Carlos Gratacos
Prof. Megret
COMP4046

Assignment 2

Part I

Q1) Analysis of existing code

NOTE: I modified the code, but this is from the original file.

By analyzing the code, when the Javascript engine loads the file and runs the first stage (of the two stages) it activates it's "strict mode", after the creation phase of the Javascript engine, which loads and creates the memory space needed by the JS variables and functions of the file, it starts its second stage, the execution phase. In this phase, when all the variables are loaded, the first execution context that is ran by the JS engine is the start function (it is called by the JS event handler when the body of the HTML page ends loading) which initializes:

1) obtaining the canvas context,
2) obtaining the canvas width and height and assigning them to variables w and h,
3) initializes the params object and assign variables, the most important are x0,y0 which initially x0 = w/2 and y0=h/2, this set of variables are used later on (will explain) for the drawing of the ship in the middle of the canvas.
4) Then it will initialize the dat GUI instance folder (with the initGUI function) and assign the variables inside the object params to the dat folder with their value range.
5) Then it will start the key event listener inside the initKeys function, this function initiates the custom key press handler and listens, initially to the keyUP and keyDown event, each event listener is set to the capture event, and have as callback functions that toggle the values of a Boolean variable.
6) Finally, the tick function is initialized, this function
    a) starts the animate function which makes the necessary calculation of the time changed between current frame and last frame for animation, these parameters are saved in global variables, then
    b) starts the the handleKey function which listens to the status of code 37(Left Key), 39(Right Key) and rotates when one of this key is on, the params.angle changes, this change affects why the ship's angle changes. After handling the keys events,
    c) the animation process starts which redraws the ship with the changed parameters. Initially these parameters are the ones from the params object, and the canvas is cleared. Before drawing, the function defines the affine transformation. This are the setTransform(a,b,c,d,e,f) where in a 3x3 matrix Row1=[a,c,e] Row2=[b,d,f] Row3=[0,0,1], which defines the skewing and scaling, in this example they aren't use, this is why it is set to the default. Then applies a translation to move to x0 and y0. Finally it applies the rotation with the rotation R = params.angle /180*Math.PI. The angle variable is changed when the key left or right are pressed or the dat GUI angle variable are change. Finally, the ship is

draw and reset the setTransformation to default, in case some JS inner function affects it.
d) Finally the requestAnimFrame is called with the the tick function as a callback and this is why the callback lopping which maintains this steps a-d running until the browser tab is closed or the canvas is hidden.

Part 2

Q3C) Geometry, Transform Points using 3x3 Matrix
Analyzing this part, What I did to satisfy the requirements is create a Matrix and pass this trnaformation to the Matrix and then apply the transformations from the Matrix to the points in the original canvas to the transform canvas, so they can be seen on the tranform canvas. The transformation operations where in the order:
Translation->Scale->Rotation->Translation,
this last Translation is of (originalCanvas.width/2,0), this is because if it is not applied the red pin would be the in the blue box border, where when applied the red pin is moved to the location is supposed to. But there is a trick, that when you see the transformation points the Matrix view on the bottom of the original canvas, the representation of x0 in the GUI is seen as (x0 + originalCanvas.width/2). To change this back, the trick is that after trnasforming everytinh that need to be transform and printing the matrix, do a translationon the matrix of (-originalCanvas.width/2,0) and then print, this will present the matrix and will fullfill the requirements. What I used to observe this changes was the Matrix representation of it and the console.log JS function

Q4) Drawing and Geometry, Transform the image
The parameters of the setTransform in the canvas are of the form:
ctx.setTransform(a,b,c,d,e,f)
this represent:
| a c e |
| b d f |
| 0 0 1|
on a 3x3 matrix, so the values of the 3x3 Matrix M passed to the function are a=M[0], b=M[1], c=M[3], d=M[4], e=M[6], f=M[7]. This is because the array matrix M of a mat3 element from the gl-matrix are stored in a column major array and not row major.