Interactive Canvas

A Demo of the Exciting Features of HTML5 Canvas and WebSocket

GUO Yuxiang
WU Sisi
ZHANG Yaofeng
ZHANG Yusi
ZHAO Guanlun

Outline

Motivation

You may find it difficult with only e-mail to do the followings

Motivation

You may find it difficult with only e-mail to do the followings

- **Discussing**
  Sharing opinions on the architecture of a piece of software

Motivation

You may find it difficult with only e-mail to do the followings

- **Discussing**
  Sharing opinions on the architecture of a piece of software
- **Designing**
  Working together on the appearance of a website

Motivation

You may find it difficult with only e-mail to do the followings

- **Discussing**
  Sharing opinions on the architecture of a piece of software
- **Designing**
  Working together on the appearance of a website
- **Explaining**
  Teaching your friend a math problem

Motivation

Now our Interactive Canvas can help you out

Motivation

Now our Interactive Canvas can help you out

- **Clear**
  Drawing, a more straightforward way than using text

Motivation

Now our Interactive Canvas can help you out

- **Clear**
  Drawing, a more straightforward way than using text
- **Fast**
  A faster way to express your idea

Motivation

Now our Interactive Canvas can help you out

- **Clear**
  Drawing, a more straightforward way than using text
- **Fast**
  A faster way to express your idea
- **User-Friendly**
  User interface is simple but elegant and convenient

Design

A CGI program based on socket and HTML5.

- **Server Side**
  Mojolicious, the Perl web framework
- **Client Side**
  HTML5 canvas and jQuery
- **Communication**
  JSON (JavaScript Object Notation)
- **User Interface**
  HTML with CSS and jQuery UI

Design - Server Side

About 500 lines of Perl code.

- **Receiving Messages**
  Receive the messages sent by the clients
- **Parsing Messages**
  Perform different tasks according to the contents of messages
- **Database Manipulation**
  A database to store the line segments and chatting messages
- **Sending Messages Back**
  Send messages to the clients

Design - Client Side

About 1,200 lines of Javascript, with the help of jQuery library

- **Initializing Connections**
  Establish connections with the server
- **Detecting Event**
  Detect and respond to mouse events
- **Sending Messages**
  Send the messages to the server
- **Receiving Data**
  Get data from the server and perform correspondingly
- **Displaying Data**
  Draw on the canvas and display chatting messages

Design - Communication

Making use of JSON for data communication

- **Stringifying (Encode)**
  Store the data in an object into a string
- **Sending through WebSocket**
  Use the WebSocket to send between the server and clients
- **Parsing String**
  Parse the strings to get the data objects

Design - User Interface

More than 400 lines of HTML, CSS and more Javascript to control UI

- **Simple and Elegant**
  GoogleDocs style appearance
- **jQuery UI**
  Making use of the jQuery UI library

Highlights

Some interesting points that worth attention

- **Multiple Canvas**
  Applied for undo and redo
- **Upload and Download (To Be Implemented)**
  For better user experience

Highlights - Multiple Canvas

Undo and Redo, requires three types of canvas for different tasks

Highlights - Multiple Canvas

Undo and Redo, requires three types of canvas for different tasks

- **Base Canvas**
  For drawing the ”static” segments

Highlights - Multiple Canvas

Undo and Redo, requires three types of canvas for different tasks

- **Base Canvas**
  For drawing the ”static” segments
- **User’s Layers**
  For drawing the tentative segments

Highlights - Multiple Canvas

Undo and Redo, requires three types of canvas for different tasks

- **Base Canvas**
  For drawing the ”static” segments
- **User’s Layers**
  For drawing the tentative segments
- **Detector**
  Detecting the mouse events

Highlights - Multiple Canvas

Undo and Redo, requires three types of canvas for different tasks

- **Base Canvas**
  For drawing the ”static” segments
- **User’s Layers**
  For drawing the tentative segments
- **Detector**
  Detecting the mouse events

Arrangement of layers is changed frequently

Highlights - Multiple Canvas

Undo and Redo, requires three types of canvas for different tasks

- **Base Canvas**
  For drawing the "static" segments
- **User's Layers**
  For drawing the tentative segments
- **Detector**
  Detecting the mouse events

Arrangement of layers is changed frequently

- **User Logging In and Out**
  Inserting and deleting canvases

Highlights - Multiple Canvas

Undo and Redo, requires three types of canvas for different tasks

- **Base Canvas**
  For drawing the ”static” segments
- **User’s Layers**
  For drawing the tentative segments
- **Detector**
  Detecting the mouse events

Arrangement of layers is changed frequently

- **User Logging In and Out**
  Inserting and deleting canvases
- **Drawing**
  Rearranging the order of canvases

Highlights - Upload and Download

This can make the application more practical

- **Download**
  Make a copy of the canvas
- **Upload**
  Increase usability, not yet implemented

Further Development

What to add to the application

- **Get A Server** Buy some space for this application
- **Signing Up and Workspace** Workspace of yourself to save you own work
- **Sharing** Share the files with others