Design Document for BOARD18

BOARD18 is an open source web application which supports remote play of various 18xx style games.

This document was last updated on May 8, 2013 at 10:40 AM.

Table of Contents

1 Introduction

I will assume that anyone reading this document is familiar with "18xx style" games such as 1830 and 1870. The web has much information about these games. Here Google is your friend.

18xx games are often played via Email, usually using a spread sheet and the CyberBoard program. CyberBoard is a general purpose game board program that runs on each players computer. Linux users must run CyberBoard under WINE. CyberBoard generates and accepts move files that synchronize the various players copies of the game file.

I have some problems with CyberBoard and the style of remote play that it supports. To address these problems I have decided to develop an alternative to this current arrangement. BOARD18 is this alternative. It is intended to replace CyberBoard. It will run as a web service accessed via the web browser on each players computer.[1]

2 Environment

2.1 Languages

2.1.1 Client Side

Board 18 will be developed using HTML5, CSS 3 (to the extent currently supported) and the JavaScript standard referred to as ECMAScript 5. I will also be using the jQuery-1.8.x library.

The Firefox browser will be used in this project. Currently Firefox version 14.x is being used. I recommend that this browser [or a later version of it] be used to run BOARD18. Other modern browsers **may** work, but I can make no guarantee that they will. Note however that, as of the end of 2010, the current version of IE does not fully support HTML 5 Canvas. Thus I recommend avoiding it when using BOARD18.

2.1.2 Server Side

The server side of BOARD18 will use LAMP. LAMP stands for Linux, Apache, MySQL and PHP. PHP will be the programming language. And MySQL will be the data base language. The current development environment is based on PHP version 5.3 and MySQL version 5.5.

2.2 Database

There will be no client side data base. All permanent storage will be via the server side MySQL data base. PHP will be used for all MySQL access. PHP will use the procedural stile of the mysqli API to access the database.

2.3 Client/Server Communication

The client side of BOARD18 will use the XMLHttpRequest protocol to communicate with the server side. I will be using the get(), getJSON() and put() Ajax functions from the jQuery library for this.

2.4 User Interface

Board18 will accept the following user inputs: keyboard, mouse movement and mouse button clicks. Right clicks will activate a pop-up menu. Scroll wheel events will not be used by BOARD18.

---

1 Another open source program, called Rails, is sometimes used to play 18xx games by Email. Rails is a much more ambitious system that combines the graphics functions of CyberBoard with the bookkeeping functions of a spread sheet. It also includes rules enforcement. BOARD18 is **not** intended as a replacement for Rails.

3 Scope

BOARD18 is intended to be used with a spreadsheet to play an 18xx style game remotely. The Google+ spreadsheet is a good choice for this. As each player takes a turn, he will send an Email to the other players. The Email will contain a list of recent moves ending with his current move. It may also have an updated copy of the spread sheet being use to track the game finances [if you don't use Google+]. But it will no longer need to contain any other files.

The BOARD18 game session associated with the game will be used to display the game board and the stock market for the game. It will be updateable by any player who can log in to the game.

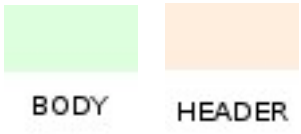
4 BOARD18 Esthetics

Like any other application, BOARD18 needs a distinctive look.

4.1 Logo and Favicon

The logo displayed to the right will be present in the upper left hand corner of every BOARD18 page. It will be part of the standard header for each such page. In addition, a 32 pixel square icon with a reduced version of the same graphic will be used as the favicon file for this application.

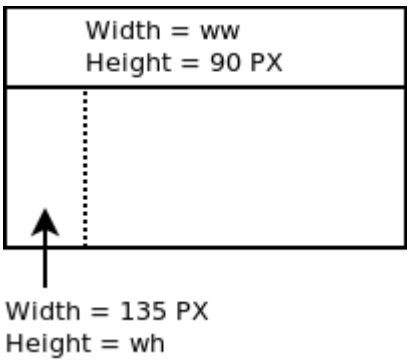
4.2 Color Standards



The background of the header and main areas of all pages will be the color shown on the far right [#FFEEDD]. The background of the left sidebar of all pages and any drop-down menus or forms will be the color shown on the near right [#DDFFDD].

Other areas may have modified background colors to enable them to stand out from their surroundings.

4.3 Page Layout



Each BOARD18 page will follow this overall layout. The page will be divided into 3 fixed divisions. The top of page division, the left of page division and the right of page division. These divisions, which are shown in the diagram to the right, are formatted via the style sheets described below.

4.4 "Top of Page" Division

This division forms the logo, heading, message and menu areas of the page. It consists of the top 90 pix of the page and it's width fills the width of the enclosing window.

4.5 "Left of Page" Division

This division forms the sidebar area of the page. It consists of the left 135 pix of the page and it's hight fills the hight of the enclosing window. The sidebar division scrolls within this division.

4.6 "Right of Page" Division

This division forms the main content area of the page. It consists of the remainder of the page and it's hight and width fills the remainder of the enclosing window. The content division scrolls within this division.

4.7 Style Sheets

BOARD18 will use cascading style sheets to control all formatting of its pages. There will be an application wide style sheet for all common formatting and another style sheet specific to each BOARD18 page. The application wide style sheet will be named board18com.css. The specific style sheets will have names based on the page with which they are associated [for example board18main.css].

4.7.1 The board18com.css style sheet

This style sheet lays out the overall structure of all BOARD18 pages. This consists of the top of page division, the left of page division and the right of page division. These divisions are described above. This style sheet also formats the header area of all BOARD18 pages. The top level block elements in this style sheet will all use fixed positioning.

4.7.2 All other BOARD18 style sheets

Every BOARD18 page will have a style sheet to format the body of the page. Some of the containing block elements in these style sheets may use absolute positioning.

5 Game Boxes

BOARD18 can be used to play any 18xx style game for which a game box exists. A game box consists of the graphics files and control tables that describe the game board, the stock market and the various playing pieces. A game box is built using a graphics editor [such as GIMP] and a text editor. The files created with these tools can then be uploaded to the BOARD18 server. This upload should eventually be accomplished via a built-in and automatic process.

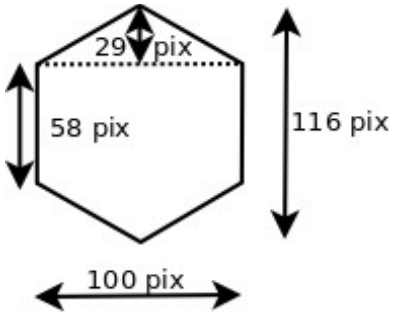
5.1 Graphics Files

5.1.1 Size Standards

In order to coordinate the size of the various graphics files, so that things fit together, BOARD18 will use the following sizing standards. Note that some of the Control Table settings can be used to resize the uploaded graphics to meet these standards.

5.1.1.1 Hex Tile Standard

All tiles should be sized as shown in the diagram to the right when displayed by BOARD18. If the Tile Sheet control values are correct, then this resizing should be done automatically.



5.1.1.2 Token Standard

All circular tokens should be sized to have a diameter of 30 pix. All Square tokens should be sized to have a width and height of 30 pix. Once again, resizing via the control values should be supported.

5.1.2 Game Board

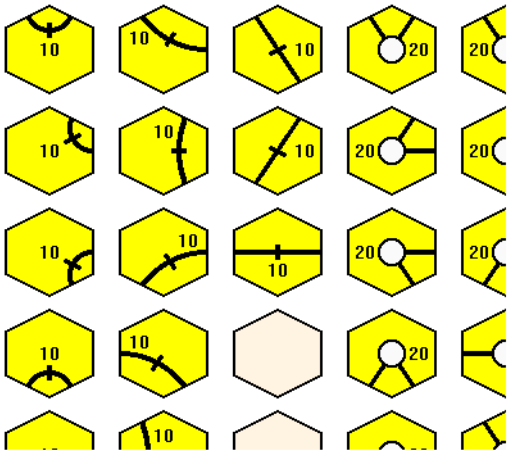
The game board graphic is the background of the BOARD18 board page. As with tiles and tokens, scaling can be controlled by the Game Board control values .

5.1.3 Stock Market

The stock market graphic is the background of the BOARD18 stock market page. Scaling can be controlled by the Stock Market control values.

5.1.4 Tiles

Hex tile graphic images are grouped onto "tile sheets" containing rows and columns of evenly spaced tiles. The illustration to the right shows a part of such a tile sheet. Each column is a different tile and each row in the column is a 60° rotation of the tile. There can be from one to six rows for each tile.



5.1.5 Tokens

Token graphic images are grouped onto "token sheets" containing rows and columns of evenly spaced tokens. The illustration to the right shows a part of such a token sheet. Each sheet will contain one or two columns of circular or square tokens. Each row in the column is a different token. If there are two columns then the image in the first shows the front side of the token and the second shows the back side.

5.2 Control Structures

In addition to the graphics files described in the above section, a BOARD18 game box also consists of a number of control structures. The actual game status also consists of control structures. These are formatted as two complex JSON data structures: one for the game box and one for the game session. Each of these structures is created and maintained as a single JSON file. And each is stored as a text field in the MYSQL database.

Appendix A contains a listing of the test game box file that is being used in the development of the BOARD18 application. [It is actually a game box for 1826.] The game box file is common to all game sessions and it is never modified by BOARD18.

Appendix B contains a listing of the initial game session file that is created by the the new game function of the BOARD18 application. This JSON structure will be constantly modified by BOARD18.

Do to the complexity of these structures, we will discuss them in detail in the following sub sections..

5.2.1 The JSON Box Data Structure

Th first part of the game box JSON file contains data structures that define the file itself, the game board and the stock market board. The **bname**, **version** and **author** keys are used to identify the JSON file itself. The **board** and **market** keys head complex structures that define the game map and the stock market. They each contain fields that define the location and size of the graphic and positioning of its grid.

5.2.2 The JSON Tray Data Structure

The data structure in the game box JSON file with the key of **tray** defines the tile and token sheets in the game box. It is an array of complex data structures each of which describes a tile or token sheet. There are actually three different types of sheets on a tray. The **type** key defines the type for each sheet in the array.

5.2.2.1 The Tile Sheet Type

The sheets with a type value of **tile** are tile sheets. The tName key defines the menu name for the sheet. The imgLoc key defines the location of the tile sheet image file. The next group of keys define the values that locate the actual tokens on the tile sheet image file.

Finally the **tile** key defines an array of values [one for each tile] that specify the the number of duplicates of that tile in the game box and the number of rotations that exist for each tile.

5.2.2.2 The Board Token Sheet Type

The sheets with a type value of **btok** are board token sheets. The first part of a board token sheet is the same as a tile sheet. But there is no **tile** key. Instead a token key defines an array of values [one for each token] that specify the the number of duplicates of that token in the game box and whether the token can be flipped. [i.e.: does it have a back side.]

5.2.2.3 The Market Token Sheet Type

The sheets with a type value of **mtok** are market token sheets. A market token sheet is the same as a board token sheet with one exception. The token key defines an array of values [one for each token] that specify only whether the token can be flipped.

5.2.3 The JSON Game Session Data Structure

Each game session has its own JSON data structure. The **gname**, and **boxID** keys are used to identify the JSON Game Session and specify the game box that the game session uses. These are followed by four array structures.

The array with the key of **brdTls** tracks the tiles that have been placed on the board. The array with the key of **brdTks** tracks the tokens that have been placed on the board. The array with the key of **mktTks** tracks the tokens that have been placed on the stock market. And the array with the key of **trayCounts** tracks the number of remaining tiles and tokens available to be played on the game board.

6 Client Side Overview

The client side application is divided into sections by page. In other words, the server can send a number of different pages to the client browser. Each such page will have its own associated JavaScript file(s) and its own associated CSS file. Pages that include PHP code will have the extension 'php'. Other pages will have the extension 'html'.

Currently I have implemented six pages, but more will be added as needed.

6.1 The Index Page

The index.html page is one of the two pages which is not password protected. It is the introductory page to BOARD18 and contains a welcome message. It is currently used for logging in to Board18 and for registering new players. Links to some player ID and password recovery functions should be added soon.

6.2 The Access-denied Page

The access-denied.html page is the other of the two pages which is not password protected. It is the page that handles unauthorized access to password protected pages. It has an access denied message and a button that transfers the player to the index.html page.

6.3 The Main Page

The board18Main.php page displays a list of the logged in player's active games. The player can select any of the listed games to start a game session in that game. He can also select a link to the board18New page if he wishes to start a new game. A link may be added in the future for adding or modifying game boxes.

6.4 The New Game Page

The board18New.php page contains a form that can be used to create a new BOARD18 game session. It displays a list of the available game boxes and a list of the available players as aids to filling out the form.

6.5 The Board Page

The board page consists mainly of the game board [or map] and the left sidebar containing the tile and token trays. This is an overview of its main features.

6.5.1 Main Menu

The main menu drops down from the header devision. Actually this is true for all BOARD18 pages. It has sub menu's that can be used for navigation and for other page specific actions.

6.5.2 Selecting Trays

The left sidebar contains the active tray beneath a drop down tray selection menu. Clicking on the name of a tray makes that tray the active tray.

6.5.3 Selecting Tiles from Tray

The tiles or tokens in the active tray are displayed with a count of available items in the upper left corner. If the count is zero then the tile is grayed out and cannot be selected. Clicking on a tile in the active tray will select that tile [it will be highlighted with a red background].

6.5.4 Dropping Tiles on Board

When a tile on the active tray is highlighted [selected], it can be placed on a map hex by clicking on the center of the hex. A tile will be centered on the hex. A token will be centered on the point that is clicked. The dropped item has NOT yet been permanently placed and can be further adjusted before it is.

6.5.5 Rotating Tiles

Clicking on a dropped tile before the move has been submitted will cause the tile to be rotated clockwise. There are also menu items that can be used to rotate a tile in this situation.

6.5.6 Moving Tokens

Clicking on a hex containing a dropped token before the move has been submitted will cause the token to be repositioned to the point of the click.

6.5.7 Submitting Moves

A move in progress will be submitted to the server [database] when the **Accept Move** menu item is clicked. A move in progress will be discarded when the **Cancel Move** menu item is clicked.

6.5.8 The Drop Down Menu

To be provided at a later date.

6.6 The Stock Market Page

To be provided at a later date.

7 Server Side Overview

The server side of the BOARD18 application will use a minimalist approach. In other words, anything that can be done on the client side, will be done on the client side. This philosophy limits the server mostly to data base access and game logging tasks. Conflict resolution for multi thread database access will also be supplied on a game by game basis.

7.1 Data Base

BOARD 18 uses MySQL for its data base access method. A single database, with a data base name of BOARD18, is used. The database consists of four tables:

> The Box Table

> The Game Table

> The Game Player Table

> The Players Table

7.1.1 The Box Table

Each row in the table named “box” contains a game box. The table layout is shown below. The json_text field contains a stringified JSON game box object as described in section 5.2.1.

Column	Type	Null	Default
box_id	Int (11)	No	
bname	Varchar (25)	No	
version	Varchar (25)	Yes	NULL
create_date	Timestamp	Yes	NULL
activity_date	Timestamp	No	CURRENT_TIMESTAMP
author	Varchar (25)	No	
json_text	Text	Yes	NULL

7.1.2 The Game Table

The “game” table contains a row for each game that is being tracked by BOARD18. The table layout is shown below. The json_text field contains a stringified JSON game session object as described in section 5.2.2.

Column	Type	Null	Default
game_id	Int (11)	No	
gname	Varchar (25)	No	
start_date	Timestamp	Yes	NULL
activity_date	Timestamp	No	CURRENT_TIMESTAMP
update_counter	Int (11)	No	0
last_updater	Varchar (16)	No	0
box_id	Int (11)	No	
json_text	Text	Yes	NULL

7.1.3 The Game Player Table

The “game_player” table contains a row for each player of each active game that is being tracked by BOARD18. The table layout is shown below. This table is the link between players and the game sessions that they are playing.

Column	Type	Null	Default
game_id	Int (11)	No	
player_id	Int (11)	No	
status	Varchar (10)	No	'New'

7.1.4 The Players Table

The "players" table contains a row for each player that has a login ID in BOARD18. The table layout is shown below. This table is used to control player access to the application.

Column	Type	Null	Default
player_id	Int (11)	No	
firstname	Varchar (25)	No	
lastname	Varchar (25)	Yes	NULL
email	Varchar (25)	Yes	NULL
login	Varchar (16)	No	
passwd	Varchar (64)	No	
create-date	Timestamp	No	CURRENT_TIMESTAMP
level	Varchar (10)	No	player

7.2 Logic

Most of the server side processing for BOARD18 is done in stand alone PHP programs that are executed from the web pages via AJAX calls. These programs are listed here with brief descriptions.

- createGame.php — Creates a new row in the game table and adds a row for each player in the game_player table.
- gameBox.php — Returns a game box in JSON format.
- gameSession.php — Returns game activity information in JSON format.
- logout.php — Logs the current user out of BOARD18.
- myGameList.php — Returns a list of the games that a player is playing.
- newUser.php — Creates a new row in the players table.
- updateGame.php — Loads game status updates in the database.
- validateUser.php — Logs the current user in to BOARD18.

In addition the auth.php program is included at the front of every php program that is listed above and of every php enabled page. This module checks that the session is signed on and redirects control to access-denied.html if it is not.

Finally the config.php module is included at the front of every php program and php enabled page that accesses the board18 database. It contains information used to locate and access the database.