Lisp in Summer Projects Submission

Submission Date	2013-10-03 12:22:33
Full Name	Leo Zovic
Country	Canada
Project Name	Deal
Type of software	web app
General category	game
LISP dialect	Commmon Lisp
GitHub URL	https://github.com/lnaimathi/deal
Did you start this project?	Yes, all the code is written by me
Project Description	I want to describe my project in this form.
Purpose	It's a web-based prototyping and play-testing tool for tabletop card and board games.
Function	It tightens the prototype/test/revise loop for game designers, delays the need for physical prototypes to be put together, and removes the need for face-to-face interaction for play-testing. In technical terms, it runs a server that play-testers/designers can connect to and simulates a tabletop as simply as possible.
Motivation	A friend of mine and I have some ideas for card games we want to develop and we currently live on opposite sides of Canada. Building a web app is cheaper and easier than moving to BC or flying the round trip every week or so.
Audience	The project can be used by anyone who wants to play-test/prototype card and board games digitally, but as stated in "Motivation", I built this for me. Because of how general the tabletop simulation is, it's also possible to use this to play things like chess or go online, so tabletop players are another potential audience (but that was something like a fourth or fifth goal).
Methodology	Kind of touched on this in "Function". It's a web server with no sign-up because it needs to be as easy as possible to

let play-testers jump into a game (and because I was experimenting a bit to see if I could build a web application that respects the users freedom). The obvious tools for that in Common Lisp are the Hunchentoot web server, cl-who and parenscript. Given that stack, the challenge is putting together a web application that can interact with players without them having to submit a request. Because Hunchentoot uses a thread-per-request model, it would have been difficult to have it serve up SSE connections or do long-polling, so I decided to cheat. The application is going to be deployed behind a reverse-proxy for static file-serving/SSL purposes anyway, so I use the nginx push-stream module to publish event streams instead of having Hunchentoot do that work. That's what makes configuration a bit of a pain; it means having to compile your own nginx with push-stream support.

This is also the first large-scale test of some theories I had about self-documenting APIs and clean separation of server from client. As a result, you can think of Deal as two projects; a game server and a reference client implementation focused on prototyping. The API is contained entirely in the file deal.lisp, and tries hard to be very obvious about what each handler expects as well as what it does. Other back-end plumbing can be found in define-handler.lisp and the model/ folder. The client implementation is contained in a sub-project called deal-ui, which consists entirely of the deal-ui/ folder. Some general utility and jQuery interaction macros are defined in pQuery.lisp, the CSS is in the obvious place, and the front-end itself is implemented entirely in the file deal-ui/deal-ui.lisp.

If you'd like more detail than the above, I did keep a fairly detailed journal as I went. It can be found here:

http://langnostic.blogspot.com/2013/08/deal-journal-part-one.html

http://langnostic.blogspot.com/2013/09/deal-part-2.html

http://langnostic.blogspot.com/2013/09/deal-journal-interlude-one-treatise-on.html

http://langnostic.blogspot.com/2013/09/deal-journal-part-three.html

Conclusion

Mission accomplished as far as the prototyping/play-testing go.

That works well, and both my friend and I have been using it to work up card game prototypes. I've also been showing it around to some local game designers, and their reaction has by and large been positive.

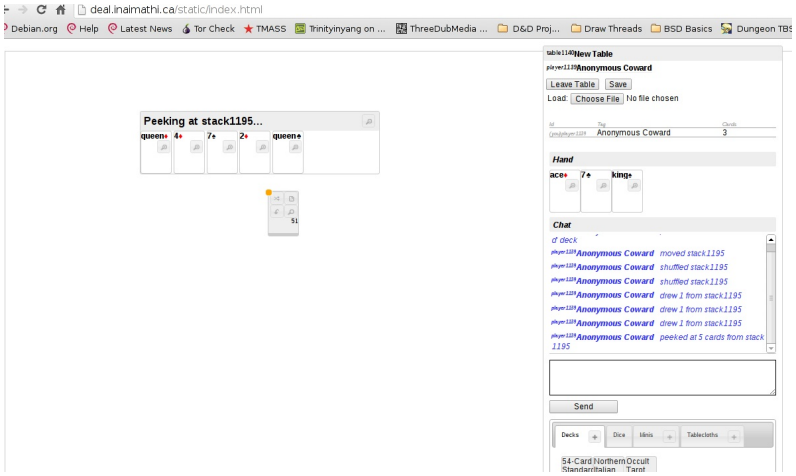
You can see some of the remaining issues at the projects issue tracker here:

https://github.com/lnaimathi/deal/issues

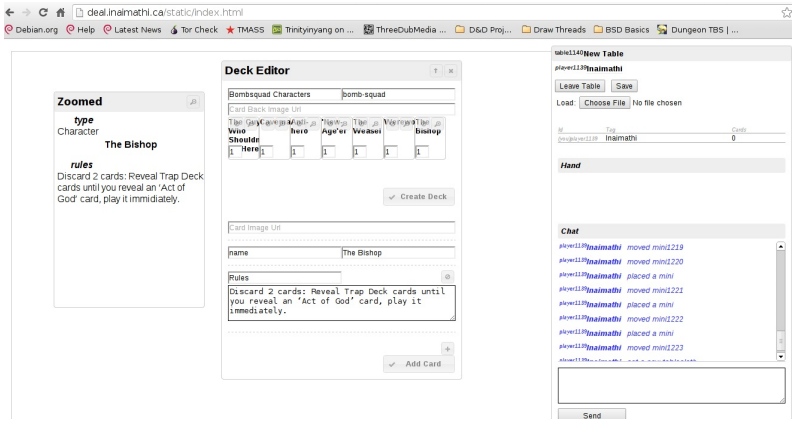
The big ones not listed there are one thing I want to do (write a custom web server specifically for the purpose of hosting web games, with an emphasis on simplicity, ease of deployment and session/SSE handling), and a few

	experiments I still want to run (writing front-ends without resorting to wrapping jQuery, and writing some special-purpose front ends. For instance, a simple one focused on playing go or some other specific public-domain game).
Build Instructions	This is covered in detail in the installation section of README.md. It's included with the code, and can be seen here https://github.com/lnaimathi/deal#installation It's complicated because the project currently depends on an external SSE publishing service to work properly. The one I developed against is the nginx push-stream module https://github.com/wandenberg/nginx-push-stream-module which means that in order to get a copy of the live server (http://deal.inaimathi.ca) running on your local, you need to compile your own nginx with push-stream support, then fiddle with the nginx.config file for a bit. There are instructions on precisely how to do that in the README.md, but the process is involved.
Test Instructions	There is no test suite. You could play it, I guess? I've been doing incremental testing the whole way through, so I know the things I want to do with it are possible at least...
Execution Instructions	Once you've got it installed, just 1. run nginx 2. run a lisp from the deal/ directory 3. load up the :deal-ui project I'm hoping to get the installation process down to this level of simplicity too.
Describe any bugs or caveats	Bugs are listed at the issues page https://github.com/lnaimathi/deal/issues The relevant ones I haven't documented there are: 1. Certain droppables are being a bit wonky. In particular the peek interface still needs a bit of work. 2. Overlay elements are mislabeled (no functionality suffers from this, it's just a label error) 3. Deck editor buttons are overlapping card name preview (again, purely cosmetic error) The first two have been patched on the live server as of this writing, but they were still outstanding when I made the tag for this contest submission.

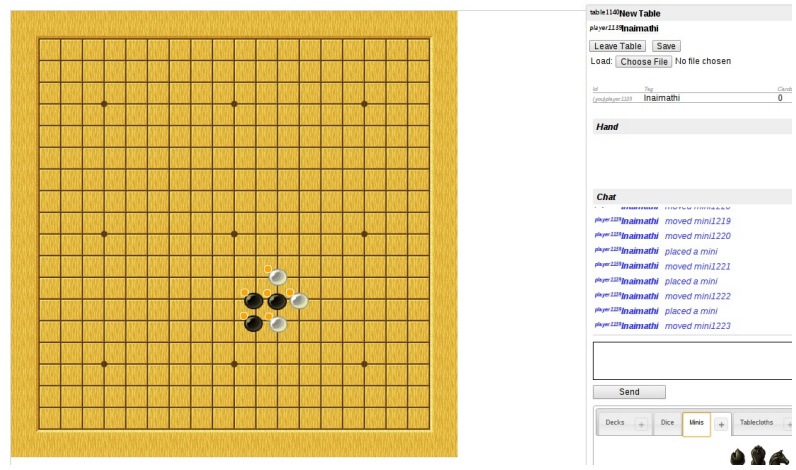
Screen shots



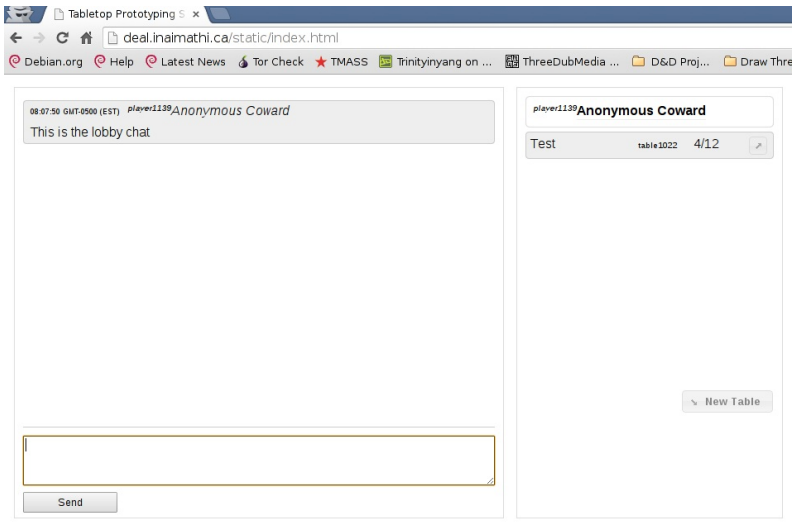
deal--card-game-peeking.png



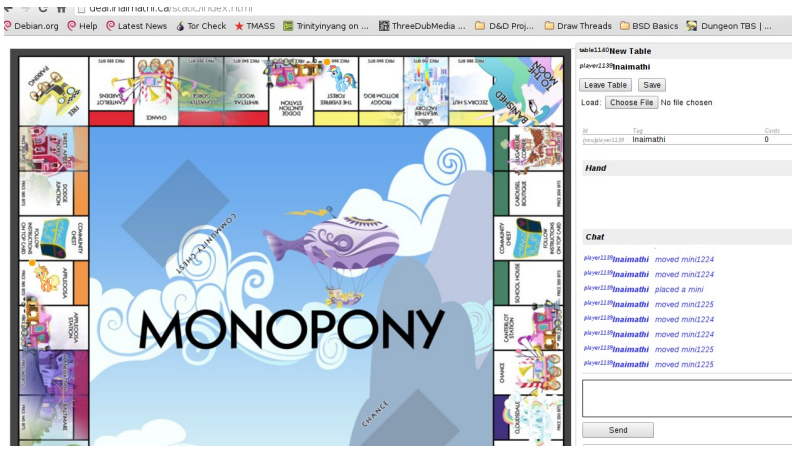
deal--deck-editor_zoomed-card.png



deal--go-game.png



deal--lobby.png



deal--monopony-setup.png

Official

I have read rules and have abided by them.
I am 18 years of age or older.
I am not living in Brazil, Quebec, Saudi Arabia, Cuba, Iran, Myanmar (Burma), North Korea, Sudan, or Syria.