**Evaluation**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Objective** | **Performance criteria** | **Evaluation** |
| 1 | It must be able to import an image, that the user uploads, onto the screen. | There must be an image visible on the screen when the user imports one from the dialogue. | -COMPLETELY ACHIEVED - My program allows the user to import an image using the upload feature, this image shows at the correct resolution to which it was uploaded at. Additionally, the user can use many image file formats, i.e. png, jpg and gif, to upload. This is useful as it allows for the user to have control. If the user received a jpg, they do not have to convert it to a png. |
| 2 | It must allow the user to rotate the image. | The user should be given the ability to rotate an image via the user interface. | -COMPLETELY ACHIEVED- My program allows the user to rotate an image to whatever orientation they choose using a slider on the side, one advantage of this is that sliders allow the user to also use the arrow keys on a keyboard to do small adjustments to a rotation so the user has full control on rotation from the GUI. This feature is contained as a background as it gives the user a bundle of lots of helpful features together, a background cannot be dragged by the user but can be rotated, this means that the user can set up the environment they want and then have it lock to suit their needs. |
| 3 | It must allow the user to navigate this image, zoom in and out and move around the image. | The user should be able to navigate around the workspace by using simple controls. | -COMPLETELY ACHIEVED- My program allows the user to navigate around one image or the entire workspace. This is done by clicking and using a button on the side to set the mode. This minimises the amount of complication for the user.  The user can zoom in and out by using the buttons on the side of the screen which can be clicked by a mouse, the screen reacts and zooms in, giving the user a closer look at their target area. |
| 4 | It must be able to create rectangles which display on screen and the user should have full control over the size of these rectangles. |  | -COMPLETELY ACHIEVED- My program supports the creation of rectangles; the user simply presses the shape button to create a rectangle. The user can use the resize dots on screen to change the size of these rectangles, this is a simple and visual feature for easily editing the size of the shape, the user just clicks and drags them. |
| 5 | It must be able to show the sizes of these rectangles with a real scale and not just pixels. |  | -COMPLETELY ACHIEVED- With the show dimensions button they can get the dimensions of the rectangle written on top of the rectangle its self, this means that the user doesn’t have to look away or press a button to get this, since getting dimensions is frequently needed, having the dimensions always in eyesight is a feature which should be very useful to my end user, this displays pixels if the user has not set a scale. IF the user has set a scale (via the scale button) then, the screen automatically updates and shows the dimensions in the scale provided. |
| 6 | It must have a user interface, using buttons or other graphical elements to control the program. |  | -COMPLETELY ACHIEVED-  My project uses HTML to show buttons and input, further to this I have sections of buttons which show depending on if they are needed or not. The actual canvas area is created using JavaScript, with all visuals created manually, this allows me to make custom behaviours like stretching images. |
| 7 | It must support multiple rectangles at once. |  | -COMPLETELY ACHIEVED-  The program uses a layer’s system so the rectangles work independently and can be resized and manipulated without effecting other rectangles. This also allows for overlapping rectangles and they display correctly. Some features even have multi rectangle integration, for instance, when a rectangle is created, it takes on the colour of the last rectangle as it is likely to be the colour they want. Zooming works with multiple rectangles, they scale and keep the context between rectangles. Saving the file works with as many rectangles as the user wants. The user could have hundreds of rectangles if they want. |
| 8 | It must be able to turn virtual units (i.e. Pixels) into real world units (i.e. Metres). |  | -COMPLETELY ACHIEVED-  Using my measure feature, the user can drag their mouse to set a distance, then the converted distance will appear in a dialogue box, this means users don’t have to type out a value. They can measure an area as if they were there with a perfectly straight tape measure, this helps the user as it emulates what would happen in real life, making it easier for the user. |
| 9 | It must be usable, with no complicated features or application breaking bugs. |  | -COMPLETELY ACHIEVED- As per my end user, the user interface is well done and easy to use, there are no complicated features which only professionals could use and there are 4 simple modes which the user can select, drag, move, scale and measure. This keeps the interface simple and only requiring the press of a button to change modes from one to another.  There are no application breaking bugs in my code, the program runs smoothly and pulls no errors, the program has never crashed in testing. |

Optional Objectives

* My program **does** calculate the Total grass area, this is displayed in the top right.
* My program **does** allow the user to save their session. This session can be saves as a file to the user’s PC. The user can then load this up at a later date by importing the file.

Additional features not included in Objectives

* Opacity, layers can be opaque, to see underneath them, this helps with user experience to see below a layer (could be used if background still needed to be somewhat visible)
* Colour, User can set the colour of a rectangle with a simple dialogue.
* Clear all, User can press a button to clear the whole environment, if they want to restart.
* Delete Layer, User can delete layer at press of a button.

**Overall Review**

Strengths

This project has been able to complete all the objectives conclusively, the solution works very well and is simple to use. It can easily mark out an area using shapes and can give the user accurate prices with a real scale. The user always has control of the environment and has the freedom to use the tool however they want. It provides a more custom solution to the problem than the makeshift Photoshop usage before.

Improvements

There are some improvements which I would have liked to implement given ore development time, keyboard shortcuts would make it easier for a user to perform tasks, rather than having to click on a button there could be a shortcut for it i.e. ctrl+c for copy, ctrl+s for saving, D for drag mode etc.

Another feature could be the ability to drag an image file from the user’s computer over the application its self to import images rather than needing them to click a button. However this is a minimal improvement.

End User feedback transcript:

What are your thoughts on the solution?

“It’s very good! The user interface is simple to navigate and I don’t get lost like I did with photoshop. The buttons at the side are really useful and I like how they hide when I haven’t clicked on a rectangle. I will be adopting the program to quote my jobs now. Its also good how the background thing works. It just locks it to stop it from moving, which is very helpful when you are as clumsy as I am.”

What improvements could be made?

“Hmm, not many I can think of except possibly adding a few keyboard shortcuts, it might be useful if I was on a laptop with a trackpad, but it’s probably not essential.”

Did the solution meet the requirements?

“Definitely, im very happy with the program and it does what I hoped it would.”

**Conclusion**

My program has met all of my required objectives and two of my optional extensions. Aditionally it has some extra features which were not mandated by definitely help and add to the user experience. The feedback from my user was very positive and I’m sure they are satisfied with my program. For this I can say my project has definitely been successful.