Statement of Purpose and Scope

Develop a statement of purpose and scope for your application. It must include:

- describe at a high level what the application will do

- identify the problem it will solve and explain why you are developing it

- identify the target audience

- explain how a member of the target audience will use it

Features

Develop a list of features that will be included in the application. It must include:

- at least THREE features

- describe each feature

Ensure that your features include the following language elements and concepts:

- use of variables and the concept of variable scope

- loops and conditional control structures

- error handling

Note: If the features you described do not require you to use all of the above it is likely that your application is not sophisticated enough. Consult with your educator to check your features are sufficient to address the assessment criteria.

User Interaction and Experience

Develop an outline of the user interaction and experience for the application.

Your outline must include:

- how the user will find out how to interact with / use each feature

- how the user will interact with / use each feature

- how errors will be handled by the application and displayed to the user

Control Flow Diagram

Develop a diagram which describes the control flow of your application. Your diagram must:

- show the workflow/logic and/or integration of the features in your application for each feature.

- utilise a recognised format or set of conventions for a control flow diagram, such as UML.

Implementation Plan

Develop an implementation plan which:

- outlines how each feature will be implemented and a checklist of tasks for each feature

1. Create a feature that will ask a user if that want play Guess Who.

1. It will need to ask if the player wants to play

2. It will need to figure out whether they’re a new player.

3. The function should only allow yes or no responses. This function is able to handle faults if incorrect input have been put in.

2. Create a Feature to ask for the users’ details:

1. Ask for user’s full name. This function can tell if the user has put in first and last name by ensuring that there is a space between names. Then the name is stored name in a name list

2. Ask user what gender They are. Store the name and gender in a dictionary called gender.

3. Ask user for hair colour. Store name and hair colour and in a hair list dictionary.

4 Ask user if they are follicly challenged (bald). Store name with yes or no response in a bald dictionary called bald.

4. Ask user for eye colour. Store name and eye colour and in eye a dictionary.

5. Ask user if they have facial hair. Store name with yes or no response in a facial hair dictionary.

6. Ask user if they wear glasses. Store name with yes or no response in a glasses dictionary.

3. Creating a Feature that create a temporary lists and dictionaries of each category.

1. Create a temporary list of all previous and new the player names. This will be used to eliminate users that do not match criteria that the program asks.

2. A function will be needed to create a temporary list. Within these lists needs to be all the descriptions within each dictionary. (ie all the different hair colours in the ‘hair colour’ dictionary.) This Will be used when generating questions during the game.

4. Create a feature that will generate questions for the user to answer.

1. It will need to be able display to the user how the game is played followed by asking if the user is ready. This function needs to be able to handle faults if an incorrect input has been entered.

If the user answers yes then start the game. If no, take player back to the start

2. This will need a function to generate a question using a random word from the temporary lists of descriptions.

3. The questions that are asked can only be answers as yes or no answers. This function will be able to handle faults if an incorrect input has been entered.

4. Depending on the users input, it will filter through the names who match the criteria.

5. A counter is to be implements to count how many questions it has asked.

6. After a few questions a function picks a name randomly and asks to user if its them, depending on the answer the program will either keep asking questions until the counter is up or completes the game.

7 If the computer has guessed it within 10 questions it will tell the user that the program has won. If not, it will ask the user if their name and congratulate them.

8 Needs to ask if the player wants to play again or exit.

- prioritise the implementation of different features, or checklist items within a feature

- provide a deadline, duration or other time indicator for each feature or checklist/checklist-item