Finite Channels

My Solution (with Peng Li)

data FiniteChan a = FiniteChan {
    readCell :: MVar (Stream a)
, writeCell :: MVar (Stream a)
}

type Stream a = [MVar a]

newFiniteChan :: Int -> IO (FiniteChan a)
newFiniteChan size = do
    q <- replicateM size newEmptyMVar
    r <- newMVar (cycle q)
    w <- newMVar (cycle q)
    return (FiniteChan r w)

-- where cycle l = l ++ l ++ l ++ ...

writeFiniteChan :: FiniteChan a -> a -> IO ()
writeFiniteChan (FiniteChan r w) x = do
    (hd:tl) <- takeMVar w
    putMVar hd x
    putMVar w tl

readFiniteChan :: FiniteChan a -> IO a
readFiniteChan (FiniteChan r w) = do
    (hd:tl) <- takeMVar r
    x <- takeMVar hd
    putMVar r tl
    return x

Final Projects
### Schedule

<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apr 9</td>
<td>STM</td>
</tr>
<tr>
<td>Apr 14</td>
<td>Composing Financial Contracts (guest lecture)</td>
</tr>
<tr>
<td>Apr 16</td>
<td>Generalized Algebraic Datatypes (guest lecture)</td>
</tr>
<tr>
<td>Apr 21</td>
<td>Final project presentations</td>
</tr>
<tr>
<td>Apr 23</td>
<td>Final project presentations</td>
</tr>
<tr>
<td>Apr 28</td>
<td>Final project presentations</td>
</tr>
</tbody>
</table>

### Final Project Milestones

<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apr 9</td>
<td>STM</td>
</tr>
<tr>
<td>Apr 14</td>
<td>Composing Financial Contracts (guest lecture)</td>
</tr>
<tr>
<td>Apr 16</td>
<td>Generalized Algebraic Datatypes (guest lecture)</td>
</tr>
<tr>
<td></td>
<td>Initial design document (3-page document)</td>
</tr>
<tr>
<td>Apr 21</td>
<td>Final project presentations</td>
</tr>
<tr>
<td>Apr 23</td>
<td>Final project presentations</td>
</tr>
<tr>
<td></td>
<td>Working prototype of core functionality (screenshots)</td>
</tr>
<tr>
<td>Apr 28</td>
<td>Final project presentations</td>
</tr>
<tr>
<td></td>
<td>Code reviews (*)</td>
</tr>
</tbody>
</table>

When? (**): Do people want to do this?

(**) We need to choose a date between April 30 and May 5

### Project Logistics

- Groups of 2 (probably with one group of 3)
- Work with any partner(s) you want—it’s fine if you’ve already worked together on a project this semester
- Use any language or combination of languages, so long as the bulk of the code is written in some functional language (Haskell, OCaml, Scheme, etc.)
- Aim to spend approximately 30 hours total

### Project Topics

- Carte blanche: Choose any programming problem that interests you

### Default Topic

- I think it would be fun if several groups ended up working on variants of the same idea.
- I propose this one:

  A collaborative virtual environment

  (i.e., a better Second Life / Sims / etc.)

- Building a real one of these is a gigantic task—need to attack just a piece of the problem...
  - 3-d modeling
  - distributed simulation
  - a new “spatial scripting language”
  - etc., etc.