5-not-10K-processes - 10K-no-async

Benefits

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

- that the core.async function enables simple 'process-level' pauses

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

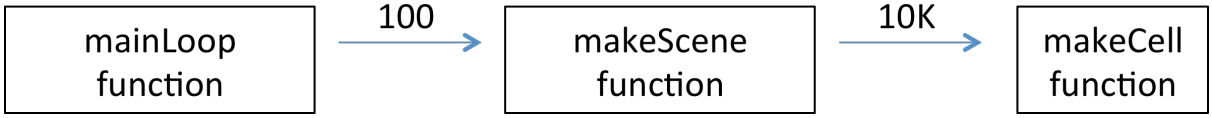
- `lambdajam-2014-core.async-workshop\5-not-10K-processes\10K-no-async\js\10K.cljs`

Things to Note In the Code

1. this is not ClojureScript – but JavaScript (to make a point)
2. `mainLoop` calls `makeScene` 100 times
3. `makeScene` calls `makeCell` 10K times per call.
4. `makeCell` sets a random colour and draws a rectangle on the canvas
5. the only pause function is at the `mainLoop` level

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\5-not-10K-processes\10K-no-async\index.html` in your web browser
2. Compare it to the tab running the previous activity
3. Observe the process-level timings in the first tab, and contrast it with the single loop in the second activity.

Questions for Reflection

1. How would you solve a problem that required concurrency in JavaScript?