

Paint.Web

- Marius și Mihai Șucan
- Universitatea "Aurel Vlaicu", Arad, România



Noiembrie 2008

Paint.Web

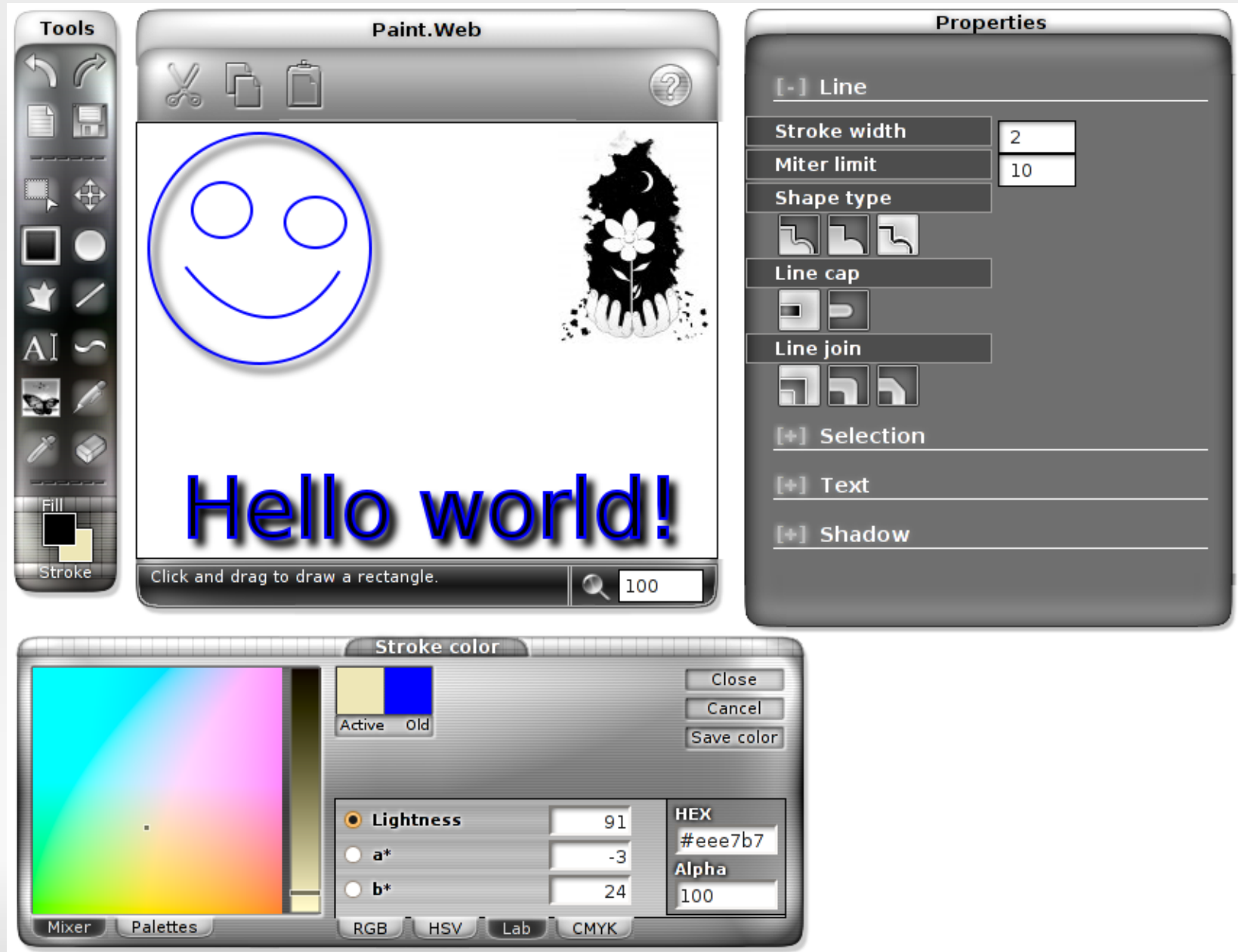
Cuprinsul prezentării

- Introducere
- Interfața
- Funcționalități
- Utilizare
- Structură
- HTML 5
- JavaScript
- Editorul de culori
- Compatibilitate
- Ce poate fi mai bine
- Planuri
- Canvasul în viitor

Paint.Web – Introducere

- Aplicație Web de desenat. Proiect open-source, GPL v3, găzduit pe code.google.com/p/paintweb.
- Demonstrație a tehnologiei **<canvas>** din HTML5, împreună cu Web Forms 2.
- Aplicația folosește contextul 2D al canvasului. Acest API permite desenarea de tip bitmap cu ajutorul unor funcții de bază.
- Tehnologia a fost prima dată implementată în Safari de Apple.

Paint.Web – Interfața



Paint.Web – Funcționalități

- Desenare:
 - selecție, dreptunghi, elipsă, linie, curbă Bézier, poligon, creion, text, imagini și gumă de șters.
- Opțiuni:
 - salvare imagine în format PNG, istoric (undo/redo), redimensionare canvas, pipetă (color picker), zoom.
- Proprietăți:
 - culoare umplere-formă, culoare linie, grosime linie, linii cu capăt rotund/pătrat/bevel, forme cu borduri sau fără, umbre (poziție și culoare), dimensiune text, alegere font, aliniere și stilul text (aldin, italic).

Paint.Web – Utilizare

- Interfață intuitivă ce afișează informații contextual.
- Comenzi din tastatură la toate uneltele și butoanele din aplicație.
- În timpul desenării multe unelte permit utilizarea tastei Shift pentru modificarea funcționalității.
 - De exemplu, dreptunghiul devine pătrat, sau elipsa devine cerc.

Paint.Web – Structură

- HTML pentru interfață.
- CSS pentru layout/design + imagini PNG.
- JavaScript pentru interactivitate.
- Delimitarea este strictă: în cod JS nu veți găsi cod CSS, nici în codul HTML nu veți găsi CSS/JS.
- Aplicația nu folosește absolut deloc cod server-side (PHP, Perl, Python, etc). Se poate rula fără server Web.

Paint.Web – HTML 5

index.html

- `<!DOCTYPE html>`
 - Relevanța DOCTYPE-ului este activarea renderului strict pentru CSS.
- Atribute noi:
 - `<meta charset="utf-8">`
 - Web Forms 2:
 - `Input type=number/range` cu `min=0 max=100` și `step=0.1`
 - ... și altele
- Nu se validează cu `validator.w3.org`.
 - Încercați www.validator.nu

Paint.Web – JavaScript

index.js

- Totul este într-un singur obiect ce se execută la încărcare.
 - Fără poluare DOM cu funcții și proprietăți globale.
 - Ușor de integrat în alte proiecte, fără conflicte de nume de funcții sau variabile.
- 6300 de linii de cod, cu comentarii la fiecare funcție, explicând alegerile făcute și cum funcționează.
- Ușor de tradus: toate mesajele din aplicație sunt grupate în două obiecte.

Paint.Web – JavaScript: Structură

Privire în ansamblu

- Vom numi "app" obiectul global al aplicației.
- ```
app = function () {
 ... elems, inputs,
 img, img_temp,
 messages, status_texts,
 init, init_tools,
 init_keys, tool_activate,
 ev_canvas, ev_keypress,
 tools, kshortcuts, ... };
```
- ```
window.onload = app;
```

Paint.Web – JavaScript: Rulare

- În **app.elems** și **.inputs** sunt memorate referințe directe la nodurile DOM importante.
- **app.img** și **.img_temp** sunt contextele 2D de desenare.
 - În timpul utilizării unei unelte toată desenarea se face pe bufferul/contextul temporar (**img_temp**).
 - La final, codul uneltei de desenare cheamă **app.img_update()** care copiază imaginea din **.img_temp** în **.img**, și face un pas în istoric (pentru undo/redo) cu **app.history_add()**.

Paint.Web – JavaScript: Rulare

- În **app.messages** și **.status_texts** se găsesc toate mesajele folosite de aplicație.
 - În **.status_texts** se pot asocia mesaje pentru nodurile din DOM, folosind ID-urile lor.
- **app.kshortcuts** menține lista de comenzi din tastatură.
 - Fiecare comandă poate activa o unealtă și/sau o funcție.
- **app.tools** menține codul fiecărei unelte.
 - Uneltele pot avea cod pentru activare și pentru fiecare eveniment al mouse-ului sau al tastaturii.

Paint.Web – JavaScript: Rulare

- **app.init()** pregătește contextele canvas și adaugă majoritatea evenimentelor din aplicație.
 - Se mai execută inițializarea uneltelor, a comenzilor din tastatură, a proprietăților, și a editorului de culori (**init_tools()**, **init_properties()**, **coloreditor.init()** și **init_keys()**).
- **app.init_tools()** se ocupă cu:
 - Adăugarea de event listeners la butoanele din bara de unelte
 - Activarea uneltei implicite (**app.tool_default**).

Paint.Web – JavaScript: Rulare

- **app.init_keys()** adaugă în titlul și statusul fiecărui buton comanda din tastatură.
 - Acest lucru se face cu scopul ca utilizatorul să fie informat automat despre comenzile disponibile.
 - Exemplu: **Undo [ctrl-z]**
- **app.ev_keypress()** este funcția care se execută când se tastează orice.
 - În **app.kshortcuts** se caută combinația de taste. Fiecare combinație poate avea o funcție/unealtă asociată, care se activează automat.

Paint.Web – JavaScript: Rulare

- **app.ev_canvas()** este funcția care se ocupă de toate evenimentele ce țin de canvas.
 - Pentru fiecare eveniment capturat se verifică dacă unealta activă în acel moment oferă un "event handler" - o funcție care să se execute (**app.tool.event**). Dacă da, atunci funcția este executată.
 - Funcția determină și poziția exactă a mouseului pe canvas, indiferent de zoom/scroll, pentru transmiterea coordonatelor către funcția chemată (**ev._x** și **ev._y**).

Paint.Web – JavaScript: Rulare

- **app.tool_activate(id, ev)** este funcția ce permite activarea uneltelor.
 - Se preia ID-ul uneltei și, opțional, un eveniment DOM.
 - Codul de construcție a obiectului unei unelte poate anula activarea.
 - Un asemenea exemplu este unealta de adăugare imagine: în cazul anulării nu se mai face activare.
 - Obiectul uneltei active este setat să fie **app.tool**.

Paint.Web – JavaScript

Exemplu minimal de cod din unealta "dreptunghi"

- În `app.tools` se găsesc implementările uneltelor.

```
▪ app.tools = {  
    'rect' : function () {  
        var _tool = this;  
        _tool.mousedown = function (ev) {  
            _tool.x0 = ev._x  
            _tool.y0 = ev._y;  
            _tool.start = true;  
        };  
        _tool.mouseup = function () {  
            app.img_update();  
            _tool.start = false;  
        };  
    };
```

.....

Paint.Web – JavaScript

Exemplu minimal de cod din unealta "dreptunghi"

```
■   _tool.mousemove = function (ev) {  
    if (!_tool.start) return;  
  
    var x = Math.min(ev._x, _tool.x),  
        y = Math.min(ev._y, _tool.y),  
        w = Math.abs(ev._x - _tool.x),  
        h = Math.abs(ev._y - _tool.y);  
  
    app.img_temp.clearRect(0, 0,  
                           app.imgW, app.imgH);  
    app.img_temp.fillRect(x, y, w, h);  
};  
  } // app.tools.rect  
}; // app.tools
```

Paint.Web – JavaScript

- În realizarea fiecărei unelte s-a încercat ca orice acțiune să aibă vizualizare în timp real.
 - De exemplu, curbele Bézier sunt vizibile încă de la primele puncte. În alte programe de același gen, rezultatul apare numai după alegerea celor 4 puncte.
- Selecția folosește un al treilea element canvas pentru stocarea pixelilor selectați.
 - Spre deosebire de alte aplicații de același gen, Paint.Web permite manipularea selecției individual, și a pixelilor selectați (mutare, redimensionare).

Paint.Web – Editorul de culori

- În **app.coloreditor** se găsesc toate funcțiile ce țin de editorul de culori:
 - Conversie de culori între diferite spații de culori și reprezentări: RGB, HSV, CIE Lab și CMYK.
 - **draw_chart()** și **draw_slider()** se ocupă cu vizualizarea spațiilor de culori. Desenarea pixelilor se face într-un element **<canvas>**.
 - Palete de culori predefinite, importate din Photoshop.

Paint.Web – Compatibilitate

Opera 9.5 (Kestrel)

- Din cauza unei erori, cursorul nu se actualizează în timp util când se folosește selecția.
- Pe Mac și pe Windows, metoda **drawImage()** întunecă transparențele.
- Umbrele și adăugarea de text nu sunt implementate.

Paint.Web – Compatibilitate

Opera 9.2 (Merlin)

- Are aceleași probleme ca Opera 9.5, plus altele.
- Nu implementează **get/putImageData**: Undo/Redo și vizualizarea spațiului de culori CIE Lab nu funcționează.
- Nu implementează **globalCompositeOperation 'lighter'**. Vizualizarea spațiului de culori RGB nu funcționează.
- Nu implementează unele selectoare CSS 3.

Paint.Web – Compatibilitate

Safari 3+

- Safari 3: similar cu Opera 9.2, nu are suport pentru **get/putImageData**, umbre și nici pentru text.
- Ultimele versiuni de Webkit (SVN trunk builds) implementează text și **get/putImageData**.
- WebKit este cel mai rapid renderer.

Paint.Web – Compatibilitate

Firefox

- Versiunea 2:
 - Înceată la render, și extrem de înceată cu **get/putImageData**.
 - Unealta "Gumă de șters" (Eraser) nu funcționează.
 - Nu are suport pentru text, nici pentru umbre.
- Versiunea 3 este mult mai rapidă, similar cu Opera 9.5.
 - Nu are suport pentru text, nici pentru umbre.
- Versiunea 3.1 beta implementează text și umbre.
 - Paint.Web funcționează complet.

Paint.Web – Compatibilitate

Alte navigatoare Web

- Konqueror 4+
 - O dată cu lansarea noii versiuni cei de la KDE au pregătit o surpriză: propria implementare **<canvas>**.
 - Are suport pentru tot API-ul, în afară de adăugarea de text.
 - Momentan întreaga aplicație este puțin instabilă, dar problemele se rezolvă rapid.
- Microsoft Internet Explorer
 - Nu are suport pentru canvas. Există eforturi din partea mai multor grupuri de a implementa **<canvas>** și pe IE.

Paint.Web – Ce poate fi mai bine

- Sistemul de stocare a istoricului (Undo/Redo) salvează numai imaginile la fiecare pas.
 - Un sistem hibrid pentru istoric ar fi mai eficient.
- Unealta pentru selecție trebuie să fie regândită la o "scară mai largă".
 - Scopul fiind selecția și de alte forme.
- Optimizări de performanță.
- Interfață mai dinamică: side panels și floating panels.

Paint.Web – Planuri

... sau niște "idei"

- Funcționalități mai importante:
 - Filtre, layers, degradeuri, patterns, "smart objects", plugins, și chiar SVG.
- Interfață animată: CSS Animations sau SVG.
- Integrare în aplicații Web (mult) mai mari.
- Mai mult HTML5:
 - Offline Web Application
 - Client-side storage și database storage
 - Drag and drop
 - Server-sent events

Paint.Web – Canvasul în viitor

- Canvas-ul poate avea mai multe contexte.
- Pe lângă contextul 2D, principalele browsere sunt în curs de implementare a contextului 3D.
- Accelerare hardware cu OpenGL/DirectX la ambele contexte.
- Pentru contextul 2D se vor mai implementa metode noi.
 - În cursul realizării aplicației Paint.Web s-a implementat API-ul pentru render text și umbre.

Paint.Web – Merci

- Vă mulțumim pentru timpul acordat.
- Pentru a testa aplicația intrați pe:
 - www.robodesign.ro/paint.web
 - <http://code.google.com/p/paintweb>

