

mopoko.com lisp games

John Fremlin

<http://john.freml.in>

2012 June 2

Contents

mopoko.com let's you play games with a custom Lisp webserver - just for fun!

Introduction

Continuation games

Conclusion

Outline

Introduction

Continuation games

Conclusion

History of mopoko.com

- Play rock, paper, scissors and Nash's bargaining game with robots or with other people.
- Just for fun! The project is open-source and on github.com/vii/teepeedee2
- Five years old, an AJAX/COMET demo for Lisp
- Based on the high-performance custom Lisp webserver `tpd2`

History of mopoko.com

- Play rock, paper, scissors and Nash's bargaining game with robots or with other people.
- Just for fun! The project is open-source and on github.com/vii/teepeedee2
- Five years old, an AJAX/COMET demo for Lisp
- Based on the high-performance custom Lisp webserver `tpd2`

History of mopoko.com

- Play rock, paper, scissors and Nash's bargaining game with robots or with other people.
- Just for fun! The project is open-source and on github.com/vii/teepeedee2
- Five years old, an AJAX/COMET demo for Lisp
- Based on the high-performance custom Lisp webserver `tpd2`

teepeedee2

- 10k+ dynamic requests per second on one core (epoll)
- Uses cl-irregsexp for fast text processing
- Custom HTML, and CSS templating
- COMET framework
- Serves mopoko.com and blogs

teepeedee2

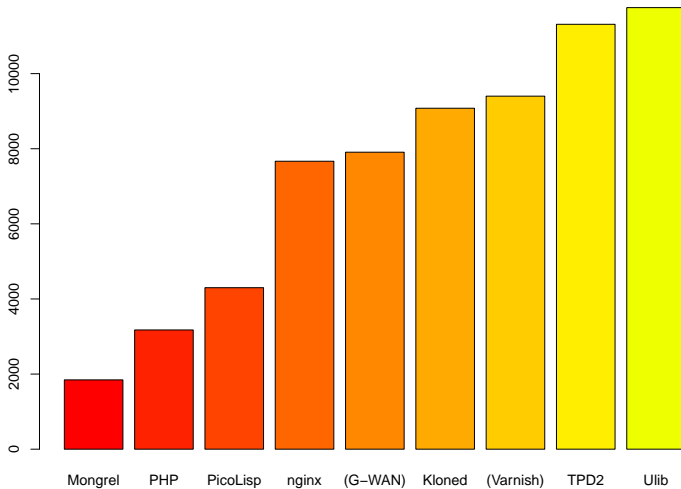
- 10k+ dynamic requests per second on one core (epoll)
- Uses cl-irregsexp for fast text processing
- Custom HTML, and CSS templating
- COMET framework
- Serves mopoko.com and blogs

teepee2 code sample

```
(defsite *balisp*)  
(with-site (*balisp*)  
  (defpage "/balisp" (name)  
    (<h1 :style (css-attr  
               :color "red"  
               :font-size "200%")  
         "Bay_Area_Lispers!_Welcome_" name)))
```

teepee2 benchmark

Request throughput on one core (request/s)



Outline

Introduction

Continuation games

Conclusion

cl-cont and continuations

- Web-based application necessarily stateless
- Game rules naturally stateful
- Transformation to continuation passing-style via cl-cont bridges the gap

cl-cont and continuations

- Web-based application necessarily stateless
- Game rules naturally stateful
- Transformation to continuation passing-style via cl-cont bridges the gap

defrules

```
(my-defun roshambo 'play ()
  (with-game
    (my new-state)
    (with-join-spawn/cc ()
      (loop for p in (my players)
            do
              (let ((p p))
                (spawn/cc ()
                  (setf (its choice p)
                        (my secret-move
                          :select p
                          '(:one
                            rock
                            paper
                            scissors))))))))
    ...))
```

Different players

- Define robots and web-players via Common Lisp Object System
- Multiple dispatch
 - Asynchronous response
 - Could build a generic robot that can play every game?

Different players

- Define robots and web-players via Common Lisp Object System
- Multiple dispatch
- Asynchronous response
- Could build a generic robot that can play every game?

Different players

- Define robots and web-players via Common Lisp Object System
- Multiple dispatch
- Asynchronous response
- Could build a generic robot that can play every game?

move example

```
(defmethod move ((controller robot-sensible)
                 (player-state truc-player)
                 (move-type (eq! :accept-new-stake))
                 choices
                 &rest args)
  (declare (ignore choices))
  (let ((new-stake (getf args :new-stake)))
    (> (* new-stake
          (- 1 (its win-probability player-state)))
        (its stake (its game player-state)))))
```

Outline

Introduction

Continuation games

Conclusion

Currently a tech demo

- Small number of simple demo games
- Showing a fun application of meta-programming
- And low-level Lisp on Linux
- But it's fairly production ready - working system with long uptime

Currently a tech demo

- Small number of simple demo games
- Showing a fun application of meta-programming
- And low-level Lisp on Linux
- But it's fairly production ready - working system with long uptime

Develop mopoko.com!

My email is john@freml.in. Questions?

Help out! github.com/vii/teepeedee2

Develop mopoko.com!

My email is john@freml.in. Questions?
Help out! github.com/vii/teepeedee2