

# CS606 Computer Graphics

## Assignment 1

### 2-Dimentional Tangram Puzzle Game

*~Adrij Sharma*

**IMT2019004**

#### ANSWERS

1. Differences:

It is mentioned in C.2a that, you need not visit the primitive again if you have already worked with it. This implies that once we are done with the manipulation of the values associated with a primitive, we can safely ignore that primitive. Therefore, in the implementation of C.2a, a single transformation matrix for all primitives is enough. First, we transform the given primitive with the single transformation matrix, then reset the transformation matrix to an identity matrix so that it can be used for updating the upcoming primitive. The transformation matrix of a shape is updated only when that primitive is visited during the time the primitive is undergoing any kind of motion. Since a shape cannot be visited twice at the same time, the transformation matrix will not undergo any change from the perspective of that primitive.

On the other hand, in C.2b, there is a need for maintaining more than one transformation matrices for transformation of different primitives simultaneously. A particular primitive can be brought under motion any number of times when the game is in mode-1, therefore, we need to keep updating the transformation matrices have to be updated for the entire duration of mode 1.

2. Javascript HTML DOM EventListener API, which can be used by calling the `addEventListener()` method, is a critical API in the implementation of “picking” using mouse button click.
3. The number of keyboards clicks can be reduced by increasing the functionalities which can be the performed by mouse click and drag.
- There is no need for multiple modes – Just 2 modes are enough, namely, Game-On and Reset-Game. This reduces the need for pressing “m” for changing modes more frequently.
  - Mouse can be used for all the translation instead of keyboard arrow keys. Even diagonal translation can be made in this case.
  - Mouse scroller can be used for zoom in and zoom out.
  - Even the rotation and proportion can be done with buttons displayed on the screen for these purposes instead of keyboard press.
  - Click, drag and select by mouse can be used to bind the primitives together and do the desired motion on the bound primitives. This eliminates the need for mode-2. Also, in mode-2 we can only select all the primitives and bind them together, whereas, in this approach, we can select any number of primitives and bind them together for a combined motion.

4. While scaling and rotating a primitive, the origin gets translated to centroid of the primitive. Once the rotation or the proportion is complete, the original origin is restored. Another choice could have been the incentre in case of triangles. But we cannot use the circumcenter, because in many cases the circumcenter can be outside the triangle and we want the triangle to rotate about the Z-axis passing through a point which is inside the triangle. In mode 2, centroids of individual primitives rotate about the centroid of the bound primitives behaving as a single entity and the primitives rotate about their individual centroids. Scaling of the bound primitives in mode 2 involves the relative motion of centroid of each primitive and the centroids and the centroid of the united entity of primitives. Therefore, centroid plays a pivotal role.

## **CODE SPECIFICS**

1. R0 and R1 reference areas are 2 different scenes using 2 different renderer objects and 2 different animation functions. The shapes used inside are also completely different because R1 is completely independent of R0 and the shapes in R0 remain fixed.
2. Translation, rotation and proportion of primitives is implemented through the functions of same name present in the transform class.
3. After each motion, updation of the centroid of the primitive and the vertices of primitive takes place through functions of the same names present in the Triangle and Parallelogram classes.
4. Parallelogram class constructor takes only 4 vertex positions v1, v2, v3, v4 as parameter (a 4-sided figure has 4 vertices) and draws 2 triangles with v1, v2, v3 and v1, v4, v3 as the vertices respectively.
5. For keeping the primitives away from the boundaries of R0 and R1 reference areas, map function has been used to scale down the vertex positions.
6. "bindPrimitivesTogether()" function implements the task of finding a common centroid of all the primitives in mode-2.

## **REFERENCES**

1. <https://github.com/Amit-Tomar/T2-21-CS-606>
2. <https://github.com/invent-box/Learn-WebGL>
3. [https://www.youtube.com/watch?v=-nqIzV9OuQM&list=RDCMUQRQCzMftIUEIX03kHjV4rmQ&start\\_radio=1](https://www.youtube.com/watch?v=-nqIzV9OuQM&list=RDCMUQRQCzMftIUEIX03kHjV4rmQ&start_radio=1)
4. <https://www.youtube.com/watch?v=-nqIzV9OuQM&t=810s>
5. <https://glmatrix.net/docs/index.html>
6. <https://www.youtube.com/watch?v=Y5eHAfuO9aY&t=491s>
7. <https://webglreport.com/>
8. <https://marketplace.visualstudio.com/items?itemName=ritwickdey.LiveServer>