# Project1.html

set browser page title to “Project 1: Self Portrait”  
link relationship to W3 style sheet  
  
create light grey panel  
 display heading “Programming for Design (11055)  
 display sub-heading “Project 1 – *Self Portrait*”  
 display paragraph “Keir Herbert (u3211239)”  
end panel  
  
create light grey container for control panel  
 create dark grey container for title bar  
 display in bold “Controls”  
 end container  
 create container for text labels  
 display paragraph “red”  
 display paragraph “green”  
 display paragraph “blue”  
 display paragraph “shape size”  
 display paragraph “frame rate”  
 display paragraph at position relative to top (20px) “geometry”  
 end container  
end container  
  
point to p5.js script source  
point to coordinates.js script source  
point to getScaleFactor.js script source  
point to createTriangle.js script source  
point to calculateDistance.js script source  
point to selfPortrait.js script source

# selfPortrait.js

function **setup**  
 constant **matrix** = 200  
 constant **mouseClearance** = 25  
 create a canvas size = **matrix** \* **getScaleFactor**  
 position canvas 150, 75  
 shape outline stroke = off  
 create **redSlider**  
 range = 0 to 255  
 initial = 63  
 width = 140px  
 position = 8, 255  
 create **greenSlider**  
 range = 0 to 255  
 initial = 81  
 width = 140px  
 position = 8, 292  
 create **blueSlider**  
 range = 0 to 255  
 initial = 181  
 width = 140px  
 position = 8, 329  
 create **heightSlider**  
 range = 2 to 48  
 initial = 12  
 width = 140px  
 position = 8, 366  
 create **diameterSlider**  
 range = 2 to 48  
 initial = 8  
 width = 140px  
 position = 8, 366  
 create **framerateSlider**  
 range = 1 to 24  
 initial = 5  
 width = 140px  
 position = 8, 403  
 create **shapeSelector**  
 option 0 = triangular  
 option 1 = circular  
 width = 95px  
 position = 24, 244  
 default = 0  
end function **setup**

function **windowResized** *(event driven when window moved or resized)*  
 var **scaleFactor** = **getScaleFactor**  
 resize canvas (**matrix** \* **scaleFactor**), (**matrix** \* **scaleFactor**)  
end function **windowResized**

function **draw**  
 clear screen  
 set framerate = **framerateSlider** value  
 set fill = **redSlider** value, **greenSlider** value, **blueSlider** value  
 var **scaleFactor** = **getScaleFactor**  
  
 let **row** = 0  
 do while **row** < length of **coordinates** array  
 var **x** = **coordinates**(**row**, 0) \* **scaleFactor**  
 var **y** = **coordinates**(**row**, 1) \* **scaleFactor**  
 **row** = **row** + 1  
 var **distance** = **calculateDistance**(**x**, **y**, mouseX, mouseY)  
 if **distance** > **mouseClearance** then  
 switch case based on **shapeSelector** value  
 case 0  
 hide circle **diameterSlider**  
 show triangle **heightSlider** var **height** =random number from **heightSlider** value to 2  
 call **createTriangle**(**x**, **y**, **height**)  
 break  
 case 1  
 hide triangle **heightSlider**  
 show circle **diameterSlider**  
 var **diameter** = random number from **diameterSlider** value to 2  
 plot circle(**x**, **y**, **diameter**)  
 break  
 else  
 do some colour trickery  
 loop  
end function **draw**

# getScaleFactor.js

function **getScaleFactor**  
 if screen.Height < 720 then  
 return value of 2  
 if screen.Height < 900 then  
 return value of 3  
 if screen.Height <= 1080 then  
 return value of 4  
 if screen.Height < 1440 then  
 return value of 5  
 else  
 return value of 6  
end function **getScaleFactor**

# calculateDistance.js

function **calculateDistance** receive values and assign to variables **xPoint1**, **yPoint1**, **xPoint2**, **yPoint2** var **xDiff** = **xPoint1** - **xPoint2**  
 var **yDiff** = **yPoint1** - **yPoint2** return value (square root(xDiff² \* yDiff²)  
end function **calculateDistance**

# createTriangle.js

function **createTriangle**  
 receive values and assign to variables **x**, **y**, **height**  
 var **segment** = height / 3  
 var **xPoint1** = **x**  
 var **yPoint1** = **y** - **segment** \* 2  
 var **xPoint2** = **x** + **segment** \* 2  
 var **yPoint2** = **y** + **segment**  
 var **xPoint3** = **x** - **segment** \* 2  
 var **yPoint3** = **yPoint2**  
 draw triangle(**xPoint1**, **yPoint1**, **xPoint2**, **yPoint2**, **xPoint3**, **yPoint3**)  
end function **createTriangle**

# coordinates.js

var **coordinates** as pseudo-multidimensional array  
populate with data line-by-line