

# Abstract Artist Application : Report

---

Ellanti Bharath Sai - IMT2018022

February 2, 2021

## Abstract

2D interactive planar rendering with 2D translation, rotation, scaling/zooming.

## 1 Approach

- There is a mode variable which keeps track of mode. When "m" is clicked on the keyboard, mode value is increased by 1 and mod of 3 for this updated value is then sent to the mode variable and set the scene according to mode variable.
- If the mode value is '0' then user can press 'r' to draw rectangle and 's' to draw square. When the mouse clicks on the canvas corresponding mouse click points are stored in a variable and then use these coordinates as center for primitive to be drawn. A primitive object is created based on the user input for the shape with center coordinates as parameters. These new objects are then pushed into an array of primitives.
- To create a shape of square in mode '0'. The square object will create four points from the center in constructor each point is equidistant from center along negative and positive x and y axes. Similarly for rectangle four points are created with 1:2 ratio of length and breadth. These points are then sent into vertex shader.
- In vertex shader we plot these four vertices by doing by transforming the centroid of the object to origin and rotating the object by specified rotation angle and then shifting the centroid from origin to original centroid. By doing this we can do rotation of an object around fixed centroid without moving the centroid.
- Color of Square is fixed to RGB [1 0 1] and rectangle is fixed to RGB [1 0 0]. If an object is selected then its color changes to black RGB [0 0 0] and retains its original color if deselected.
- In mode 1 the transformations are made by updating the vertices of the object by some amount. While doing transformations the center of the object also changes. While doing scaling center remains same.
- In mode 2 the centroid of the whole scene is calculated by taking the min of x and y, max of x and y for all vertices present in primitive array. Then we assume a rectangle with these points and finds the centroid of this rectangle. Then we update origin of each object in the primitive array as this centroid.

## 2 Questions

- 1. For transforming a matrix up or down or left or right, I updated the center coordinates of object in index file necessarily and respectively update the center coordinates in respective primitive object and in the draw function of primitive vertex attributes are computed again with new center coordinates and size if changed to do scaling.
- 2. To pick a element on canvas using mouse clicks I first get the coordinates of mouse click and then search if this point lies inside the region of primitive for every primitive in primitive array. To check if a point lies inside a square or rectangle the x value of the point should lie between max x and min x of primitive and similarly for y. But by doing this I got a problem if two primitives gets overlapped and I point on the region that belongs to both primitives then both primitives are getting selected. So to avoid this problem after checking if the point lies inside the region, If there two primitives like above then I select the primitive that has min distance from the center of primitive.
- 3. To reduce number key clicks we could implement another key click that could move the mode backwards so that we can move forwards and backwards in changing mode whereas now we need to go till end of the mode to again come to starting mode.
- 4. The centroid of an object plays an important role in rotation and scaling. While doing scaling if centroid moves from its original position then it looks like the object is scaling and moving in the canvas. While doing

rotation of an object if we don't do rotation around centroid then the object rotates around the origin, while doing rotation around origin what happens is we rotate the x and y axes in some desired angle, by doing this the vertex coordinates of the object changes and it looks like the object is moving in circle, but this is not the desired way. So we need to rotate an object around its centroid. While doing rotation around centroid we first move the centroid of the object to origin and rotate in desired angle and then move the centroid to original position so that centroid remains same and object rotates at some given angle.