JavaScript Review 1

1. Download gameLoopA.html and complete the following exercises.

   The gameLoopA.html code should generate randomly placed 20x20 pixel squares on the screen.

   a) Debug the code to make it work.

   b) How and when does the init() function get called?

   c) Why are the rectangles appearing so fast?

   d) Do the rectangles keep drawing when the window is minimised?

2. Using your working gameLoopA.html code, adjust the speed at which the squares are being drawn.

   This can be achieved by using the currentTime variable that is passed-in (or given) to the gameLoop function (by the system). Basically the logic is that a new square is only drawn when the elapsed time since the last square is at least 1 second, so we should see squares appearing at a rate of one per second.

   This will require code like the following:

```
if ((currentTime-previousTime)>1000) {          // Time is in milliseconds
    draw();
}
```

   Note: This is not all the code needed, there is at least one line missing.

3. Download gameLoopB.html and complete the following exercises.

   a) Explain the benefit of the width and height variables and where do they get a starting value from?

   b) Note that currentTime is not used in this program. Explain how the setInterval() function replaces it.

c) Create a new function to contain the canvas drawing code. Use the following function definition.

```
function drawCurve(sx,sy,ex,ey,rx,ry) {


}
```

Move all the code that actually draws the curve into this function. The code to move is shown below.

```
// Set the colour for the drawing
context.strokeStyle = "red";
// Now draw the curve
context.beginPath();
context.moveTo(startX, startY);
context.quadraticCurveTo(refX, refY, endX, endY);
context.stroke();
```

You will need to change variable names to make this work.

Don't forget to call the new function in place of the code that was moved.

Make sure this step is working before moving on to the next step.

d) Add another function as shown below.

```
function rnd(max) {
    // Generate a random value up to the max allowed
    const value = Math.floor(Math.random() * max);
    return value;
}
```

Use this function to replace **all the lines** that use the Math functions. For example, this line needs to be replaced - startX = Math.floor(Math.random() * width);

Make sure this step is working before moving on to the next step.

e) Do the curves keep drawing when the screen is minimized? Explain your answer.

4. Continuing with gameLoopB.html. Copy all the JavaScript code into a separate file called app.js.

   Remove all the script section from the head area of the html. Reference the javascript file by changing the html file as follows.

   ```
   <body>
       <canvas id="canvas" width="800px" height="800px"></canvas>
       <script src="app.js"></script>
   </body>
   ```

   Notice that the onload=”init();” has been removed also, so the init() function needs to be called in the Javascript file (app.js). Check the recent three.js projects for a reminder of how and where this is done.

   Make sure this step is working before moving on to the next step.

5. Move the rnd function into a separate file, which can be called rnd.js, but doesn’t have to be.

   Refer to the loadAvatar() function being used with three.js for hints on how to now include an import statement in the app.js file and an export statement in the rnd.js file to get this working again.

   Also, because import and export statements are being used, the script area in the html has to change to include type=”module”, as follows.

   ```
   <body>
       <canvas id="canvas" width="800px" height="800px"></canvas>
       <script src="app.js" type=”module”></script>
   </body>
   ```

6. As an extension, try to move the drawCurve function into the rnd.js file as well and then import that into app.js too. It is possible to have many imports and exports from the one file, in fact this is common. Probably the name of the file would be changed from rnd.js to something like utilities.js to better reflect the multi-purpose use of the file, but once again this is not necessary.