

# Mobilny Paint



**Politechnika  
Śląska**

*Autorzy :*

Szymon Babula, Krystian Barczak

Aleksander Boronowski, Krzysztof Dragon

Wydział Matematyki Stosowanej

Kierunek Informatyka

V semestr - Grupa 2C

# Spis treści

1	Opis programu . . . . .	2
2	Instrukcja obsługi . . . . .	3
3	Specyfikacja techniczna . . . . .	6
4	Szczegóły techniczne . . . . .	7

# 1 Opis programu

Webowa, uproszczona wersja programu Paint. Wersja ta posiada takie funkcje jak:

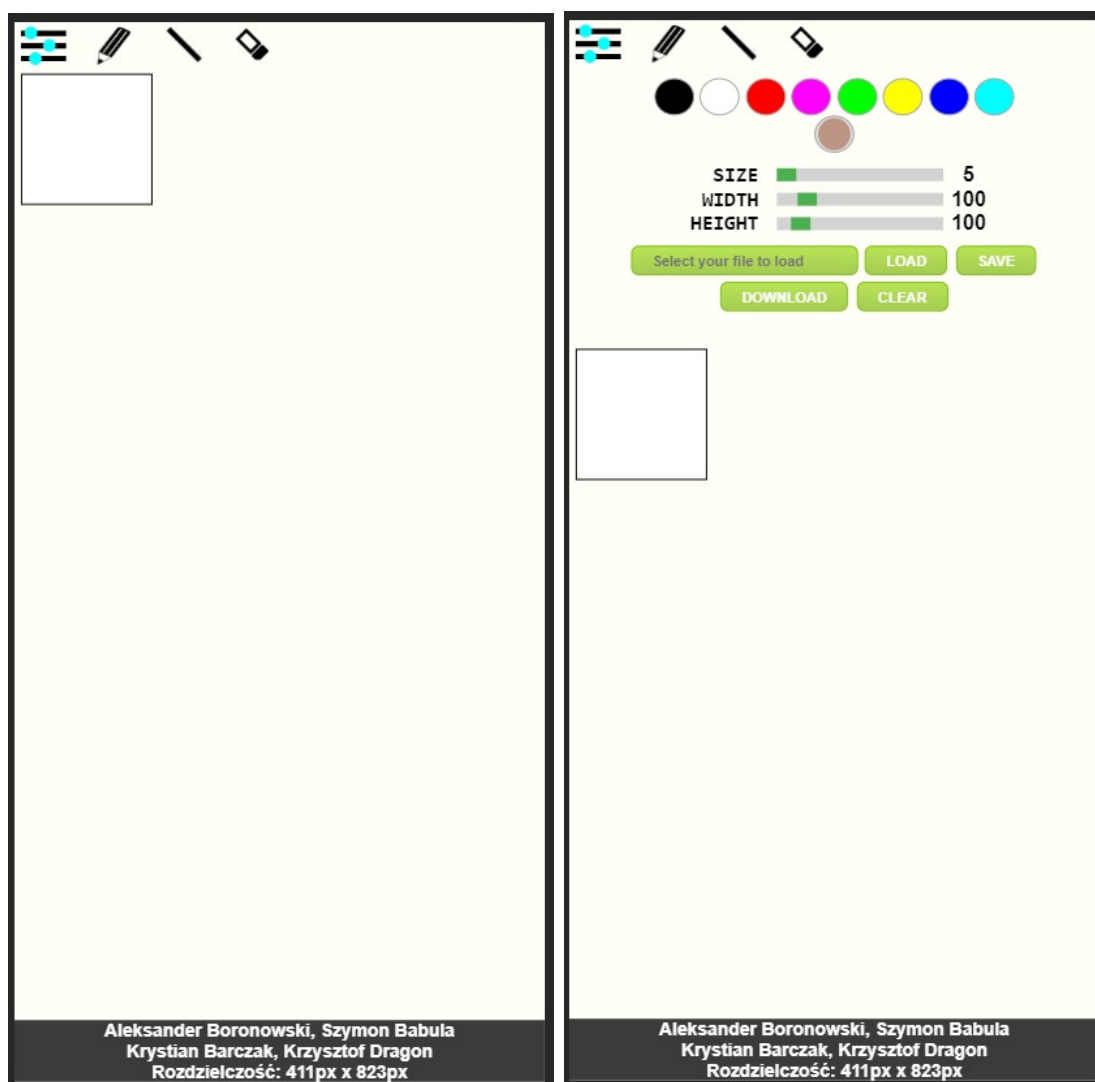
- Rysowanie za pomocą ołówka
- Rysowanie prostych linii
- Gumka
- Zmiana używanego koloru
- Dostosowanie planszy do różnych rozmiarów ekranów w urządzeniach mobilnych
- Możliwość wyboru rozmiaru planszy
- Zapis oraz odczyt obrazu z rozszerzeniem .png z serwera
- Możliwość zapisu stanu planszy jako obrazu png na urządzenie

Program został wykonany w celu projektu zaliczeniowego z przedmiotu Mobilne aplikacje webowe.

## 2 Instrukcja obsługi

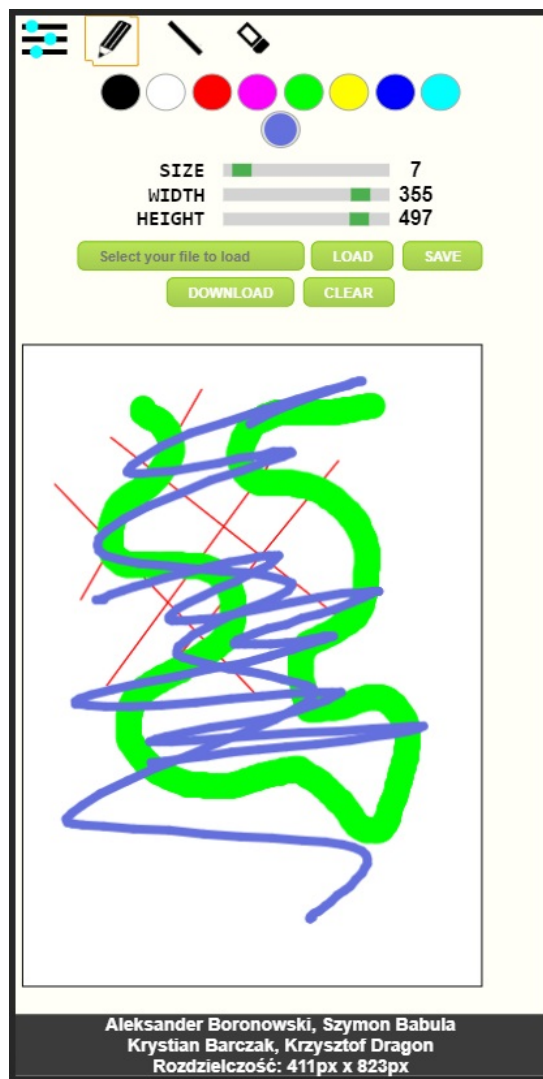
### 1. Obsługa ustawień i planszy

Po otwarciu Paint'a ukazuje się przycisk ustawień, przyciski funkcyjne, a także mała plansza do rysowania. Aby wyświetlić więcej ustawień należy kliknąć przycisk ustawień. Wysunie się wtedy menu ustawień. Umożliwia ono wybór używanego koloru (spośród wcześniej zdefiniowanych jak również możliwość wyboru własnego), wybór rozmiaru narzędzia, wybór rozmiaru planszy, a także zapisanie i wczytanie obrazu zapisanego na serwerze, pobranie obrazu i wyczyszczenie planszy.



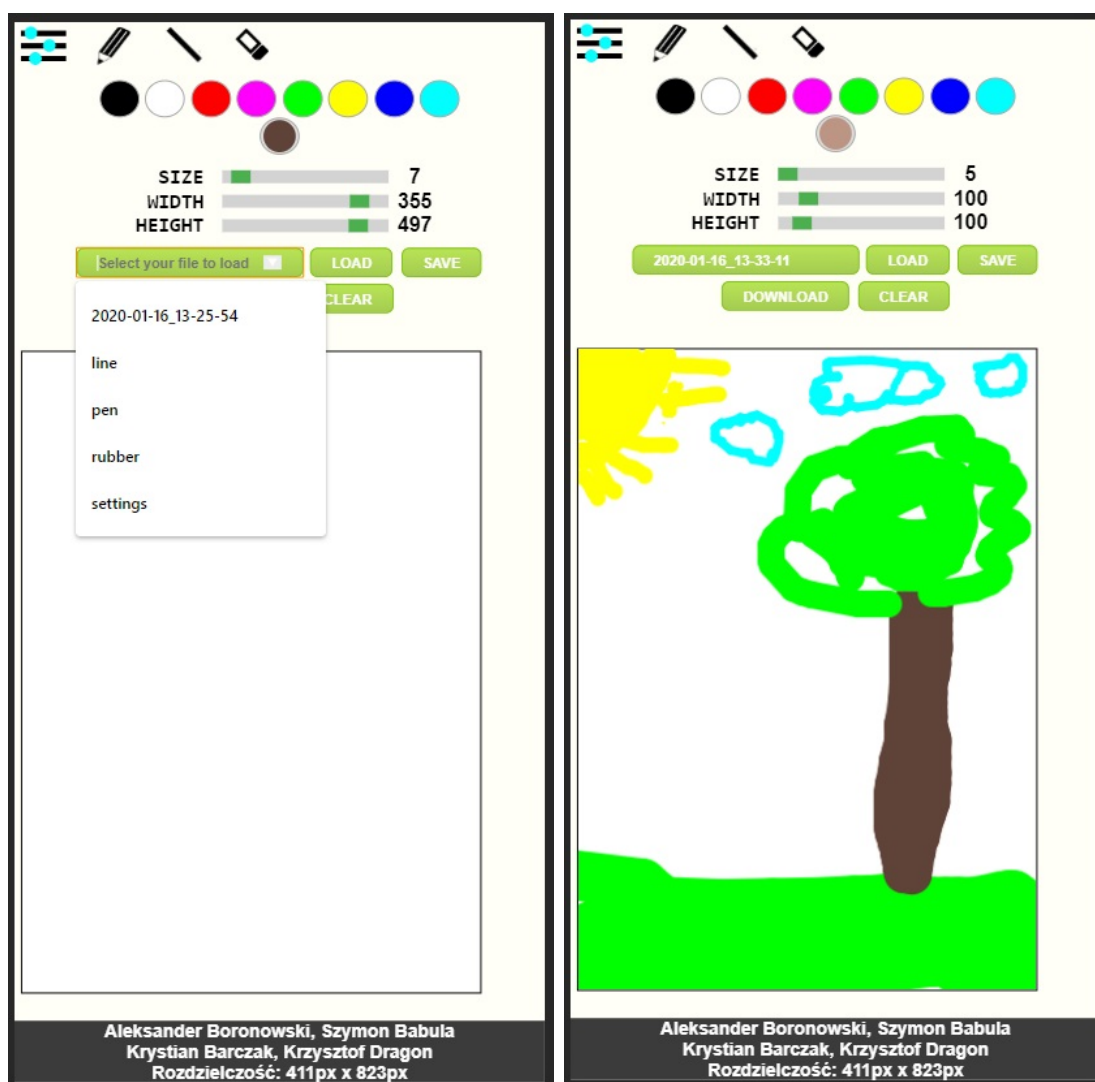
## 2. Instrukcja rysowania

Aby rysować na planszy wystarczy wybrać narzędzie (domyślnie ołówek), kolor (domyślnie czarny), a następnie rysować po planszy wyznaczonej czarną ramką. Chcąc wyczyścić od razu całą planszę wystarczy kliknąć przycisk **Clear**.



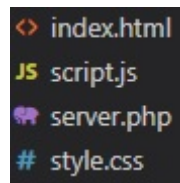
### 3. Zapis, odczyt, pobieranie

Ponadto znajdziemy takie opcje jak **Load** - wczytanie obrazu z serwera, **Save** - zapisanie aktualnego stanu planszy jako obrazu, **Download** - pobranie aktualnego stanu planszy jako pliku z rozszerzeniem .png. Kliknięcie pola **Select your file to load** lub pola z nazwą pliku, wyświetli listę dostępnych plików. Po wybraniu wystarczy kliknąć **Load**, a wcześniej zapisany obraz zostanie wczytany. Zapisane obrazy znajdują się na serwerze.



### 3 Specyfikacja techniczna

**Podział na pliki:**



```
<> index.html  
JS script.js  
🐘 server.php  
# style.css
```

**Kompilacja projektu:**

Do stworzenia projektu wykorzystany został program Microsoft Visual Studio Code oraz przeglądarka Google Chrome. Aplikacja napisana została w języku HTML oraz CSS w pełnej responsywności dla urządzeń mobilnych. Funkcjonalność aplikacji natomiast napisana została w języku JavaScript, jQuery i PHP.

## 4 Szczegóły techniczne

### 1. Funkcja odpowiedzialna za rysowanie

Funkcja odpowiedzialna za rysowanie planszy napisana jest w JavaScript. Funkcja oparta jest na Canvas.

```
function startDotyku(e)
{
    tablicaDotyk.length=0;
    for (var i=0; i<e.changedTouches.length; i++)
    {
        var x=e.changedTouches[i].clientX;
        var y=e.changedTouches[i].clientY;
        tablicaDotyk.push({x:x,y:y});
    }
    CurrentCanvasState(can, false);
    e.preventDefault();
    e.stopPropagation();
}
function stopDotyku()
{
    tablicaDotyk.length=0;
}
function ruchDotyku(e)
{
    var x1=can.getBoundingClientRect().left;
    var y1=can.getBoundingClientRect().top;

    for (var i=0; i<e.changedTouches.length; i++)
    {
        var x=e.changedTouches[i].pageX;
        var y=e.changedTouches[i].pageY;

        if (mode === 'pen') {
            q.beginPath(); //rozpociecie sciezki malowania
            q.moveTo(tablicaDotyk[i].x-x1, tablicaDotyk[i].y-y1);
            q.lineWidth=document.getElementById("sizePen").value;
            q.lineTo(x-x1, y-y1); //deklaracja rysowania
            q.strokeStyle = color;
            q.lineJoin = "round";
            q.lineCap = "round";
            q.stroke();
            tablicaDotyk[i].x=x;
            tablicaDotyk[i].y=y;
        }

        if (mode === 'line')
        {
            CurrentCanvasState(can, true)
            q.beginPath();
            q.moveTo(tablicaDotyk[i].x-x1, tablicaDotyk[i].y-y1);
            q.lineTo(x-x1, y-y1);
            q.strokeStyle = color;
            q.closePath();
            q.stroke();
        }
        if (mode === 'rubber')
        {
            document.getElementById("sizePen").value;
            q.clearRect(e.changedTouches[i].pageX-x1, e.changedTouches[i].pageY-y1, document.getElementById("sizePen").value, document.getElementById("sizePen").value);
        }
    }
    //brak odswiezania
    e.preventDefault();
    e.stopPropagation(); }
```



## 2. Funkcja odpowiedzialna za ustawienie rozmiaru planszy

Funkcja odpowiedzialna za ustawienie rozmiaru planszy napisana jest w JavaScript. Funkcja oparta jest na Canvas.

```
function sizeOfCanvas(){
    document.getElementById("heightY").max = Math.floor(window.innerHeight)-253;
    document.getElementById("heightX").max = Math.floor(window.innerWidth)-10;

    var sliderY = document.getElementById("heightY");
    var outputY = document.getElementById("valuesY");

    outputY.innerHTML = sliderY.value;

    sliderY.oninput = function() {
        sliderY.innerHTML= this.value;

        valuesY.innerHTML = this.value;
    }

    var sliderX = document.getElementById("heightX");
    var outputX = document.getElementById("valuesX");
    outputX.innerHTML = sliderX.value;

    sliderX.oninput = function() {
        sliderX.innerHTML = this.value;
        valuesX.innerHTML = this.value;
    }
    ResizeCanvas(sliderX.value,sliderY.value);
}

function ResizeCanvas(width,height)
{
    localStorage.setItem(can,can.toDataURL());

    can.width = width;
    can.height = height;
    can2.width = width;
    can2.height = height;

    var img = new Image;
    img.src = localStorage.getItem(can);

    img.onload = function () {
        q.drawImage(img, 0, 0);
    };
}

function ResizeCanvas2(width,height)
{
    can.width = width;
    can.height = height;
    can2.width = width;
    can2.height = height;
}
```

### 3. Funkcje odpowiedzialne za zapis oraz odczyt

Funkcje odpowiedzialne za zapis oraz odczyt napisane są w języku PHP oraz formatu JSON. Zapis polega na sczytaniu oraz zapisaniu stanu planszy oraz wysłaniu go na serwer w JSON. Następnie w PHP dane zapisane są w pliku, której nazwa jest datą i godziną zapisu. Odczyt polega na wysłaniu na serwer próśby w formacie JSON do odczytu zawartości wybranego pliku. Serwer wysyła dane z pliku do JavaScript a ten wysyła je do funkcji, która wypełnia planszę otrzymanymi danymi.

#### Zapis pliku w JSON:

```
function save(){
    var request = new XMLHttpRequest();
    request.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            console.log(this.response);
        }
    }
    request.open("POST", "server.php", true);
    request.send(JSON.stringify({
        polecenie: 1,
        dane: can.toDataURL()
    }));
}
```

#### Odczyt pliku JSON:

```
function load(){
    console.log("load");
    clearCanvas();
    var plik = document.getElementById('plik').value;
    var request = new XMLHttpRequest();
    request.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            var img = new Image;
            img.src = this.response;
            img.onload = function ()
            {
                ResizeCanvas2(img.width, img.height);
                q.drawImage(img, 0, 0);
            }
        }
    }
    request.open("POST", "server.php", true);
    request.send(JSON.stringify({
        polecenie: 4,
        plik: plik+".png"
    }));
}
```

## Działanie serwera w PHP:

```

if(isset($daneJSON['polecenie']))
{
    $polecenie = intval($daneJSON['polecenie']);
    switch($polecenie)
    {
        case 1:
            $data = date("Y-m-d_H-i-s");
            $img = $daneJSON['dane'];
            $img = str_replace('data:image/png;base64,', '', $img);
            $img = str_replace(' ', '+', $img);
            $dane = base64_decode($img);
            $file = $data.'.png';
            $success = file_put_contents($file, $dane);
            print $success ? $file : 'Unable to save the file.';
            break;
        case 2:
            $data = date("Y-m-d_H-i-s");
            $file = $data.'.png';
            echo $file;
            break;
        case 3:
            $wynik = '';
            $files = glob('{*.png}', GLOB_BRACE);
            foreach($files as $file){
                $file = substr($file, 0, -4);
                if ($wynik == '') {
                    $wynik = '<option value="'. $file. '">';
                } else {
                    $wynik .= '<option value="'. $file. '">';
                }
            }
            print_r($wynik);
            break;
        case 4:
            $img_file = $daneJSON["plik"];
            $imgData = base64_encode(file_get_contents($img_file));
            $src = 'data: '.mime_content_type($img_file).'';base64,'.$imgData;
            echo $src;
            break;
        default:
            $wynik = array('status' => false, 'kod' => 3, 'wartosc' => 'Podane zostalo
            zle polecenie');
    }
}

```

# Bibliografia

- [1] *[https://www.w3schools.com/html/html5\\_canvas.asp](https://www.w3schools.com/html/html5_canvas.asp)*
- [2] *[https://en.wikipedia.org/wiki/Canvas\\_element](https://en.wikipedia.org/wiki/Canvas_element)*
- [3] *[https://developer.mozilla.org/pl/docs/Web/API/Canvas\\_API/Tutorial](https://developer.mozilla.org/pl/docs/Web/API/Canvas_API/Tutorial)*