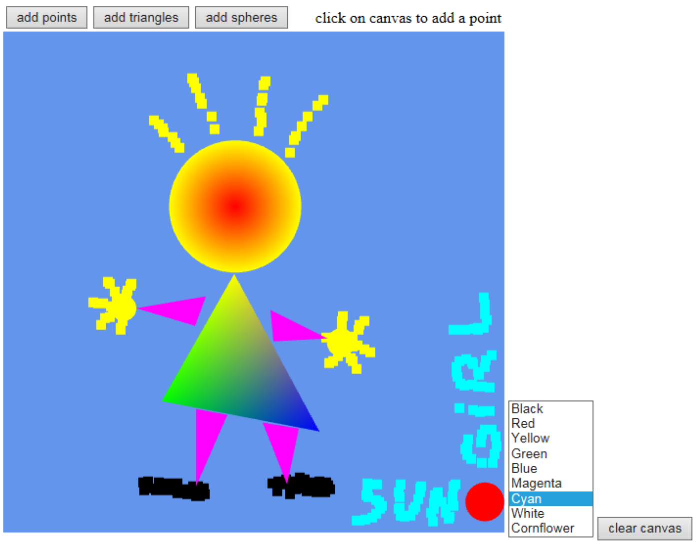
02561 COMPUTER GRAPHICS DTU COMPUTE

Worksheet 2: Input devices and interaction

Reading	Angel: Sections 3.2-3.11
Purpose	The purpose of this set of exercises is to start interacting with the graphics elements that we draw using input devices. You will make a small 2D drawing program where primitive shapes of different colors can be added using the mouse. Although it is possible to draw 2D shapes using a 2D canvas context, you will do it with a WebGL context as it is really not much harder, and WebGL enables later extension to full 3D. The following screenshot is an example of what the final web application could look like. Your version doesn't have to be exactly like it. The important point is that it has the features we want.  add points add triangles add spheres click on canvas to add a point SUN GiRL Black Red Yellow Green Blue Magenta Cyan White Cornflower clear canvas
Features	In your drawing program, you should be able to: • Use the mouse to draw different primitive shapes (point, triangle, and circle). • Select the color of a shape while adding it. • Draw points at the first two vertices of a triangle and replace the points by a triangle for every third vertex. • Draw a point in the circle center and replace the point by a circle for every second mouse click (the second point sets the circle radius). • Use color interpolation between triangle vertices and radially in circles. • Clear the canvas to the selected color (delete all shapes). In the following, we divide the development of the application into parts.

02561 COMPUTER GRAPHICS DTU COMPUTE

Worksheet 2: Input devices and interaction

Part 1	• Start from your solution to Part 2 of Worksheet 1: A web application that clears the canvas and then draws three points. [Angel 2.8] • Attach an event handler to the mouse click event and draw points on the canvas where the mouse was clicked. [Angel 3.7] • Points are offset from the tip of the mouse cursor. This is not as desired. Get the bounding rectangle of the canvas in the client area using `event.target.getBoundingClientRect()` and correct the mouse position using the `left` and `top` coordinates of this rectangle.
Part 2	• Add a button that clears the canvas. [Angel 3.6.2] • Add a color selection menu where you can choose the color to be used when clearing the canvas. [Angel 2.5, 3.6.3, 3.10] • Add a color selection menu to set the color of points drawn when clicking the mouse. This requires updating your shaders to work with colors as in Part 3 of Worksheet 1.
Part 3	• We would now like to have two different drawing modes. One where we draw points and one where we interactively build a triangle by placing three points. Add a button for each drawing mode. [Angel 3.6.2] • (Hint) What we do has some relation to the textbook CAD example [Angel 3.10], where a polygon is built interactively. • Let us draw all our shapes as triangles (using `gl.TRIANGLES`). When a point is drawn, add vertices (positions and colors) for two triangles representing this point to the vertex buffers. In the triangle drawing mode, keep a record (array) of the former points that were clicked and their colors. When the third point is clicked, replace the two points and their colors (four triangles) with the one triangle to be drawn and clear the record.
Part 4	• Add a button for a circle drawing mode. [Angel 3.6.2] • When drawing in circle mode, add a point on the first click and keep a record of the point that was clicked. On the second click, use the position of the point on record and that of the second click to find the circle center and radius and replace the point with vertices (positions and colors) for the circle (refer to Part 5 of Worksheet 1). If needed, modify your circle drawing routine so that the circle can be drawn as triangles (using `gl.TRIANGLES`).