Programming Assignment-1 Report
Abhinav Kamath(IMT2019001)

- ## Summary

  In this programming assignment, I have implemented an application which allows the users to play Tangram.

  - A correct configuration using the shapes appears on the left hand side. The same shapes appear on the right hand side in their exploded state i.e. random positions and orientations.
  - In mode 0, no change can be made
  - In mode 2, the shapes on the right hand side can be moved. The shape closest to a left mouse click is chosen and is selected for movement. The selected shape can be moved left, right, up down by pressing ArrowLeft, AtrowRight, ArrowUp, ArrowDown button respectively. The image scan also be scaled up and down by pressing +/- buttons. Rotated left/right by pressing (/) buttons.
  - In mode 2 all the images can be translated , rotated, scaled together as in mode 1, using the same buttons.
  - In mode 3, the right box is emptied.
  - On moving from mode 3 to mode 1 the shapes get rendered again in their exploded state.

- ## Questions posed in the assignment

  - **What is the difference in the implementation of the two methodologies mentioned in C.2. a. and C.2.b.? (Hint: how would the transformation matrices for the primitives be managed?)**

    Answer - In C.2.a we would not have to update the coordinates of the primitives if there are individual model view matrices and that too would only be updated once until the selected primitive reaches the final position. Whereas in C.2.b the modelvalue matrices must be reset to their default values and the coordinates of the primitives must be updated every time they are moved.

  - **What API is critical in the implementation of “picking” using mouse button click?**

    Answer - All the primitives are added as objects into an array whenever they are drawn on the canvas in mode 0. Therefore in mode 1, the click coordinates which are obtained using the function (DOM).addEventListener() are stored in a variable and then I calculate the distance of the centroid of each of the primitives from the click coordinates. The primitive, whose centroid is at the least distance from the click coordinates, is then ”picked”.

- **What would be a good alternative to minimize the number of key click events used in this application?**
  Answer - in mode 1, the application goes into 1 ”move” mode and a random particular primitive is highlighted and then translated, rotated and scaled using the usual keys. But once in ”move” mode, keys such as ”a”, ”s”, ”d” and ”w” can be used for navigating among the primitives on the canvas. The primitive selected will be highlighted and then moved by using the usual arrow keys and scaled using ”+/-” keys and rotated by pressing (/) keys . This reduces the number of clicks as the primitives on the screen can be traversed through the mentioned keys and no need to be manually clicked on.
- **Why is the use of centroid important?**
  Answer - The operations of scaling and rotating are done about a point which is the origin of the canvas. So by translating the centroid of the primitive to the origin of the canvas helps in scaling and rotation of the primitive properly

- Sources
  - https://javascript.info/keyboard-events - For the keypress events.
  - https://github.com/Amit-Tomar/T2-21-CS-606/tree/main/src/example5 - Examples from the tutorial class
  - https://github.com/davidwparker/programmingtil-webgl
  - https://www.youtube.com/watch?v=oDiSqQT_szo&list=PLPqKsyEGhUnaOdIFLKvdkXAQWD4DoXnFl&ab_channel=DavidParker