```
/**
 * Clickable Button
 * Includes all functions from interactive object
 * @param {number} x  x
 * @param {number} y  y
 * @param {number} w width
 * @param {number} h height
 * @param {string} fill fill colour
 * @param {string} over hover over  colour
 * @param {string} selected button has been clicked colour
 * @param {string} stroke stroke colour
 * @param {string} text  button text
 * @param {string} text  button text colour
 */
class ButtonBase extends InteractiveObject{
    constructor(x,y, w, h, fill, over, selected, stroke, text, textColour){
        super()
        this.x = x;
        this.y = y;
        this.w = w;
        this.h = h;
        this.fill = fill;
        this.over=over;
        this.selected = selected;
        this.currentFill = fill
        this.stroke=stroke;
        this.text = text;
        this.textColour = textColour
        this.inBounds = false
    }
    mUp(e) {
        super.mUp(e);
        console.log("up call from button")
    }

    update(){
        this.inBounds = this.getBoundary(this.x, this.y, this.w, this.h, this.xMouse, this.yMouse)
        let fill = this.currentFill
        this.draw(this.x, this.y, this.w, this.h, fill, this.stroke, this.text, this.textColour )

        if(this.inBounds){
            this.currentFill = this.over
        }else{
            this.currentFill = this.fill
        }

    }
    mClick() {}

    draw(x,y, w,h,fillColour,strokeColour, txt, txtCol){
        this.strokeFillRect(x,y,w,h,fillColour, strokeColour, 0.5)
        this.centredText(txt, x+w/2,  y+h/2, txtCol)

    }
}
ButtonBase.prototype.strokeFillRect = strokeFillRect
ButtonBase.prototype.centredText = centredText


/**
 * Basic, selectable options button
 * Includes all functions from interactive object and button base
 * @param {number} x  x
 * @param {number} y  y
 * @param {number} w width
 * @param {number} h height
 * @param {string} fill fill colour
 * @param {string} over hover over  colour
 * @param {string} selected button has been clicked colour
 * @param {string} stroke stroke colour
 * @param {string} text  button text
 * @param {string} text  button text colour
 */
class OptionsButton extends ButtonBase{
    constructor(x,y, w, h, fill, over, selected, stroke, text, textColour) {
        super(x,y, w, h, fill, over, selected, stroke, text, textColour);
    }
    mClick() {

        if(this.inBounds){
            console.log("Click call from button:" + this.text)
            OptionsButton.selected = this;
        }
    }
    update(){
        super.update()
           if(OptionsButton.selected === this){
               this.currentFill = this.selected
           }
    }
}
OptionsButton.selected = null;
```

```

// noinspection DuplicatedCode
class SwatchButton extends InteractiveObject{
    constructor(x,y,r,fillColour, strokeColour, strokeOverColour, strokeWidth) {
        super()
        this.x = x;
        this.y = y;
        this.r = r;
        this.fillColour = fillColour;
        this.strokeColour = strokeColour;
        this.strokeOverColour = strokeOverColour;
        this.strokeWidth = strokeWidth;
        this.inBounds = false
    }
    mClick(e) {
        super.mClick(e);
        if(this.inBounds){
            SwatchButton.colour = this.fillColour

        }
    }

    update(){
        this.draw()
        this.inBounds=this.getDistance(this.x, this.y, this.xMouse, this.yMouse, this.r)
    }
    draw(){
       let currentStroke = this.strokeColour
        if(this.inBounds){
            currentStroke = this.strokeOverColour
        }
        if(this.fillColour === SwatchButton.colour){
            currentStroke = this.fillColour
        }
        this.filledStrokedCircle(this.x,this.y,this.r, this.fillColour, currentStroke, this.strokeWidth)
    }

}
SwatchButton.prototype.filledStrokedCircle = drawFilledStrokeCircle
SwatchButton.colour = "rgb(0,0,0)"
```

Listing 1: Buttons