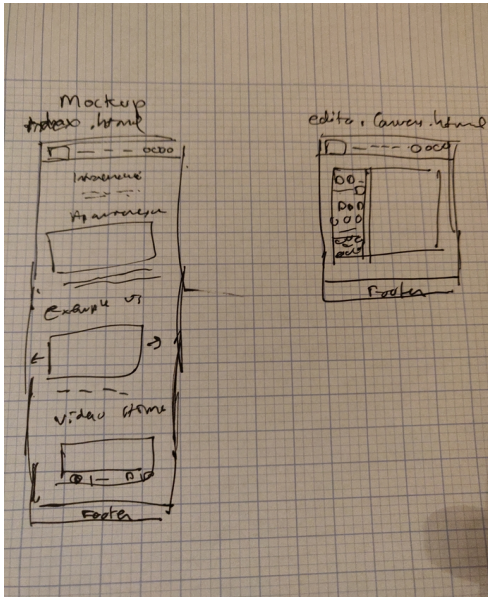


Mockup:



Documentació javascript:

Primer posem el event onload al window perque s'espera a que carregui la pàgina html i ja després comenci amb el script en qüestió

```
js > JS canvas.js > ...
1 //Al carregar el html es carrega aquest event amb la seva funcio corresponent
2 window.onload = function() {
3     var c = document.getElementById("canvas");
4     var ctx = c.getContext("2d");
5     var mouse = { x: 0, y: 0 };
6     var herramienta = "pincel";
7     var tamlinea = document.getElementById("rangoInput").value;
8     var tamFigura = document.getElementById("rangoInputFigura").value;
9
10    function pintar() {
11        pintarColor();
12        switch (herramienta) {
13            case "pincel":
14                c.addEventListener("mousemove", pinselada, true);
15                break;
16            case "circulo":
17                pintarCirculo();
18                break;
19            case "imagen":
20                ponerImagen();
21                break;
22            case "cuadrado":
23                cuadradoPrueba();
24                break;
25            case "rectangulo":
26                pintarRectangulo();
27                break;
28            case "triangulo":
29                pintarTriangulo();
30                break;
31            case "goma":
32                c.addEventListener("mousemove", goma, true);
33                break;
34            case "texto":
35                pintarTexto();
36                break;
```

```

37         case "color":
38             pintarColor();
39             break;
40         case "clear":
41             clearCanvas();
42             break;
43         default:
44             alert("YO NO PINTO NI MIELDA");
45             break;
46     }
47 }
48

```

Carreguem en una variable canvas, el context, el mouse, la eina i els rangs.

En la funció pintar mirem amb un switch quin quina eina estem utilitzant en qüestió i cridem a la funció que es necessita.

```

//cada cop que movem el mouse es truca all callback obtenirCoordenades
c.addEventListener("mousemove", obtenirCoordenades, true);

c.addEventListener("mousedown", () => {
    if (herramienta == "pincel") {
        ctx.beginPath();
        ctx.moveTo(mouse.x, mouse.y);
    }

    if (herramienta == "goma") {
        ctx.beginPath();
        ctx.moveTo(mouse.x, mouse.y);
    }

    pintar();
});

//Cuando levantamos el dedo del boton quitamos el listener de mousemove tanto del pincel como de la goma para que dejen de pintar/borrar
c.addEventListener(
    "mouseup",
    () => {
        if (herramienta == "pincel") {
            c.removeEventListener("mousemove", pinselada, true);
        } else if (herramienta == "goma") {
            c.removeEventListener("mousemove", goma, true);
        }
    },
    true
);

```

Cada cop que movem el mouse es truca a la funció obtenir coordenades.

Al estar el mousedown comprovem si la eina es el pinzell o goma, si es alguna d'aquestes iniciem el path i ens movem cap aquesta posició.

Al estar en mouseup mirem si l'eina es el pinzell o la goma si es alguna d'aquestes li treiem el mousemove amb el seu callback corresponent.

## FUNCIONES:

```
//obtenemos las coordenadas en el eje de las x e y
function obtenerCoordenades(evt) {
    var rect = c.getBoundingClientRect();
    mouse.x = evt.clientX - rect.left;
    mouse.y = evt.clientY - rect.top;
}

//funcion del pincel para pintar
function pinselada() {
    ctx.lineWidth = tamLinea;
    ctx.lineJoin = "round";
    ctx.lineCap = "round";
    ctx.lineTo(mouse.x, mouse.y);
    ctx.stroke();
}

//funcion de la goma para borrar
function goma() {
    ctx.lineWidth = tamLinea;
    ctx.lineJoin = "round";
    ctx.lineCap = "round";
    ctx.strokeStyle = "white";
    ctx.lineTo(mouse.x, mouse.y);
    ctx.stroke();
}
```

```
//cuadrado Al pulsar boton
function cuadradoPrueba() {
    //c.removeEventListener("mousedown", onPaint);
    ctx.lineWidth = tamLinea;
    ctx.lineJoin = "round";
    ctx.lineCap = "round";

    ctx.beginPath();
    ctx.moveTo(mouse.x, mouse.y);
    ctx.rect(mouse.x, mouse.y, tamFigura, tamFigura);
    ctx.closePath();
    ctx.stroke();
}

//circulo al pulsar el boton
function pintarCirculo() {
    ctx.lineWidth = tamLinea;
    ctx.lineJoin = "round";
    ctx.lineCap = "round";
    ctx.beginPath();
    ctx.moveTo(mouse.x, mouse.y);
    ctx.arc(mouse.x, mouse.y, tamFigura, 0, 2 * Math.PI);
    ctx.closePath();
    ctx.stroke();
}
```

```

//pinta un rectangulo al pulsar
function pintarRectangulo() {
    ctx.lineWidth = tamLinea;
    ctx.lineJoin = "round";
    ctx.lineCap = "round";
    ctx.beginPath();
    ctx.rect(
        mouse.x,
        mouse.y,
        parseInt(tamFigura) + parseInt(tamFigura),
        tamFigura
    );
    ctx.closePath();
    ctx.stroke();
}

//pinta un triangulo al pulsar
function pintarTriangulo() {

    ctx.beginPath();
    ctx.lineWidth = tamLinea;
    ctx.moveTo(mouse.x, mouse.y); //cogemos la posicion en x e y en la recta

    //TENIENDO EN CUENTA QUE LA RECTA ARRIBA A LA IZQUIERDA VALE 0,0 !
    ctx.lineTo(mouse.x - (parseInt(tamFigura) / 2), mouse.y + parseInt(tamFigura)); //desde la posicion x e y de antes, restamos en la posicion x
    //el tamaño de la figura y lo dividimos entre dos ya que hay dos
    //partes esta parte unicamente dibuja del punto inicial una linea hacia la izquierda
    //tambien usamos la posicion de y mas el tamaño de la figura para que haga la linea
    //hacia abajo
    ctx.lineTo(mouse.x + (parseInt(tamFigura) / 2), mouse.y + parseInt(tamFigura)); //hacemos lo correspondiente pero ahora sumandole el tamaño de la figura a x para
    //que se mueva hacia la derecha la linea
    ctx.closePath(); //con close path se cerraran las lineas y de esta forma se formara el triangulo
    ctx.stroke();
}

```

```

//funcion para pintar el texto que hayamos puesto en el input text
function pintarTexto() {
    var tamLetra = document.getElementById("selectTamLetra");
    var tamLetraSeleccionada = tamLetra.options[tamLetra.selectedIndex].text;
    var styleLetra = document.getElementById("selectFontStyle");
    var styleLetraSeleccionada =
        styleLetra.options[styleLetra.selectedIndex].text;
    var textoPersonal = document.getElementById("textoPersonalizado").value;
    ctx.font = styleLetraSeleccionada + " " + tamLetraSeleccionada + "px Arial";
    ctx.fillText(textoPersonal, mouse.x, mouse.y);
    ctx.stroke();
}

//limpia el canvas entero
function clearCanvas() {
    ctx.beginPath();
    ctx.clearRect(0, 0, c.width, c.height);
    ctx.closePath();
}

//cambia el color de las formas o del texto
function pintarColor() {
    var color = document.getElementById("botonColor");
    ctx.fillStyle = color.value;
    ctx.strokeStyle = color.value;
}

//funcion para guardar la imagen
function guardarComImatge() {
    document.getElementById("descargarImagen").download = "canvas.png";
    document.getElementById("descargarImagen").href = document
        .getElementById("canvas")
        .toDataURL("image/png");
}

```

```

//Esto hace que al cambiar el valor del input range del tamaño del pincel se nos muestre en un pequeño input text al lado
function mostrarValorRango() {
    var valorPincel = document.getElementById("rangoPincel").value;
    document.getElementById("rangoInput").value = valorPincel;
    tamLinea = valorPincel;
}

//Esto hace que al cambiar el valor del input range del tamaño de la figura se nos muestre en un pequeño input text al lado
function mostrarValorRangoFigura() {
    var valorFigura = document.getElementById("rangoFigura").value;
    document.getElementById("rangoInputFigura").value = valorFigura;
    tamFigura = valorFigura;
}

//funcion para volver la imagen a grises usando bitmap
function grayScale() {
    var imgd = ctx.getImageData(0, 0, c.width, c.height);
    var data = imgd.data;
    for (var i = 0; i < data.length; i += 4) {
        var grayscale = data[i] * 0.34 + data[i + 1] * 0.5 + data[i + 2] * 0.16;
        data[i] = grayscale;
        data[i + 1] = grayscale;
        data[i + 2] = grayscale;
    }
    ctx.putImageData(imgd, 0, 0);
}

```

```

//funcion para sacar el color negativo usando bitmap
function negativo() {
    var imgd = ctx.getImageData(0, 0, c.width, c.height);
    var data = imgd.data;
    for (var i = 0; i < data.length; i += 4) {
        //var grayscale = data[i] * 0.34 + data[i + 1] * 0.5 + data[i + 2] * 0.16;

        data[i] = 255 - data[i];
        data[i + 1] = 255 - data[i + 1];
        data[i + 2] = 255 - data[i + 2];
    }
    ctx.putImageData(imgd, 0, 0);
}

//funcion para cargar una imagen y ponerla en el canvas al hacer click
function ponerImagen() {
    var background = new Image();
    background.src = "img/j.png";
    background.onload = function() {
        ctx.drawImage(background, mouse.x, mouse.y, tamFigura, tamFigura);
    };
}

//funcion para rotar las figuras o el texto
function rotarFigura() {
    ctx.rotate(0.17);
    ctx.restore(); //dibuja la figura
}

```

Recuperem els elements del html5 i els hi afegim els listeners corresponents amb les funcions anònimes, en aquests casos dintre especifiquem quina eina es, perquè la funció que tenim al principi del script pugui treballar.

```
//RECUPERAMOS LOS ELEMENTOS DEL HTML5 Y LES AÑADIMOS SUS LISTENERS CORRESPONDIENTES
var botonPincel = document.getElementById("botonPincel").addEventListener(
  "click",
  () => {
    herramienta = "pincel";
  },
  true
);
var botonCirculo = document.getElementById("botonCirculo").addEventListener(
  "click",
  () => {
    herramienta = "circulo";
  },
  true
);
var botonImagen = document.getElementById("botonImagen").addEventListener(
  "click",
  () => {
    herramienta = "imagen";
  },
  true
);
var botonCuadrado = document
  .getElementById("botonCuadrado")
  .addEventListener("click", () => {
    (herramienta = "cuadrado"), true;
  });
var botonRectangulo = document
  .getElementById("botonRectangulo")
  .addEventListener("click", () => {
    (herramienta = "rectangulo"), true;
  });
var botonRotacion = document
  .getElementById("botonRotacion")
  .addEventListener("click", rotarFigura, true);
var botonTriangulo = document
```

```

  .getElementById("botonTriangulo")
  .addEventListener("click", () => {
    (herramienta = "triangulo"), true;
  });
var botonGoma = document
  .getElementById("botonGoma")
  .addEventListener("click", () => {
    (herramienta = "goma"), true;
  });
var botonTexto = document
  .getElementById("botonTexto")
  .addEventListener("click", () => {
    (herramienta = "texto"), true;
  }); //NOTA SI PONES UNA FUNCION CON PARAMETROS Y LA LL
var botonColor = document
  .getElementById("botonColor")
  .addEventListener("change", () => {
    (herramienta = "color"), true;
  });
var botonClear = document
  .getElementById("botonClear")
  .addEventListener("click", clearCanvas, true);
var botonGuardar = document
  .getElementById("descargarImagen")
  .addEventListener("click", guardarComImatge, true);
var botonGris = document
  .getElementById("botonGris")
  .addEventListener("click", grayScale, true);
var botonAlternarColor = document
  .getElementById("botonAlternarColor")
  .addEventListener("click", negativo, true);
var rangoPincel = document
  .getElementById("rangoPincel")
  .addEventListener("change", mostrarValorRango, true);
var botonFigura = document
```

```
var rangoFigura = document
    .getElementById("rangoFigura")
    .addEventListener("change", mostrarValorRangoFigura, true);
};|
```