

Computer Graphics Assignment 1

Naveen Kumar V R

IMT2017029

International Institute of Information Technology

naveenkumar.vr@iiitb.org

January 2021

1 Goal

2D interactive planar rendering with 2D translation, rotation, scaling/zooming. The program has three modes:

- ◇ Drawing
- ◇ Instance-transformation
- ◇ Scene-transformation

2 Drawing

Mode 0 is called drawing mode. In this mode, user can click anywhere on the canvas and a shape appears there with the clicked point as it's centroid. There are three shapes, rectangle, square and circle and it is chosen by keyboard commands 'r' for rectangle, 's' for square and 'c' for circle.

2.1 Working

- ◇ Initially mode is 0 and default shape is rectangle.
- ◇ When an user presses the key 'r', 's' or 'c' on the keyboard, the keyboard event is captured and the shape is changes accordingly.
- ◇ When the user clicks, this event is captured by mouse event listener and the coordinates of the click is converted to coordinate system of the canvas using the function called mouseToClipCoord.
- ◇ Origin is initialised as centroid for all shapes with defined size, sent to the vertex shader. Each shape has it's own color which is sent to fragment shader.
- ◇ The selected shape is created and translated to the location of the click, when user clicks or selects a coordinate on the canvas.
- ◇ When mode 0 is active, user can create how many ever shapes they want on the canvas.
- ◇ When user wants to move to next mode (mode 1), the key 'm' has to be pressed.

3 Instance Transformation

Mode 1 is called Instance Transformation mode. In this mode, the user can click anywhere on the canvas and the closest shape is chosen from the point/coordinates clicked. After the selection of the object, we can perform three operations to that shape, translation scaling and deletion. For translation (moving), the selected object can be moved around using the arrow keys. For scaling, the selected object can be scaled bigger or smaller using '+' or '-' keys. For deletion the selected object can be deleted by pressing 'x' key.

3.1 Working

- ◇ Every shape has its translation matrix, scaling matrix and rotation matrix which is updated and used for making the Transformation Matrix for that object.
- ◇ In this mode, the arrow keys, '+', '-' and 'x' keys are listened.
- ◇ When the user clicks anywhere on the canvas, the event is captured by mouse event listener and the coordinates of the click is converted to coordinate system of the canvas.
- ◇ Now the object closest to this system coordinates is calculated and selected.
- ◇ When the user wants to move the object, the user has to press any of the arrow keys. The selected object's translation matrix is changed by 0.04 accordingly in the direction of the pressed arrow key.
- ◇ When the user wants to scale the object, '+' or '-' key is pressed. And the shape vector of the selected object is sized accordingly by 4 percentage in all direction.
- ◇ The shapes are being rendered every time, their position matrix is multiplied with their recently updated transformation matrix.
- ◇ To delete the selected object, the user has to press 'x', which will remove the shape from the primitive list.
- ◇ While this mode is active, user can select any object and perform transformations on it.
- ◇ To go to the next mode (mode 2), press 'm'.

4 Scene Transformation

Mode 2 is called Scene Transformation mode. In this mode, we rotate the whole scene around the centroid of the bounding box. Bound box is considered as enclosed shape of all the shapes on the canvas.

4.1 Working

- ◇ In mode 2, the left arrow (anti-clockwise) and right arrow (clockwise) are going to be used for rotation.
- ◇ First, when we come to mode 2, the bounding box is found and it's centroid is calculated.

- ◇ Now all objects are rotated around the bounding box centroid.
- ◇ This done by translating the centroid of each shape with respect to the centroid of bounding box and then shape is rotated by the same angle as the scene.
- ◇ When 'm' key is pressed the mode is changed to 0, and objects gets back to their previous position before rotation.

5 Key Concepts

5.1 Transformation Matrix

Each object/shape has stored their translation matrix, scaling matrix and rotation matrix. These three matrices are multiplied to form the transformation matrix. The shape and position of the object appearing on the canvas is the product of the transformation matrix and the position matrix.

5.2 Event listeners

This event listener is responsible for both mouse click and key-press. Every action done by the user on the canvas are listened by event listener. We use these actions to get to communicate with our program.

5.3 Bounding Box

This is used in mode 2 (screen transformation mode), where we rotate the screen. Here, the scene is the bounding box which is tightly enclosed of all the objects on the canvas. This is calculated by finding the minimum and maximum x and y values of among all objects. The centroid is the average of minimum and maximum of the values.

6 Questions

1. How did you program separate transformation matrices for all the object instances (primitives), and the scene?

Ans: Each object has its instance of transform class which keeps tracks of its transformation matrix. Translation, scaling and rotation matrix information are held by each object's transformation matrix. So when the scene is rotated, transformation matrix of each objects on the screen are changed accordingly (translation with respect to bounding box centroid + rotation angle).

2. What API is critical in the implementation of "picking" using mouse button click?

Ans: The click event is used to get the coordinates of the point where the mouse click is clicked. We use this to create objects on that point/coordinate and use to select the closest already existing primitive on the canvas.

3. 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: We can keep check boxes for selecting the mode. So, we give more check boxes to select the shape in mode 0. We can give sliders for translation and scaling values in mode 1. We can give slider for rotation angle in mode 2.

4. Why is the use of centroid important? (Hint: Consider it with respect to rotation and scaling.)

Ans: The centroid is important because,

- a) In scaling, we need to maintain the centroid, so that the shape is equally scaled in all sides.
- b) In rotation, we rotate an object with respect to its centroid.
- c) In translation, it is responsible for moving the centroid and if rotation or scaling doesn't maintain the centroid it will cause problems in translation.