Reflection for sprint 1

In the video above you can see the code responses correctly to the expected input, creating a rectangle of the correct proportions. Including not creating a shape when the mouse is out of the canvas.  There is one main issue with this sprint and this is the circle because it responds to the width. When a rectangle is very long and not tall, the circle will have a larger radius than the height of the circle, therefore the circle is outside the rectangle. To fix this I will add an if statement if radius > height made the radius  the height divided by 10.

Another main issue is that if you go out of bounds it does not draw the rectangle to the bounds, only making no rectangle appear. To fix this in the next version I will add a cut piece of code to allow the rectangle to form but only cutting off the part that is out of bounds.

**Testing**

Both the circles and the ellipse have advantages and disadvantages. Both groups currently have the disadvantage of being once the mouse leaves the area they do not work once coming back into the canvas. They also both expand from the centre, which can be changed later

The ellipse can also be a circle if the width and the height are the same, the circle however can only be a circle not an ellipse. Therefore I believe that the ellipse will be more helpful as a drawing tool due to its versatility.

Plan sprint two

The plan for sprint two is to make a functional drag and drop ellipse function. This is based on testing done above as I believe that an ellipse is a versatile tool that will be helpful and entertaining to use. My aim is to have a rectangle outline when the ellipse is being drawn and to expand from one point to another instead of from the centre.

Reflection

In the video above you can see the code responses correctly to the expected input, creating an ellipse of the correct proportions. Including not creating a shape when the mouse is out of the canvas. A main issue is that if you go out of bounds it does not draw the ellipse to the bounds, only making no ellipse appear. To fix this in the next version I will add a cut piece of code to allow the ellipse to form but only cutting off the part that is out of bounds.

Plan for sprint 3

The plan for sprint three is to make a designated drawing space within the canvas, this allows the shapes to only be drawn in this drawing space. I would also like to add some buttons to allow the user to choose if they would like to draw a rectangle or an ellipse.

I have an error when I try to draw a shape from the buttons there seems to be a delay in the outlines for the shapes. When the first shape is drawn it has no outline and then after I change shapes the original outline remains for the first shape of that type. Another error is the boundary as there currently are none so a shape can be drawn anywhere.These will be fixed in the next sprint, as I am happy with the buttons and canvas function

Sprint 4

The aim of this sprint is to fix all the errors from the previous sprints to make a function code. To do this I will add a boundary function to test the parameters of the rectangle canvas. I would also like if the shape is partly in the canvas only that area is shown.

Testing

I have successfully solved two of these issues from previous sprints. We now have a responsive

How I plan to fix this is using a clear function to crop the outside of the canvas rectangle because currently the shape can be drawn from inside the rectangle to the outside, but when it starts at the outside it can’t come into the rectangle.

Sprint 7

Plan

For this sprint I wanted to test two options for clearing objects outside my drawing canvas area. The first idea was to get the boundary and where the shape finished  and use these values to clear a rectangle. The other option was to save the shape then clip it to make that the only canvas area when the mouse is down.

Testing

I believe that the clip function is best for this problem as it does not require values from the mouse to work, as well as clipping the dragged shape. The finding boundaries code works however the clear function also clears the grid in the background, which means it would need to be updated regularly.

Reflection

I am very happy with this sprint and the testing done to find the best solution to allow the user to only draw shapes inside the given canvas area. The aim for the next sprint is to have an undo and redo buttons to allow the user more freedom.

Sprint 8

Plan

For this sprint I wanted to allow the user to change their minds or start over, as when getting others to try my code they cited that this was a concern. To remedy this issue I will make undo and clear buttons. The undo will remove the last action the user did and the clear will return the canvas to the original look.

Reflection

I am happy with my undo and clear buttons as they give users more options, in order to make the user think before accidentally undoing their whole projet, the undo button gets deselected after being used. This means the user can undo more than once however they have to make a conscious decision to do so.

Sprint 9

Plan

The plan for this sprint is to add more options for the user, this will add a line and a smiley face. As seen above there were many questions surrounding how the smiley face would look. I then decided to draw a smiley face and model more code off this. I also wanted to allow the user to choose how width they wanted to make the line/outline. Therefore I needed to make a separate class for width buttons, in which there would be three, small, medium and large.

Reflection

The buttons work and I am pleased with the outcome as they resemble drawings I would make with pen and paper. The width function gives the user more options for how they want to personalise and use the application. For the next sprint I would like to change the amount of points the star could have to give the user further options.

Sprint 10

Plan

The plan for this sprint is to give the user more options when drawing a star. I wanted the user to have four options for the number of points they wanted. When I first looked into making stars, I was trying to decide between five and six points so I knew that I wanted to include these as options. I also want to include a seven and eight pointed star, because I believe these will be most helpful and are visually appealing.

Reflection

After some adapting I changed part of the function for the star, to allow the points to always be evenly spaced apart. This meant that I needed to get twice the amount of points that would be visible because this is drawn based on the lines. This however was successful and we are now drawing symmetrical stars that work for a variety of points.

Relevant implications

* Cultural
* Usability
* Functionality
* Sustainability and future proofing

**<h3>Usability </h3>**

<p> Usability is how easy the program is to use, this is done by clear informative user feedback to make the experience more enjoyable. </p>

<p> I believe that my drawing program is user friendly because there is a hover function over the buttons and the user therefore knows where they are and what their options are. The naming is also mainly self explanatory with the width being defined as S, M, L to indicate small, medium and large. This is the same with the number of points on the star, the numbers start from 5 to indicate to the user that these buttons do not control the rectangle, ellipse or heart. So by process of deduction it must  control the star. Also to help the user if a line width or a star point number is not selected but the object is selected, these values have a default. </p>

**<h3>Functionality</h3>**

<p> Functionality is that the program works properly this includes not crashing despite the users interesting inputs. It also needs to meet the users requirements of being able to add and subtract pizzas, add delivery/pick up option, confirming order, closing the order and starting a new order. </p>

<p>In order to test the functionality of my code, I got others to test my code as seen in the video above. Despite the users changing it’s mind and making sessions that might not always be what was expected. This all worked without crashing, only calmy displaying the user with an error message that was targeted and helpful. This was because I systematically tested after each sprint.</p>

**<h3>Sustainability and Future Proofing</h3>**

<p>Future proofing and sustainability is the task of ensuring the code can be used by future generations, through easy update options, adding new features, and another developer can work on with ease. This is mostly deployed in the main.js with the buttons, this allows a person to update where the buttons are positioned based on the x, y, w, h programmed at the start which makes it  </p>

<p>Code commenting and doctyping to make it easy to follow and understand with basic code knowledge. The code starts from a main function and this allows another developer to work on with minimal distress. I tried to explain my thought process through this code. </p>

<p>This was a big thing for me as this helped me to make decisions regarding lists and how to use them. In the first 5 sprints I only had a pizza list of three pizzas, this was because it was easier to manage in the early stages. However for the final version I changed this to a list of 10 pizzas for Marsden Pizza’s menu. This showed the ability for the code to be changed  if needed, this is a great example of sustainability, and future proofing methods. The pizza list was not the only list I used, we also had delivery and customer order lists, these stored data, to then be used again and again. Another sustainability feature of this code is the functions, these validations functions are used multiple times one is for getting a string and the other is for getting a number. The only function used less than three times is getting the phone number because this is specialist code only used twice. If I had more time I would work out a way of integrating the phone number validation into the getting number function. </p>

<h3>What have I learnt from this planning process?</h3>

<p>I have learnt about structuring the planning into manageable chunks, this made hard ideas much simpler and easier to follow. Each Sprint took about a week and focused on one or two main ideas. This started from the most basic code that could be needed, and added a new function that I discovered I could need.<p/>

<p>These Sprints were very achievable and allowed me to feel a sense of accomplishment on completing each one. </p>

<h3>GitHub? </h3>

<p>I used github as a control center for all my code and sprints. The dashboard helped me to structure and keep track of my ideas. With the doing, to do and done column for each sprint. This was also where I stored the project backlog, an ever updating file with what was happening and what had the highest priority and what was next in the big scheme. </p>

<p>I would also push all my code to github, this allowed me to have only one version of the code on my computer, with the other sprint codes being stored in github. This also records the date and time of each sprint, with relevant comments from myself. As seen below</p>

<h3>Decision Making</h3>

<p>In sprint 4 I had to make a large decision regarding validation and the phone number. To make this decision, I used a trial that clearly showed me which option was best and I learnt more about python and regex.</p>

<h3>Project Backlog</h3>

<p>Project backlog was my hub for all the ideas and things that needed to be done. These were big projects that would become sprints. The project backlog was also where ideas were placed in priority order. This was a document that changes regularly to adapt to the newest high priority item. </p>