M05: Final Project Planning Assignment

(2021-09-13) drm

**Objective**

This assignment is a continuation of the Project Plan you started in M04. The objective of the M04 assignment was to decide what type of program you wanted to create, including the topic (Balloon Shop, for example), and to provide rough sketches of what the two requiremed windows might look like.

The objective of this M05 assignment is to start flushing out more details for how you want your program to look and work. This includes thinking about what type of data the program will collect from the user, how the program will validate the data, and how the program will use the data. Another objective is to start defining UI elements, such as text boxes and buttons.

There is no coding needed for this assignment. It is all planning tasks.

You will use the same document you submitted in M04 and add to it. You may revise the previous sections if you wish. It would be a good idea to save your document under a new name, such as myname\_ProjectPlan\_M05, for example.

Upload your revised document to complete the assignment.

**Deliverables**

There are five deliverables for this project assignment:

1. Add a section to your document named **UI Elements**. Make a list of the data elements that will be presented in each window (you must have at least 2 windows). For each element, include a name, element type, description, and size (if appropriate).
2. Add a section to your document named **Window Positions.** Include the proposed position (grid or xy coordinates) for each element. Explanation and examples below.
3. Add a section to your document named **Events**. Describe what each button will do, in general terms.
4. Add a section named **Validation**. Identity how data entry fields will be validated, and if they are required.
5. Add a section named **Branding**. Describe any parts of the interface that will use your brand colors (this might be images, text, backgrounds, etc.)

**Process**

Consider the program description and sketches you submitted in M04. Your project must have at least two windows. The UI (user interface) for each window will contain elements such as labels (read-only description fields), text boxes and buttons. The windows may also contain other elements, such as check boxes, radio buttons, and images.

For each window, make a bulleted list of all the elements, including a name, a short description, and a size (if appropriate). Element names must be unique for each window, but may be re-used from one window to the next.

You will also need to plan on the location or position for each element, such as row and column, or an idea of an exact x/y coordinate (Examples below).

The sections below are similar to what you should add to your document. The examples

**UI Elements**

A GUI can be constructed with many types of elements. Elements are basically variables. For each window, each element must have a unique name. Developers often develop a naming scheme, so that elements of similar types have similar names. Examples:

* lblFirstName: label, displays "First Name:"
* lblLastName: label, displays "Last Name:"
* txtFirstName: text box for entry of first name, 100px (100 pixels wide)
* txtLastName: text box for entry of last name, 100px
* txtAge: text box for entry of age, 50px
* btnClear: button to clear data entry fields

A label is read-only text. These include descriptions for data entry fields, instructions, messages, and sometimes titles.

The naming scheme above use Camel Case and prefixes of "txt", "lbl" and "btn" to identity elements that are labels, text boxes and buttons, respectively. This type of scheme can be helpful when there are many elements and long sections of code.

Text boxes will often be sized (in pixels) to reflect what type of data the user might enter. The txtAge field above is shorter compared to the name fields, because it is anticipated the user will only enter two digits. Notice that ALL text boxes are text (strings). To be used as a number, the text box value will need to be converted.

**Window Positions**

Each element on a UI will exist in a relative or absolute position. "Relative" means in relation to other elements. "First", "second", "last" or all relative positions. So are "left", "right", and "below".

Most GUI software provides multiple ways to position elements in the window. Two common ways are using a grid (rows and columns) and using absolute coordinates (x/y).

For this assignment, provide the position for each element (using either a grid or x/y positions). These positions do not have to be exact, and may be changed later. The objective is to draft a general plan.(This might seem like a needless exercise, but your Python code will be required to provide either a grid location or exact position for every element.)

**Example of Element Positions**

Provide positions for all of your elements like below. In these examples, Window 1 uses a grid system and Window 2 specifies x and y coordinates. Your program should only use one or the other. You will learn more about how to specify positions in M06.

Window 1 (main window)

* lblProgramName: row 0, column 0
* lblFirstName: row1, col 0
* txtFirstName: row1, col 1
* lblLastName: row 2, col 0
* txtLastName: row 2, col 1
* lblAge: row 3, col 0
* txtAge: row 3, col 1
* btnClear: row 4, col 0

You are not required to create tables like those below, but they may be helpful.

Conceptual layout of fields in grid:

|  |  |  |
| --- | --- | --- |
|  | col 0 | col 1 |
| row 0 | lblProgramName |  |
| row 1 | lblFirstName | txtFirstName |
| row 2 | lblLastName | txtLastName |
| row 3 | lblAge |  |
| row 4 | btnClear |  |

Conceptional visible layout

|  |  |  |
| --- | --- | --- |
|  | col 0 | col 1 |
| row 0 | **DEWEY's Calculator** | |
| row 1 | First Name: |  |
| row 2 | Last Name: |  |
| row 3 | Age: |  |
|  |  |  |

Window 2 (Payment Quote)

* lblProgramName: x=30, y=20
* lblFirstName: x=30, y=70
* lblFirstNameData: x=130, y=70
* lblLastName: x=30, y=120
* lblLastNameData: x=130, y=120
* lblAge: x=30, y=170
* lblAgeData: x=130, y=170
* btnClose: x=50, y=220

***Notice***: all the elements on Window 2 are labels! There are no data entry elements on Window 2 because it only displays the output of the program calculations. The element names may be reused from Window 1, but all names must be unique. So it is not possible to use the name "lblFirstName" for both the descriptive label and for the data. The data element must have a unique name.

**Events**

The purpose of this section is to prepare for coding what happens when each button is clicked.

Example:

btnCalculate: calculates a loan payment amount. Converts both txtRate and txtLoanAmount into floats. Converts txtYears into an int. Calculates loan payment based on loan amount, rate and number of years. Displays payment amount in lblEstimatedPayment.

btnClear: clears all data entry fields and unchecks any check boxes

**Validation**

txtLoanAmount: Required. Must be a number.

txtRate: Required. Must be a number.

txtYears: Required. Must be a number between 5 and 20, inclusive.

**Branding**

Branding colors are Gold and Lime Green.

The main window will have a background of Gold.

The second window will have a background of Lime Green.

One image used will be a logo for the company, displayed on both windows in the upper right.

A second image will be some type of Association logo, such as the Realtor "R"; this will be displayed in the lower right corner on each window.