Drawing Editor - Object Oriented Drawing Editor with Typescript

* This project is an implementation of drawing editor using object oriented paradigm.
* This program is written with vue + typescript. It uses classes to represent objects.

**Requirements**

* There shall be three shapes in this program: Circle, rectangle, line.
* Users shall be able to move circles, rectangles.
* Users shall be able to draw all the shapes.
* Users shall be able to delete all circles.
* Users shall be able to delete all rectangles.
* Users shall be able to delete all lines.
* Users shall be able to change background color of the canvas.
* Users shall be able to change width and height of the canvas.
* The shapes shall stay in the canvas after moving them.
* Shapes shall be able to collide each other.
* There shall be a class Shape that has
  + name of the shape - string
  + draw() method that draws the shape into canvas.
* Line class shall have
  + x1 - real number
  + x2 - real number
  + y1 - real number
  + y2 - real number
* Rectangle class shall extend Shape class and shall have
  + x1 - real number
  + y1 - real number
  + x2 - real number
  + y2 - real number
* There shall be a Circle class that has
  + centerX - real number
  + centerY - real number
  + radius - real number
* Canvas class shall have
  + width - integer
  + height - integer
  + backgroundColor - html hls color value as string

**Stages of Development**

First, I declared my program’s requirements as written in above statements.

Second, I created Canvas component in vue and gave it the attributes that are defined above. I gave a mouse click event listener to the canvas that I define in main component. This function listens the clicked position and dependent on the operation mode of the canvas something happens. This function is my main function.

Third, I created several operation modes. For each shape there is an operation mode for the creation of the shape. The descriptions of how to add a shape is shown when hovering over the button associated with a shape.

Fourth, I added shape creation codes. For example, in order to add a circle, two points are chosen, the center point and a point on the border.

Fifth, I added the drag functionality by adding mouse up and mouse move event listeners to all shapes.

Sixth, I added canvas background color and size changer widget. I made the widget hidable so that users are not distracted with the widget after their operation is done.

**Conclusion**

I learned a lot of things in this project.

* Designing user interfaces object oriented is a good idea since it saves you code.
* Making user interface simple and having tutorials is important.
* Making menus hidable is also a good idea to keep user inside of the environment.

Thanks

Serkan Özel

2015400123

serkan.ozel@boun.edu.tr