

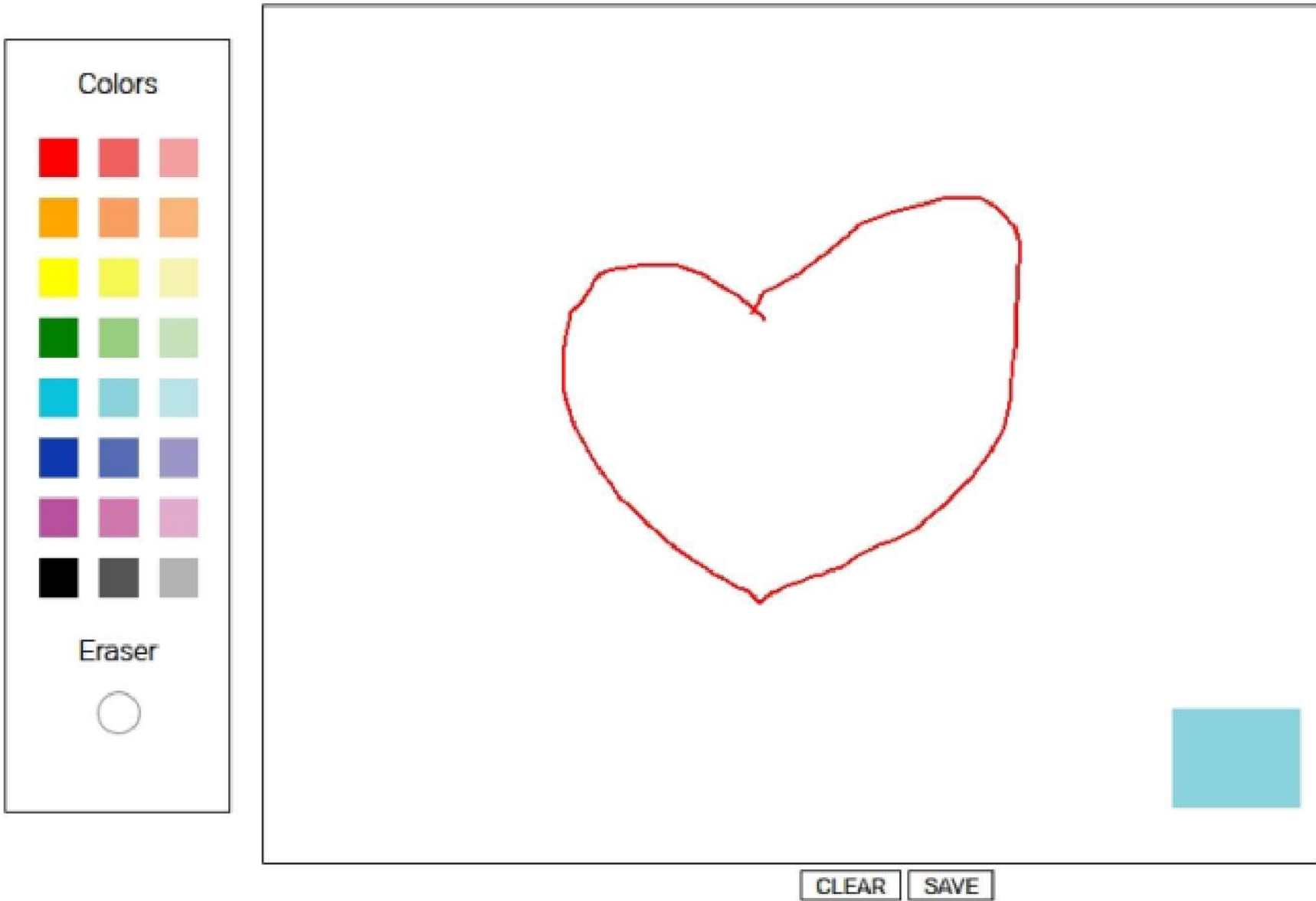
Article: Документация к проекту "Paint"

»» Андрей Огурцов ««
©»» 2021 «« Андрей Огурцов ««

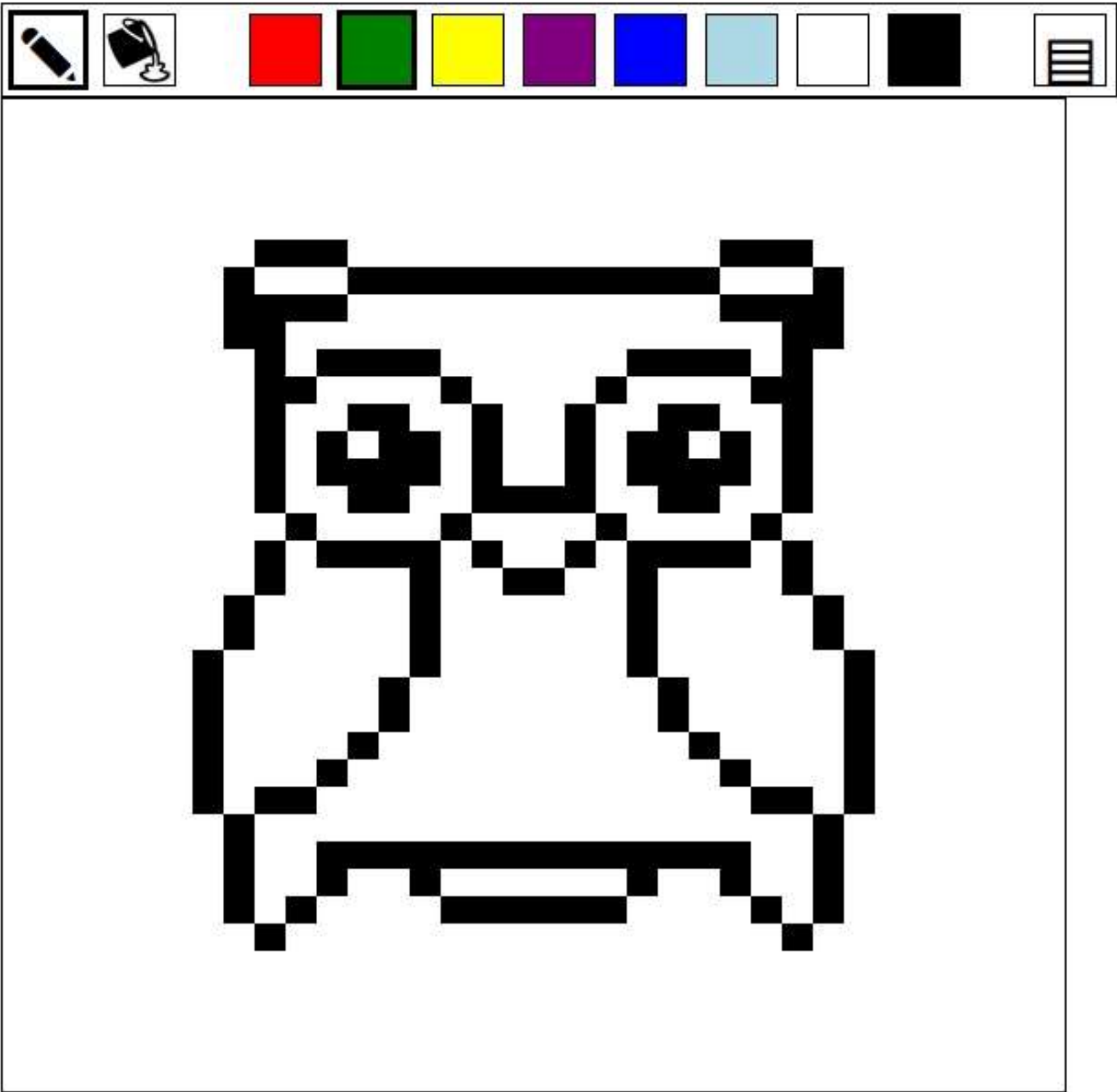
Chapter 1: Описание проекта

- «Графический редактор» — компьютерная программа для рисования на холсте при помощи кисти разными цветами.
- Можно выбирать цвет кисти из блока слева.
- Можно Выбрать стирающую кисть.
- Можно очистить холст.
- Можно рисовать прямоугольники
- Можно рисовать пикселями.
- Имеется возможность выбора толщины кисти и стиля кисти, который будет определять каким образом будет соединяться кисть между собой (с закруглением, квадратно или криво).

Created by Andrew Ohurtsov |



Pixels Paint by Andrew Ohurtsov



Chapter 1: Детали реализации

Особенности реализации отдельного функционала представлены ниже:

Section 0.1: Функция "Кисть"

Зададим функцию для рисования линии

```
function lines() {
  //painting = false;
  //Remove event listeners so line won't draw rectangle
  if (removeRectangleInLine == 1) {
    canvas.removeEventListener('mousedown', rectMouseDown);
    canvas.removeEventListener('mouseup', rectMouseUp);
    canvas.removeEventListener('mousemove', rectMouseMove);
    canvas.removeEventListener('mouseout', rectMouseout);
  };

  //Initialize mouse coordinates to 0,0
  var mouse = { x: 0, y: 0};

  //Paint includes line width, line cap, and color
  paint = function() {
    ctx.lineTo(mouse.x, mouse.y);
    ctx.lineWidth = lineWidthRange();
    ctx.lineJoin = 'round';
    ctx.lineCap = brushstyle;
    ctx.strokeStyle = colors;
    ctx.stroke();
  };
}
```

```

    }

    //Find mouse coordinates relative to canvas
    linesMousemove = function(e){
        mouse.x = e.pageX - this.offsetLeft;
        mouse.y = e.pageY - this.offsetTop;
    };

    //User clicks down on canvas to trigger paint
    linesMousedown = function(){
        ctx.beginPath();
        ctx.moveTo(mouse.x, mouse.y);
        canvas.addEventListener('mousemove', paint, false);
    };

    //When mouse lifts up, line stops painting
    linesMouseup = function(){
        canvas.removeEventListener('mousemove', paint, false);
    };

    //When mouse leaves canvas, line stops painting
    linesMouseout = function() {
        canvas.removeEventListener('mousemove', paint, false);
    };

    //Event listeners that will trigger the paint functions when
    //mousedown, mousemove, mouseup, mouseout
    canvas.addEventListener('mousedown', linesMousedown, false);
    canvas.addEventListener('mousemove', linesMousemove, false);
    canvas.addEventListener('mouseup', linesMouseup, false);
    canvas.addEventListener('mouseout', linesMouseout, false);

};
```

Section 0.2: Функция "Выбор цвета"

В зависимости от нажатой кнопки на цветовой палитре, выбираем нужный цвет

```

var colors;

/**
 * Выбрать цвет из палитры цветов
 * @param palette Палитра цветов
 * @return цвет
 */
function changeColors(palette) {
    switch(palette.id) {
        case "red":
            colors = "red";
            break;
        case "red1":
            colors = "#F16161";
            break;
        case "red2":
            colors = "#F69FA0";
            break;
        case "orange":
            colors = "orange";
            break;
        case "orange1":
            colors = "#F99F62";
            break;
        case "orange2":
            colors = "#FBB57B";
            break;
        case "blue":
            colors = "#09C2DB";
            break;
        case "blue1":
            colors = "#8BD3DC";
            break;
        case "blue2":
            colors = "#B9E3E8";
            break;
        case "indigo":
            colors = "#0E38AD";
            break;
        case "indigo1":
            colors = "#546AB2";
            break;
        case "indigo2":
            colors = "#9C96C9";
            break;
        case "green":
            colors = "green";
            break;
        case "green1":
            colors = "#97CD7E";
            break;
        case "green2":
            colors = "#C6E2BB";
            break;
        case "black":
            colors = "black";
            break;
        case "black1":
            colors = "#545454";
    }
}
```

```

        break;
        case "black2":
            colors = "#B2B2B2";
            break;
        case "yellow":
            colors = "yellow";
            break;
        case "yellow1":
            colors = "#F7F754";
            break;
        case "yellow2":
            colors = "#F7F4B1";
            break;
        case "purple":
            colors = "#B9509E";
            break;
        case "purple1":
            colors = "#D178B1";
            break;
        case "purple2":
            colors = "#E3ABCE";
            break;
        case "erase":
            colors = "white";
            break;
    }
};

```

Section 0.3: Функция "Нарисовать прямоугольник"

Позволяет пользователю нарисовать прямоугольники

```

function rectangle() {
    removeRectangleInLine = 1;

    canvas.removeEventListener('mousedown', linesMouseDown, false);
    canvas.removeEventListener('mousemove', linesMouseMove, false);
    canvas.removeEventListener('mouseup', linesMouseUp, false);
    canvas.removeEventListener('mouseout', linesMouseOut, false);

    //Initialize mouse coordinates to 0,0
    var mouse = {x: 0, y: 0};

    //Draw rectangle
    draw = function() {
        ctx.fillStyle = "black";
        ctx.fillStyle = colors;
        ctx.fillRect(mouse.x, mouse.y, mouse.w, mouse.h);
    };

    //Find mouse coordinates relative to canvas
    rectMouseMove = function(e) {
        mouse.w = (e.pageX - this.offsetLeft) - mouse.x;
        mouse.h = (e.pageY - this.offsetTop) - mouse.y ;
    };

    //User clicks down on canvas to trigger draw
    rectMouseDown = function(e) {
        ctx.beginPath();
        mouse.x = e.pageX - this.offsetLeft;
        mouse.y = e.pageY - this.offsetTop;
        canvas.addEventListener('mousemove', draw, false);
    };

    //When mouse lifts up, line stops drawing
    rectMouseUp = function() {
        canvas.removeEventListener('mousemove', draw, false);
    };

    //When mouse leaves canvas, line stops drawing
    rectMouseout = function() {
        canvas.removeEventListener('mousemove', draw, false);
    };

    //Event listeners that will trigger the draw functions when
    //mousedown, mousemove, mouseup, mouseout
    canvas.addEventListener('mousedown', rectMouseDown, false);
    canvas.addEventListener('mouseup', rectMouseUp, false);
    canvas.addEventListener('mousemove', rectMouseMove, false);
    canvas.addEventListener('mouseout', rectMouseout, false);
};

```