|  |  |  |  |
| --- | --- | --- | --- |
| **Task -** | | | |
| Each unit has steps you have to take. Complete each section in as much detail as possible – your answers will help to complete the unit | | | |
| **About your task or Activity what tools did you use ?** | | | |
| I used my CMD version of this program, as a baseline, and and then used Tkinter to make a framework, and meshed the two together.  I used pygubu as my designer, and that served me very well. The grid and snap features were what I used most, and the tree system worked amazingly for this style of design.  I created a framework using pygubu and then used my CMD version as a baseline, fusing the two code bases together into a functional program by grabbing the variables from the UI and passing them onto the CMD version’s logic. | |  | |
| **Organising your task…** | | | |
| ***How did you finalise your task and did it suit its purpose?***  I finalised the task by adding a user data input, which I had forgotten, and adding input blocking  so that I would be able to stop users from entering numbers into objects that required strings  and vice versa.  I checked the logic alongside my CMD version, and the numbers that it was required to put  out and it worked perfectly.  I feel it suits its purpouse more in this python state, rather than in a C# state, because a C#  version would be single platform, whilst this is cross platform to anything that can run python.  It works perfectly, and inputs and outputs data as it should. |  | | |
|  | | | |
|  | | | |
| **Reviewing Your Work**  ***Did you have any problems? If so how did you overcome these issues, are there any steps you could take to prevent similar instances in the future.***  My largest problems were that I didn’t know how to use Tkinter at the start of the task, nor did I know *anything* about UI based programming, so it was quite a risk to jump in the deep end like this, it was quite a shock using Pygubu initally, but then once I got used to it, it worked perfectly for me.  The second problem was working out how to pass the data from the logic to the frontend and vice versa, so that I would be able to pass the data outwards, with the right inputs and outputs, and this required me to understand new syntax and logic, but once I had understood this, it worked perfectly.  I had to convert my price output from a decimal (rounded to 2 sig figs) to a float, because the output of the tkinter system did not support decimal functions, but did support floats, and thus I had to do this.  I also had to change my array of prices, to merely just outputting the prices of each material directly into the equation, pulling it from the radio button’s outputs and putting it into a price variable.  Alongside this I made changes to the mathematical processing of the program, and the way it iterated to achieve more accurate results.  There were no bugs that could not be repaired.  I would next time use a framework I am more used to, which I was unable to do as I wasn’t comfortable with any framework up till now, and understand UI based data gathering better. | | |  |
| ***What improvements could you make if asked to do something similar in the future?***    I would make a more visually appealing program, with a more pleasing design and a less spartan color scheme.  I would also use a framework I understand rather than learning fresh. | | |  |
| ***Did you receive any feedback from anyone?***  ***The feedback I received was the program looks a little old, and it doesn’t work amazingly in a tilling window manager, alongside it being a Python program, rather than a conventional EXE*** | | |  |