11055 - Programming for Design - Project 1

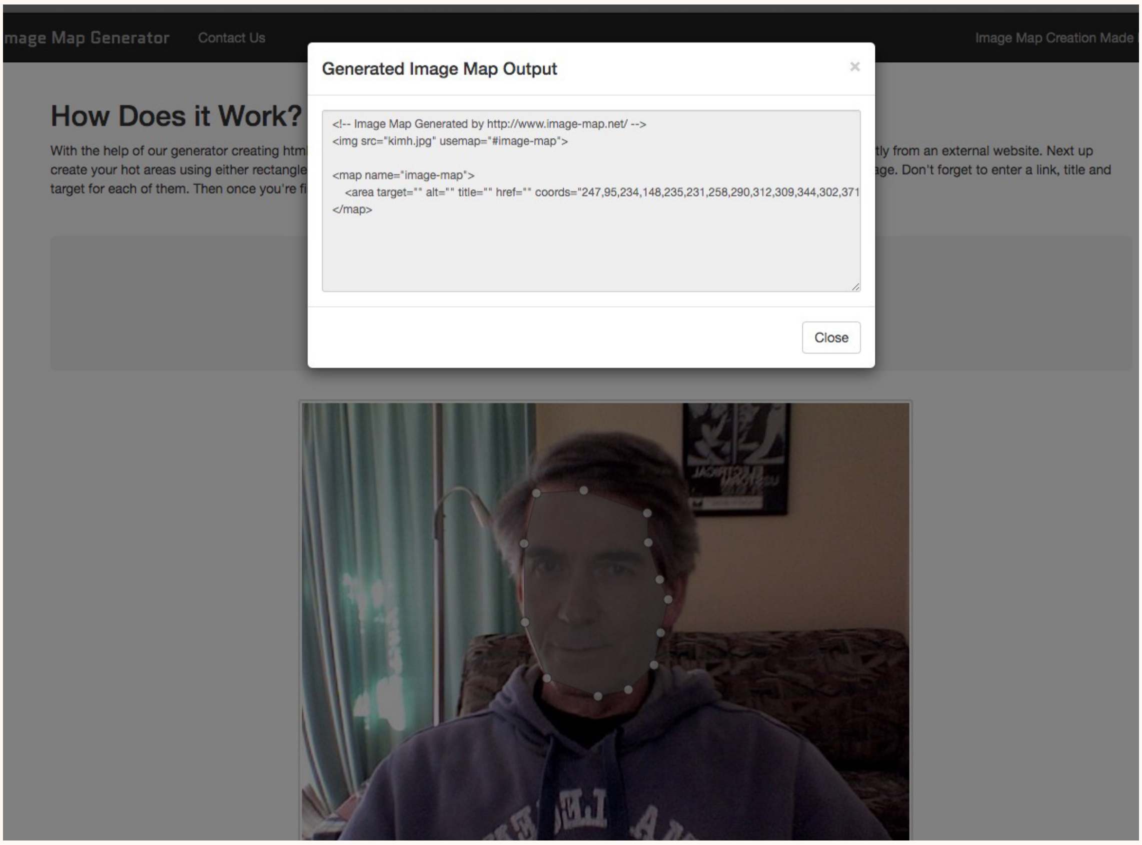
P5 Self Portrait Reflection

12th September 2019

u3179196 Kim Hughes

This project proved to challenging and the final result seemed sufficient albeit far from satisfactory. Having reviewed the options in P5 available for such an exercise I determined the options were broadly speaking to use beginShape() and endShape() - without any value of the kind parameter - to define vertices's of (largely) irregular polygons - or to combinations of quad(), rect(), ellipse(), triangle() and line().

I found and reviewed an Image Map Generator (https://www.image-map.net/) which generates the (x,y) edge coordinates of an images hot spots as specified by the user on an uploaded image with results be displayed in an HTML image-map within <area> tags. While the approach initially seemed to hold merit it proved somewhat futile given the complexity of all the facial features.



How Does it Work?

With the help of our generator creating htm
create your hot areas using either rectangle
target for each of them. Then once you're fi

tly from an external website. Next up
ge. Don't forget to enter a link, title and

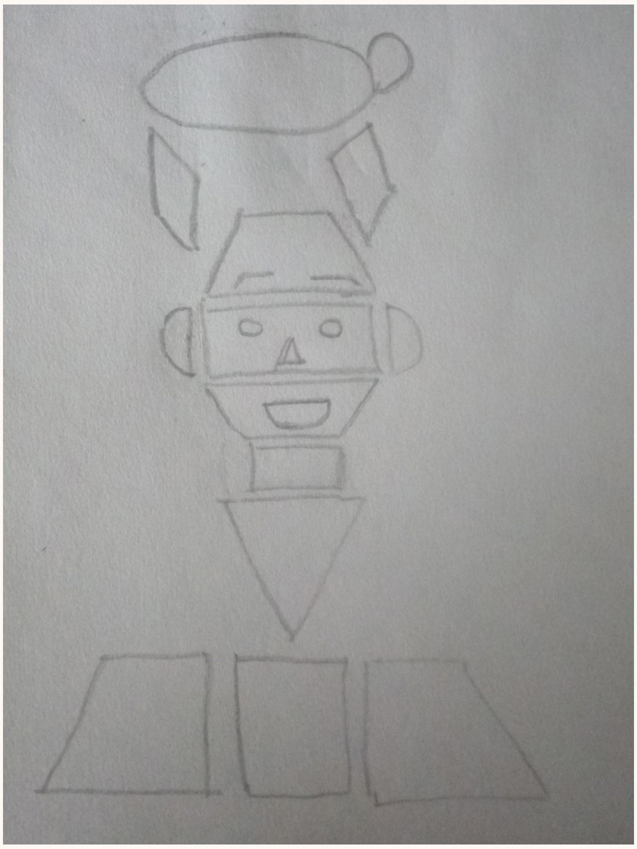
```
<!-- Image Map Generated by http://www.image-map.net/ -->
<img src="kimh.jpg" usemap="#image-map">

<map name="image-map">
   <area target="" alt="" title="" href="" coords="247,95,234,148,235,231,258,290,312,309,344,302,371
</map>
```

Given my lack of drawing skills I opted for the approach of constructing the portrait using the illustrator shape builder methodology rather than draw a freestyle shape in outline mode and incorporating emphasis lines and more complicated fills and strokes to effect contours and shadows (begin beyond my drawing and visualisation skills).

I originally used (an arbitary) canvas size of 500 x 500 px then increased it to 1000 x 1000px and used the scale() function to increase the size of the drawing after finding the original to be too small.

I sketched out the outline for each component - hair, face and shirt - and determined to shapes required construct (an admittedly crude) representation of the portrait. I then considered facial features such as eyes, ears, nose and mouth from a shape perspective.

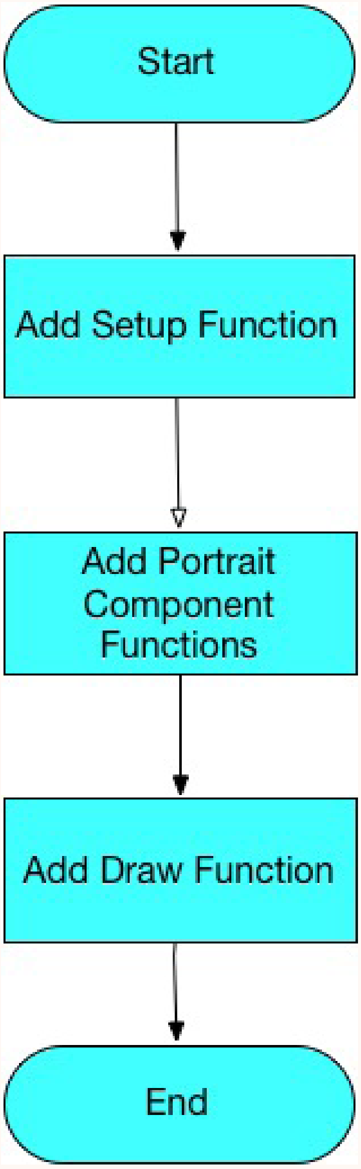
Given that P5.js works in a layered manner (similar to Photoshop or Illustrator) I also needed to consider the order of rendering each component in the context of the shape building approach I had opted for.



The result of this shape building construction is very robotic portrait. Attempts to use P5 arc() to round corners proved frustrating with copious trial and error and given a lack of math (and drawing) skills this approach was abandoned.

I adopted an approach of creating functions for each major drawing component and calling these within the draw function. Considering the trivial nature of this exercise from a coding perspective this was not necessary and would have be acceptably achieved by appropriate ordering of the shapes being rendered directly from the draw function but it seemed like best practice to call functions in terms of code maintainability and readability strategy. Similarly I used variables for fill colors - again not necessary in this simple exercise but best practice to adopt in principle.

The program flow logic is simple as no conditionals, decision points or storage is involved. The process is to add the set up function, create a function for each portrait component and finally add the P5 draw function which calls each component function as described in the flowchart diagram below.



As per flow diagrams the need for psuedocode was not required as programming logic (in particular the development of algorithm’s) was minimal. Clarity is a primary goal of pseudocode - in particular explaining what each line of the program should do. The exercise in this case was simply to (by trial and error) determine what parameters a shape should be and in what combination with other shapes to produce an output with a resemblance to a face. As a secondary consideration the grouping of shapes in components (hair, face, shirt) giving consideration to sub-grouping components into sections commented appropriately (such as ears, mouth, eyes, ...).

For the P5 portrait code the need to, step-by-step, develop a written outline of the code which can be gradually transcribed into the programming language was not needed to any significant extent. The decisions involved were largely concerning the method of drawing the portrait having first considered what type of portrait would be developed (largely a choice between a shape based construction or line drawing outline).

Having decided on the build by shapes approach (creating an abstraction with no recognisable resemblance to myself excepting roughly the hair) the programming decisions were then reduced to breaking the shapes into sections - in this case being face, hair and shirt- then deciding what shapes (P5 2D primitives) were required, the precise shape parameters and the attributes of those shapes largely relating to color.