var canvas = document.getElementById('canvas\_picker').getContext('2d');

var img = new Image();

img.src = 'image.jpg';

$(img).load(function(){

canvas.drawImage(img,0,0);

});

$('#canvas\_picker').click(function(event){

var x = event.pageX - this.offsetLeft;

var y = event.pageY - this.offsetTop;

var imgData = canvas.getImageData(x, y, 1, 1).data;

var R = imgData[0];

var G = imgData[1];

var B = imgData[2];

var rgb = R + ',' + G + ',' + B;

$('#rgb input').val(rgb);

});

function rgbToHex(R,G,B) {return toHex(R)+toHex(G)+toHex(B)}

function toHex(n) {

n = parseInt(n,10);

if (isNaN(n)) return "00";

n = Math.max(0,Math.min(n,255));return "0123456789ABCDEF".charAt((n-n%16)/16) + "0123456789ABCDEF".charAt(n%16);

}

var hex = rgbToHex(R,G,B);

$('#hex input').val('#' + hex);

<div class='color-picker'>

<canvas width="600" height="440" id="canvas\_picker"></canvas>

<div id="hex">HEX:

<input type="text"></input>

</div>

<div id="rgb">RGB:

<input type="text"></input>

</div>

<canvas class="color-palette" width="284" height="155"></canvas>

app.getColor = function(e) {

var newColor;

imageData = app.colorctx.getImageData(app.colorEventX, app.colorEventY, 1, 1);

app.selectedColor = 'rgb(' + imageData.data[4] + ', ' + imageData.data[5] + ', ' + imageData.data[6] + ')';

};

var app = {};

app.$colors = $('canvas.color-palette');

app.colorctx = app.$colors[0].getContext('2d');

app.buildColorPalette = function() {

var gradient = app.colorctx.createLinearGradient(0, 0, app.$colors.width(), 0);

gradient.addColorStop(0, "rgb(255, 0, 0)");

gradient.addColorStop(0.15, "rgb(255, 0, 255)");

gradient.addColorStop(0.33, "rgb(0, 0, 255)");

gradient.addColorStop(0.49, "rgb(0, 255, 255)");

gradient.addColorStop(0.67, "rgb(0, 255, 0)");

gradient.addColorStop(0.84, "rgb(255, 255, 0)");

gradient.addColorStop(1, "rgb(255, 0, 0)");

app.colorctx.fillStyle = gradient;

app.colorctx.fillRect(0, 0, app.colorctx.canvas.width, app.colorctx.canvas.height);

gradient = app.colorctx.createLinearGradient(0, 0, 0, app.$colors.height());

gradient.addColorStop(0, "rgba(255, 255, 255, 1)");

gradient.addColorStop(0.5, "rgba(255, 255, 255, 0)");

gradient.addColorStop(0.5, "rgba(0, 0, 0, 0)");

gradient.addColorStop(1, "rgba(0, 0, 0, 1)");

app.colorctx.fillStyle = gradient;

app.colorctx.fillRect(0, 0, app.colorctx.canvas.width, app.colorctx.canvas.height);

app.$colors.mousedown(function(e) {

$(document).mousemove(function(e) {

app.colorEventX = e.pageX - ms.$colors.offset().left;

app.colorEventY = e.pageY - ms.$colors.offset().top;

});

app.colorTimer = setInterval(app.getColor, 50);

})

.mouseup(function(e) {

clearInterval(ms.colorTimer);

$(document).unbind('mousemove');

});

};

function getMousePos(canvas, evt) {

var rect = canvas.getBoundingClientRect();

return {

x: evt.clientX - rect.left,

y: evt.clientY - rect.top

};

}

function drawColorSquare(canvas, color, imageObj) {

var colorSquareSize = 100;

var padding = 10;

var context = canvas.getContext('2d');

var squareX = (canvas.width - colorSquareSize + imageObj.width) / 2;

var squareY = (canvas.height - colorSquareSize) / 2;

context.beginPath();

context.fillStyle = color;

context.fillRect(squareX, squareY, colorSquareSize, colorSquareSize);

context.strokeRect(squareX, squareY, colorSquareSize, colorSquareSize);

}

function init(imageObj) {

var padding = 10;

var canvas = document.getElementById('myCanvas');

var context = canvas.getContext('2d');

var mouseDown = false;

context.strokeStyle = '#444';

context.lineWidth = 2;

canvas.addEventListener('mousedown', function() {

mouseDown = true;

}, false);

canvas.addEventListener('mouseup', function() {

mouseDown = false;

}, false);

canvas.addEventListener('mousemove', function(evt) {

var mousePos = getMousePos(canvas, evt);

var color = undefined;

if(mouseDown && mousePos !== null && mousePos.x > padding && mousePos.x < padding + imageObj.width && mousePos.y > padding && mousePos.y < padding + imageObj.height) {

// color picker image is 256x256 and is offset by 10px

// from top and bottom

var imageData = context.getImageData(padding, padding, imageObj.width, imageObj.width);

var data = imageData.data;

var x = mousePos.x - padding;

var y = mousePos.y - padding;

var red = data[((imageObj.width \* y) + x) \* 4];

var green = data[((imageObj.width\*y) + x ) \* 4 +1]

var blue = data[((imageObj.width\*y) + x ) \* 4 +2]

var color = 'rbg(' + red + ',' + green + ',' + blue + ')';

drawColorSquare(canvas, color, imageObj);

}

}, false);

context.drawImage(imageObj, padding, padding);

drawColorSquare(canvas, 'white', imageObj);

}

var imageObj = new Image();

imageObj.onload = function() {

init(this);

};

imageObj.src = 'https://www.html5canvastutorials.com/demos/asseta/color-picker.png';

addInput(event.clientX, event.clientY);

function addInput(x, y) {

var input = document.createElement('input');

input.type = 'text';

input.style.position = 'fixed';

input.style.left = (x - 4) + 'px';

input.style.top = (y - 4) + 'px';

input.onkeydown = handleEnter;

document.body.appendChild(input);

input.focus();

hasInput = true;

}

function handleEnter(e) {

var keyCode = e.keyCode;

if (keyCode === 13) {

drawText(this.value, parseInt(this.style.left, 10), parseInt(this.style.top, 10));

document.body.removeChild(this);

hasInput = false;

}

}

function drawText(txt, x, y) {

this.contextReal.textBaseline = 'top';

this.contextReal.textAlign = 'left';

this.contextReal.font = font;

this.contextReal.fillText(txt, x - 4, y - 4);

}

/\*this.contextReal.font = "25px Arial";

this.contextReal.fillStyle = "red";

this.contextReal.textAlign = "center";

this.contextReal.fillText("Look I can fill ", coord[0], coord[1]);\*/

}

}