requirements

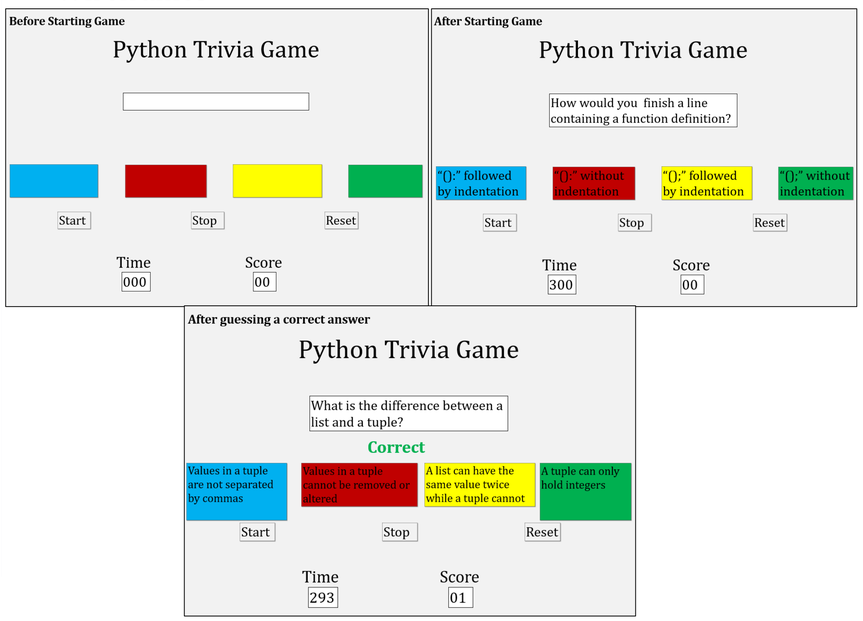
1. It MUST randomly select a question to do with Python and output it to the user alongside the matching answers.
2. It MUST award a point to the user if they get a correct answer and give them nothing if they get it wrong.
3. It MUST reset the score each time that a request is made to start the game from the beginning.
4. It MUST have at least one "Start" button which prevents the program from selecting a question or allowing the answer buttons to become usable until it is pressed, at which point, the game will begin.
5. It MUST have a "Reset" button that the user to start the game from the beginning regardless of if they're currently running the game or not.
6. It MUST have a "Stop" button that stops the timer and prevent any answers from being chosen until "Start" is pressed to continue the game or "Reset" to play it from the beginning.
7. It MUST give the user four different options for the answer.
8. It SHOULD use a timer to challenge users on how many points they can score.
9. It SHOULD have separate databases for "Easy" and "Hard" questions and provide two start buttons relating to which set of questions the user would prefer.
10. It COULD allow the user to choose the length of the timer from 30 to 300 seconds (1/2 of a minute to 5 minutes exactly)

Non-Functional Requirements

1. It MUST provide an interface to the user that displays the question and all specified buttons in a single window.
2. It MUST be able to output at least 30 different questions to do with Python with appropriate answers.
3. It MUST be able to handle scores of at least three digits.
4. When "Start" is pressed, it MUST output the first question and answers within three seconds.
5. When the previous question has been answered, it MUST also select each follow-up question and answers within three seconds.
6. It MUST generate a follow-up question regardless of if the answer was right or wrong.
7. It MUST maintain the current score if "Stop" or "Start" are pressed, but set it back to "0" if "Reset" is pressed
8. It SHOULD start the timer at the number that the user requests and then decrease it to 0 at intervals of 1 second exactly.
9. It COULD remember the best scores and times and keep a leader board of them.
10. It WON'T be designed to run perfectly on MAC computers. This is to test if the game can run well enough on Windows first.

Design, Implementation & Testing (Design 3 wireframes)

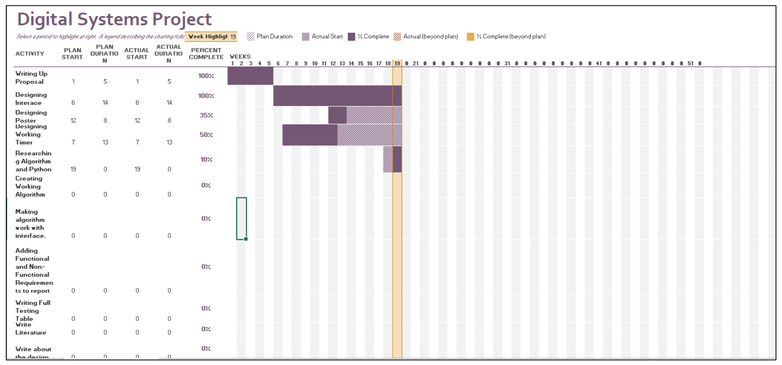
Wireframes



Testing

Test Number	Object	Description	Expected Outcome	Actual Outcome	Status
1	Timer	The timer can be set to any time between 60 and 300 seconds (between 1 minute and 5 minutes) The game ends when the counter reaches 0.	The timer should start going down by the second when the game starts.		
2	Score	This variable keeps track of the player's current score.	The score should go up by 1 each time a correct answer is given, but not change when the answer is incorrect.		
3	Start Button	The game should start when this button is pressed.	Pressing the score button makes the program start generating questions and answers as well as start the timer going down. It should not work if the game is running already.		
4	Stop Button	The game stops immediately when this button is pressed.	Pressing the stop button stops the timer and prevents any further questions from being answered.		
5	Reset Button	Pressing this button resets the score and timer and starts the game from the beginning.	Pressing this button sets the score back to 0, generates a new question and sets the timer back to what the player initially set it to.		

Planning and Management:



References:

Altexsoft (2021). Functional and Nonfunctional Requirements: Specification and Types. [online] AltexSoft. Available at: https://www.altexsoft.com/blog/business/functional-and-non-functional-requirements-specification-and-types/. [Accessed 13th November 2022]

Bi Worldwide (2022). What is gamification? [online] BI WORLDWIDE. Available at: https://www.biworldwide.com/gamification/what-is-gamification/. [Accessed 31st January 2023]

Growth Engineering (2018). What is the Definition of Gamification and What Does it Mean? [online] Growth Engineering. Available at: https://www.growthengineering.co.uk/definition-of-gamification/. [Accessed 31st January 2023]

Second Street. (n.d.). 5 Tips for Building Awesome Trivia Quizzes. [online] Available at: https://uplandsoftware.com/secondstreet/resources/blog/building-trivia-quizzes/ [Accessed 31 Jan. 2023].

www.tutorialsteacher.com. (n.d.). Create UI using Tkinter in Python. [online] Available at: https://www.tutorialsteacher.com/python/create-gui-using-tkinter-python. [Accessed 1st November 2022]

Testing Table: (Add this using Snipping Tool when done with it)

Test Number	Object	Description	Expected Outcome	Actual Outcome	Status
1	Timer	The timer can be set to any time between 60 and 300 seconds (between 1 minute and 5 minutes) The game ends when the counter reaches 0.	The timer should start going down by the second when the game starts.		
2	Score	This variable keeps track of the player's current score.	The score should go up by 1 each time a correct answer is given, but not change when the answer is incorrect.		
3	Start Button	The game should start when this button is pressed.	Pressing the score button makes the program start generating questions and answers as well as start the timer going down. It should not work if the game is running already.		
4	Stop Button	The game stops immediately when this button is pressed.	Pressing the stop button stops the timer and prevents any further questions from being answered.		
5	Reset Button	Pressing this button resets the score and timer and starts the game from the beginning.	Pressing this button sets the score back to 0, generates a new question and sets the timer back to what the player initially set it to.		

Before Starting Game

Python Trivia Game

Start

Stop

Reset

Time

000

Score

00

After Starting Game

Python Trivia Game

How would you finish a line containing a function definition?

“():” followed by indentation

“():” without indentation

“();” followed by indentation

“();” without indentation

Start

Stop

Reset

Time

300

Score

00

After guessing a correct answer

Python Trivia Game

What is the difference between a list and a tuple?

Correct

Values in a tuple are not separated by commas

Values in a tuple cannot be removed or altered

A list can have the same value twice while a tuple cannot

A tuple can only hold integers

Start

Stop

Reset

Time

293

Score

01