4-tim-baldridge-10K-processes

Benefits

In this exercise you'll gain understanding of the following:

- How core.async processes work in ClojureScript

Assumptions

- You have Leiningen installed.
- You have an internet connection (if you don't have this – then we can copy the maven archive across)
- You have worked through the previous exercises

Code to Read

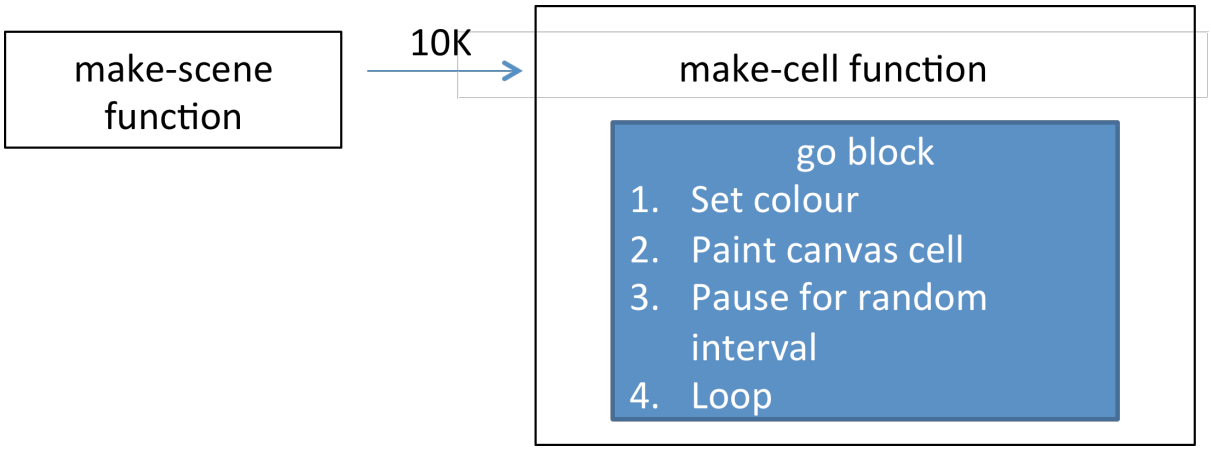
- `lambdajam-2014-core.async-workshop\4-tim-baldridge-10K-processes\baldridge-10000\src\cljs\baldridge_10000\client.cljs`

Things to Note In the Code

1. this is a copy of the code by Timothy Baldridge at: https://github.com/halgari/clojure-conj-2013-core.async-examples/blob/master/src/clojure_conj_talk/core.clj You can see the talk here: https://www.youtube.com/watch?v=enwIIGzhahw
2. the *one* `go` block
3. that the `make-scene` function calls the `make-cell` function 10K times
4. that 10K `go` blocks are created
5. that the `go` block sets a random colour
6. that the `go` block draws a coloured square on the canvas
7. that the `go` block pauses for a random internal max 1 second
8. that potentially each `go` block pauses for a different interval between updates

Code Model

This is a quick way to understand what is going on in the code:



Activities

1. Open the page `lambdajam-2014-core.async-workshop\4-tim-baldridge-10K-processes\baldridge-10000\index.html` in your web browser
2. Leave it open in a tab as we do the next activity

Questions for Reflection

1. How would you solve a problem that required concurrency in JavaScript?
2. Could this be replaced by a simple for loop that did the updates in a single pass?