

COMPUTER GRAPHICS

Assignment-1 Report

Samaksh Dhingra (IMT2019075)

February 18, 2022

1 ANSWERS TO GIVEN QUESTIONS:-

Q1. What is the difference in the implementation of the two methodologies mentioned in *C.2. a.* and *C.2.b.* (Refer problem statement)?

Ans.

Primitives are stored as objects of some shape class. Each object uses 3 memory spaces to store information about 'translation', 'rotation angle' and 'scaling' taken place till current time. Using these memory spaces we create a transform matrix before each time we render and send it to shader along with initial positions of the primitive.

In *C.2.a.* methodology, once a position of primitive is finalized, it is not selected and changed afterwards. Hence, we can just save the transformation matrix of the final position of the finalized primitive and reuse the memory spaces(for translation,rotationAngle,scaling) for next primitive.

In *C.2.b.* methodology (also the methodology I implemented), since all the primitives can be picked any time and transformed, we have to keep separate independent memory spaces for each primitive that keeps track of the overall translation, rotation and scaling for each primitive. So, this will require extra memory.

Although, we saved memory in implementing with methodology *C.2.a.*, but we also lose the ability to select and transform multiple primitives with liberty.

Q2. What API is critical in the implementation of “picking” using mouse button click?

Ans.

'Document' and 'Window' APIs are the most critical API used in the implementation of "picking" using mouse button click.

Specifically, *addEventListener()* method of one of these APIs is used to detect the key/mouse clicks. Also it tells the various information about the click like position on screen where the mouse is clicked. As we are picking the object based on distance of position mouse is clicked and the centroid of primitive, it plays very important role in "picking".

'Document' API is used if a click has to be detected on a document.

'Window' API is used if a click has to be detected inside the window of which document is part of.

Q3. What would be a good alternative to minimize the number of key click events used in this application? Your solution should include how the mode-value changes are incorporated.

Ans.

In the current implementation of this application, we are:-

- Selecting primitive one by one with mouse click.
- Keeping separate key click for motion in each direction, separate key click for rotation in different directions.
- Keeping a separate mode, where we transform group of primitives which itself requires separate clicks for each transformation in different directions.

Hence, to reduce the number of click events, we can alternatively use following features for the same:-

- Performing transformation of primitive with help of 'Mouse Drag'. With just 1 click event of mouse, we would be able to transform primitive in any direction.

- Selection of multiple primitives with help of 'Mouse Drag' and performing the transformation on this selected primitives with respect to the bounded rectangle of these with help of 'Mouse Drag'. This helped removing one full Mode from the application which thereby saved a lot of key clicks(that were being done in that mode). It also gives user to select custom set of primitives to perform transformation on collectively instead of only ability to transform on whole set in same number of clicks.
- Instead of keeping the clear state, we can directly reset to Mode 1. (Saving clicks done to change states)

Since the number of modes are reduced to 2 after this solution, mode-values now are only 0 or 1.

Q4. Why is the use of centroid important in transforming a primitive or a group of primitives?

Ans.

According to physics laws of rotation, when an object is rotating (or scaling), the position of points that lie on the axis of rotation (or scaling) does not change.

In the problem statement, it is asked to perform scaling and rotation of a primitive along its centroid. Hence during rotation and scaling, centroid of the primitive or group of primitive does not change.

The final position of point we get after rotating or scaling from the initial position point in a coordinate system, it is algebraically easier to find it with respect to origin. So if we have to find these transformations with respect to centroid, we have to first change the frame of reference where centroid is the origin.

Logically, we do this by:-

- First, translating the object such that the centroid lies on origin. (Changing frame of reference)
- Rotate or Scale in this position (Performing calculations in new frame of reference).

- Translate back to original position (Changing back to original frame of reference).

Hence, the position of centroid plays important role in the transformation of primitives or group of primitives as we have the required translation for changing frame of references depends on those values.

2 REFERENCES USED

- a. Basic Template of fragment shader, vertex shader and different classes like Rendering, Shading, Shapes, Transformation are inspired from the TA's 'Example Code' uploaded on Github for reference of this course.
<https://github.com/Amit-Tomar/T2-21-CS-606>
- b. Basic concepts of WebGL, concepts, formulas used for Linear Algebra operations for transformations.
<https://webglfundamentals.org/>
- c. Syntax and concepts of javascript.
<https://developer.mozilla.org/en-US/docs/Web/JavaScript>