Temporal semantics for a live coding language

Sam Aaron, Dominic Orchard, Alan Blackwell

@samaaron @dorchard

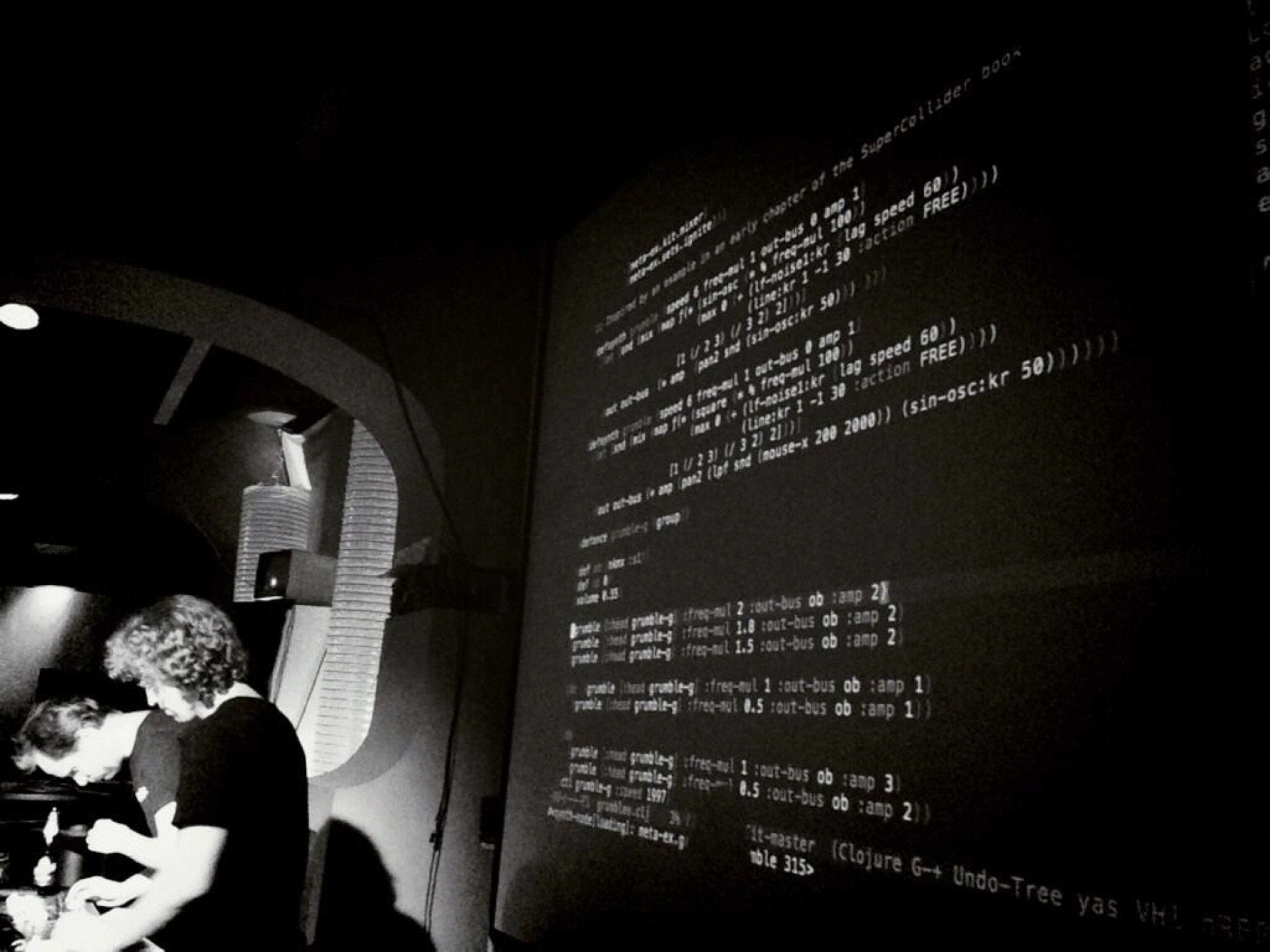
Fibonacci Crisis

UNIVERSITY OF CAMBRIDGE

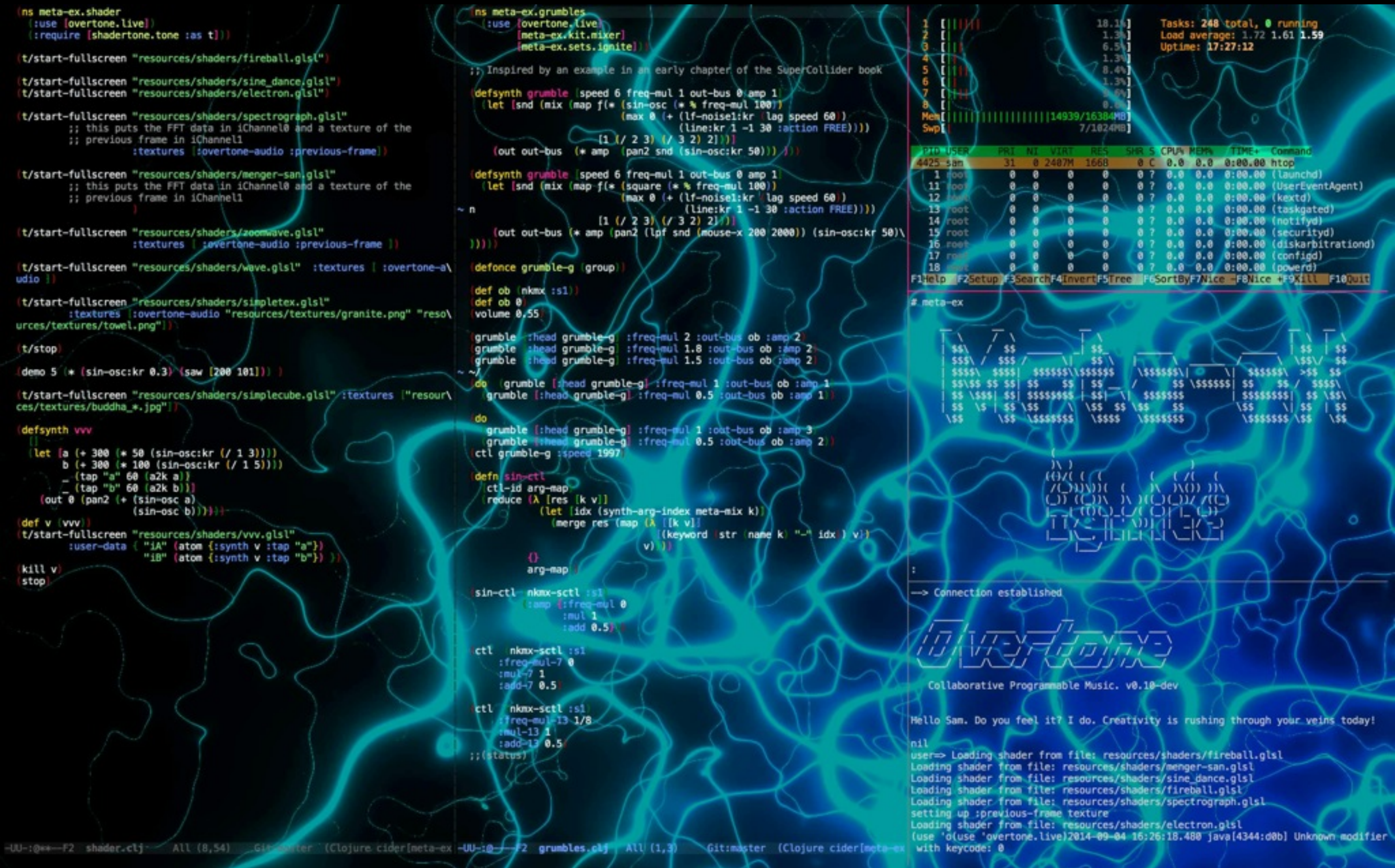
FARM 2014, 6th September 2014, Gothenburg

overtone

http://overtone.github.io

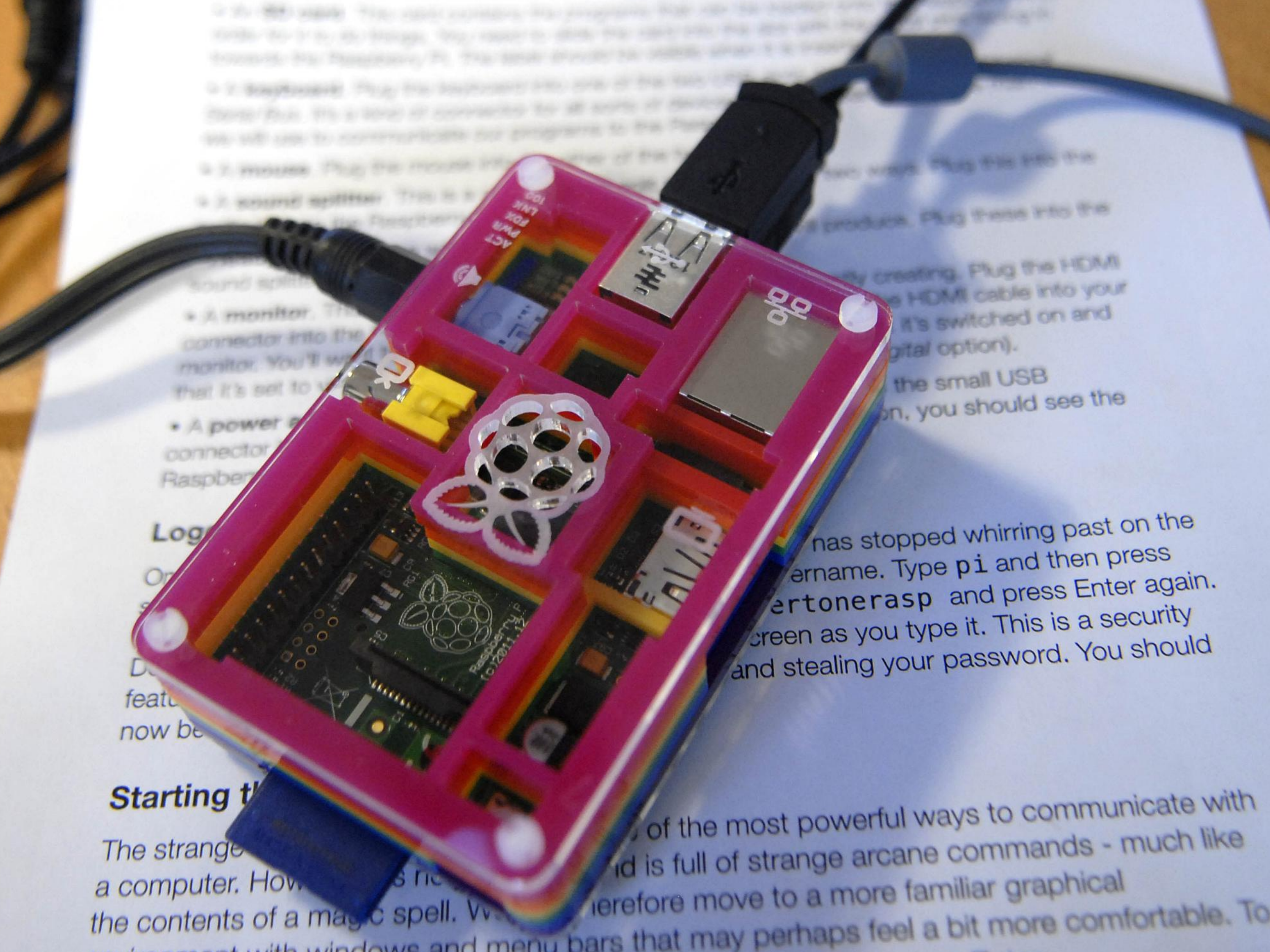


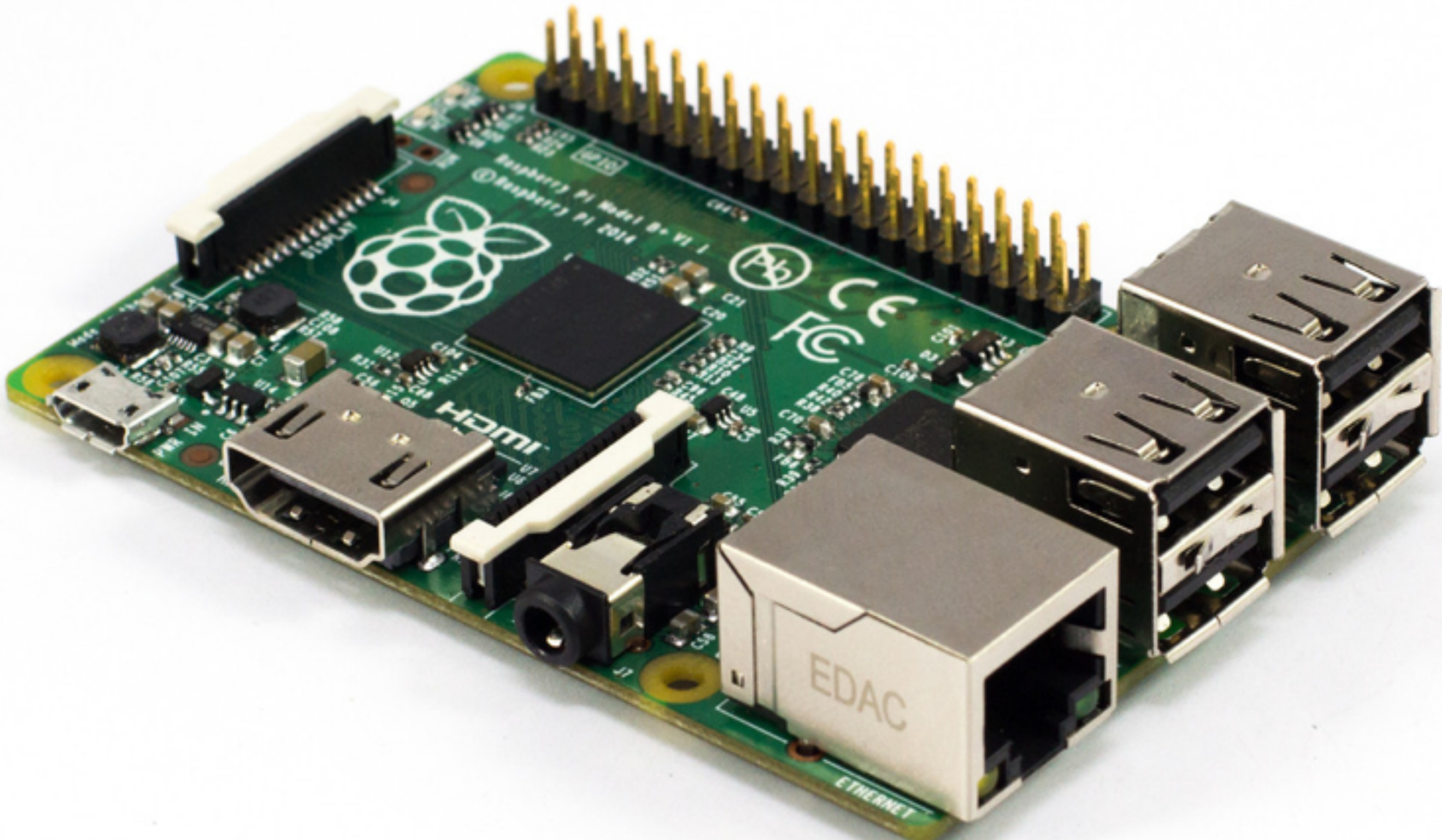
Inspired by an example in an early chapter of the SuperCollider book
speed 6 freq-mul 1 out-bus 0 amp 1
(sin-osc (* % freq-mul 100))
(max 0 (+ (lf-noise1:kr (lag speed 60))
(line:kr 1 -1 30 :action FREE))))
[1 (/ 2 3) (/ 3 2) 2]))
(pan2 snd (sin-osc:kr 50))
speed 6 freq-mul 1 out-bus 0 amp 1
(square (* % freq-mul 100))
(max 0 (+ (lf-noise1:kr (lag speed 60))
(line:kr 1 -1 30 :action FREE))))
[1 (/ 2 3) (/ 3 2) 2]))
(pan2 (lpf snd (mouse-x 200 2000)) (sin-osc:kr 50))))))
(out out-bus (* amp
(group)
:freq-mul 2 :out-bus ob :amp 2)
grumble grumble-g :freq-mul 1.8 :out-bus ob :amp 2)
grumble grumble-g :freq-mul 1.5 :out-bus ob :amp 2)
grumble grumble-g :freq-mul 1 :out-bus ob :amp 1)
grumble grumble-g :freq-mul 0.5 :out-bus ob :amp 1))
grumble grumble-g :freq-mul 1 :out-bus ob :amp 3)
grumble grumble-g 0.5 :out-bus ob :amp 2))
grumble-g :speed 1997
meta-ex.g
it-master (Clojure G-+ Undo-Tree yas
mble 315>



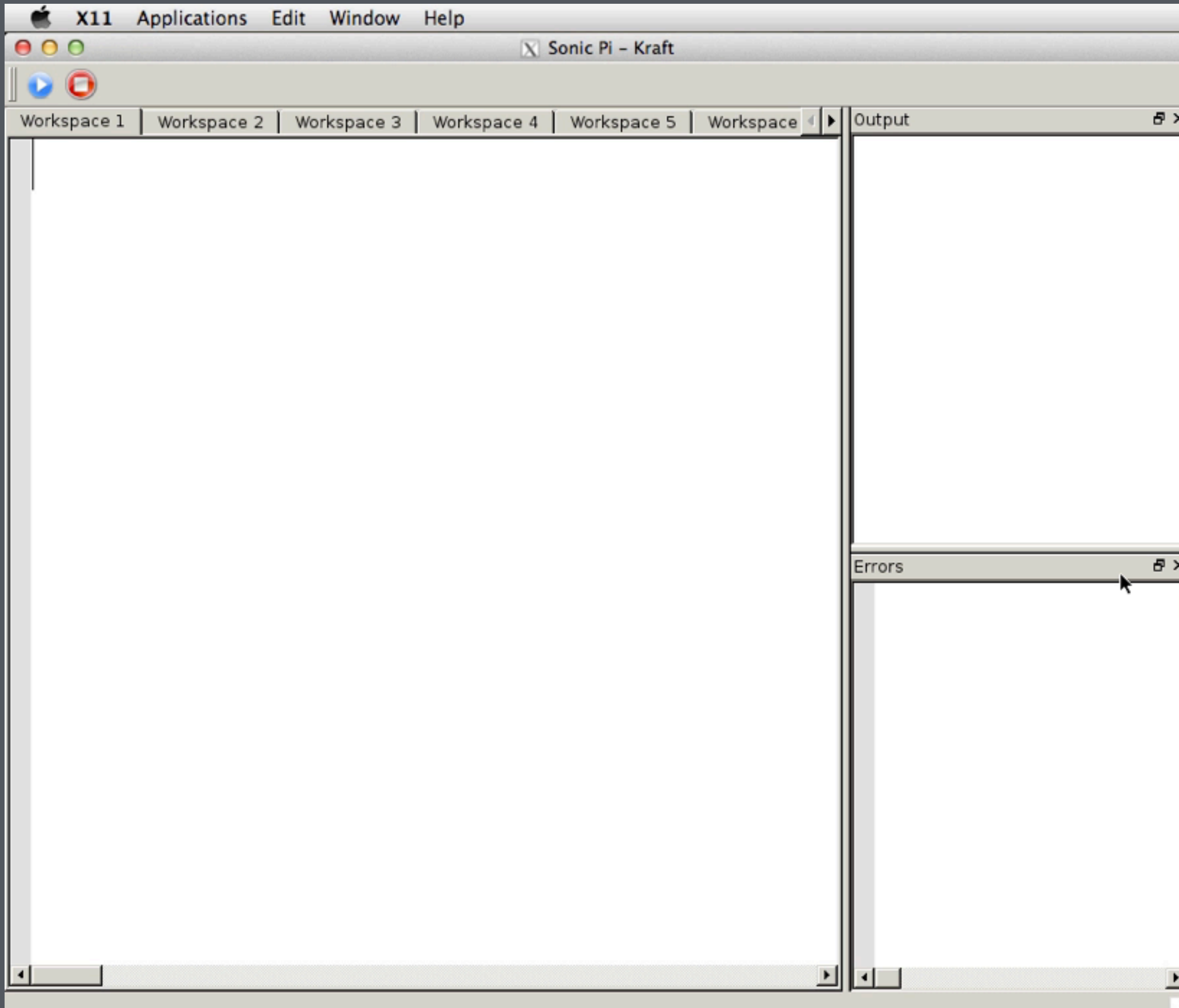
(ns meta-ex.shader
(:use [overtone.live])
(:require [shadertone.tone :as t]))
(t/start-fullscreen "resources/shaders/fireball.glsl")
(t/start-fullscreen "resources/shaders/sine_dance.glsl")
(t/start-fullscreen "resources/shaders/electron.glsl")
(t/start-fullscreen "resources/shaders/spectrograph.glsl"
;; this puts the FFT data in iChannel0 and a texture of the
;; previous frame in iChannel1
:textures [:overtone-audio :previous-frame])
(t/start-fullscreen "resources/shaders/menger-san.glsl"
;; this puts the FFT data in iChannel0 and a texture of the
;; previous frame in iChannel1
)
(t/start-fullscreen "resources/shaders/zoomwave.glsl"
:textures [:overtone-audio :previous-frame])
(t/start-fullscreen "resources/shaders/wave.glsl" :textures [:overtone-a\
udio])
(t/start-fullscreen "resources/shaders/simpletex.glsl"
:textures [:overtone-audio "resources/textures/granite.png" "reso\
urces/textures/towel.png"])
(t/stop)
(demo 5 (* (sin-osc:kr 0.3) (saw [200 101])))
(t/start-fullscreen "resources/shaders/simplecube.glsl" :textures ["resour\
ces/textures/buddha_*.jpg"])
(defsynth vvv
[]
(let [a (+ 300 (* 50 (sin-osc:kr (/ 1 3))))
b (+ 300 (* 100 (sin-osc:kr (/ 1 5))))
_ (tap "a" 60 (a2k a))
_ (tap "b" 60 (a2k b))]
(out 0 (pan2 (+ (sin-osc a)
(sin-osc b))))))
(def v (vvv))
(t/start-fullscreen "resources/shaders/vvv.glsl"
:user-data { "iA" (atom {:synth v :tap "a"})
"iB" (atom {:synth v :tap "b"}) })
(kill v)
(stop)
-UU-:@**--F2 shader.clj All (8,54) Git:master (Clojure cider[meta-ex
(ns meta-ex.grumbles
(:use [overtone.live]
[meta-ex.kit.mixer]
[meta-ex.sets.ignite]))
;; Inspired by an example in an early chapter of the SuperCollider book
(defsynth grumble [speed 6 freq-mul 1 out-bus 0 amp 1]
(let [snd (mix (map #(* (sin-osc (* % freq-mul 100))
(max 0 (+ (lf-noise1:kr (lag speed 60))
(line:kr 1 -1 30 :action FREE))))
[1 (/ 2 3) (/ 3 2) 2]))]
(out out-bus (* amp (pan2 snd (sin-osc:kr 50))))))
(defsynth grumble [speed 6 freq-mul 1 out-bus 0 amp 1]
(let [snd (mix (map #(* (square (* % freq-mul 100))
(max 0 (+ (lf-noise1:kr (lag speed 60))
(line:kr 1 -1 30 :action FREE))))
[1 (/ 2 3) (/ 3 2) 2]))]
(out out-bus (* amp (pan2 (lpf snd (mouse-x 200 2000)) (sin-osc:kr 50)\
)))))
(defonce grumble-g (group))
(def ob (nkmx :s1))
(def ob 0)
(volume 0.55)
(grumble :head grumble-g :freq-mul 2 :out-bus ob :amp 2)
(grumble :head grumble-g :freq-mul 1.8 :out-bus ob :amp 2)
(grumble :head grumble-g :freq-mul 1.5 :out-bus ob :amp 2)
(do (grumble [:head grumble-g] :freq-mul 1 :out-bus ob :amp 1)
(grumble [:head grumble-g] :freq-mul 0.5 :out-bus ob :amp 1))
(do
(grumble [:head grumble-g] :freq-mul 1 :out-bus ob :amp 3)
(grumble [:head grumble-g] :freq-mul 0.5 :out-bus ob :amp 2))
(ctl grumble-g :speed 1997)
(defn sin-ctl
[ctl-id arg-map]
(reduce (λ [res [k v]]
(let [idx (synth-arg-index meta-mix k)]
(merge res (map (λ [[k v]]
[(keyword (str (name k) "-" idx)) v])
v))))
{}
arg-map))
(sin-ctl nkmx-sctl :s1
{:amp {:freq-mul 0
:mul 1
:add 0.5}})
(ctl nkmx-sctl :s1
:freq-mul-7 0
:mul-7 1
:add-7 0.5)
(ctl nkmx-sctl :s1
:freq-mul-13 1/8
:mul-13 1
:add-13 0.5)
;;(status)
-UU-:@---F2 grumbles.clj All (1,3) Git:master (Clojure cider[meta-ex
Tasks: 248 total, 0 running
Load average: 1.72 1.61 1.59
Uptime: 17:27:12
Mem[14939/16384MB]
Swp[7/1024MB]
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
4425 sam 31 0 2407M 1668 0 C 0.0 0.0 0:00.00 htop
1 root 0 0 0 0 0 ? 0.0 0.0 0:00.00 (launchd)
11 root 0 0 0 0 0 ? 0.0 0.0 0:00.00 (UserEventAgent)
12 root 0 0 0 0 0 ? 0.0 0.0 0:00.00 (kextd)
13 root 0 0 0 0 0 ? 0.0 0.0 0:00.00 (taskgated)
14 root 0 0 0 0 0 ? 0.0 0.0 0:00.00 (notifyd)
15 root 0 0 0 0 0 ? 0.0 0.0 0:00.00 (securityd)
16 root 0 0 0 0 0 ? 0.0 0.0 0:00.00 (diskarbitrationd)
17 root 0 0 0 0 0 ? 0.0 0.0 0:00.00 (configd)
18 root 0 0 0 0 0 ? 0.0 0.0 0:00.00 (powerd)
F1Help F2Setup F3Search F4Invert F5Tree F6SortBy F7Nice -F8Nice +F9Kill F10Quit
meta-ex
--> Connection established
Collaborative Programmable Music. v0.10-dev
Hello Sam. Do you feel it? I do. Creativity is rushing through your veins today!
nil
user=> Loading shader from file: resources/shaders/fireball.glsl
Loading shader from file: resources/shaders/menger-san.glsl
Loading shader from file: resources/shaders/sine_dance.glsl
Loading shader from file: resources/shaders/fireball.glsl
Loading shader from file: resources/shaders/spectrograph.glsl
setting up :previous-frame texture
Loading shader from file: resources/shaders/electron.glsl
(use 'o(use 'overtone.live)2014-09-04 16:26:18.480 java[4344:d0b] Unknown modifier
with keycode: 0

®


ly creating. Plug the HDMI
e HDMI cable into your
it's switched on and
gital option).
the small USB
you should see the
has stopped whirring past on the
ername. Type pi and then press
ertonerasp and press Enter again.
creen as you type it. This is a security
and stealing your password. You should
Starting t
of the most powerful ways to communicate with
The strange
id is full of strange arcane commands - much like
a computer. How
erefore move to a more familiar graphical
the contents of a ma
c spell. W
perhaps feel a bit more comfortable. To


HDMI
EDAC

Sonic Pi


X11
Applications
Edit
Window
Help
Sonic Pi – Kraft
Workspace 1
Workspace 2
Workspace 3
Workspace 4
Workspace 5
Workspace
Output
Errors


DELL
DELL
SENNHEISER


Max File Edit View Object Arrange Options Debug Window Extras Help


Sonic Pi
DELL


DON VALLEY BOWL
10th & 11th JUNE 2011
ARCTIC MONKEYS
MILES KANE
THE VACCINES
ANNA CALVI
MABEL LOVE
DEAD SONS

Defining Pi workshop proves big success

THE Raspberry Pi and a group of lucky artists were the stars of the show at Cambridge Junction on Saturday.

A quirky programme called Defining Pi saw five chosen artists getting to unleash their creative sides by making music with the device.

The workshop, entitled How To Make Music On Raspberry Pi, was held as part of the programme, which is a collaboration between Wysing Arts Centre and the Crucible Network at Cambridge University.

Cambridge Junction hosted the event as part of the Festival of Ideas and Junction University – an initiative offering short artist-led courses, workshops and experiences for the public exploring the intersection of art and life.

Dr Sam Aaron, the technical developer of sonic pi, led the workshop and the artists who participated were Richard Healy, Kate Owens, Rob Smith, Chooc Ly Tan and Dan Tombs.

All five artists are embarking on learning new skills in computer programming in order to help them make new work with the Raspberry Pi.

LET'S HEAR IT: The workshop on how to make music using a Raspberry Pi. Top, Dr Sam Aaron, research associate with Cambridge University, addresses the audience; above right, Sonny Osman, 13, and his mum Janine Woods

Pictures: Warren Gunn

Did firework cause blaze in copse?

A STRAY firework could have caused an area of trees to go up in flames.

One fire crew from Sutton was called to the blaze in Nelsons Lane, Haddenham, at around 7.05pm on Saturday.

They arrived to find an area of trees approximately 20 square metres well alight and a second crew from Cottenham was called for back-up.

Hose reels were used to extinguish the flames and the fire was finally brought under control at 8.15pm.

Crews were due to re-inspect the site yesterday morning to determine the cause of the blaze.

Trailer on A10 overturns

A TRACTOR driver had a lucky escape after the trailer he was towing overturned on the A10.

The incident happened at the Grange Lane roundabout in Littleport at around 9.30am on Saturday.

Emergency services attended but nobody was injured. The road was not closed as a result of the incident, although motorists did face minor delays.

CODE LIKE A

ROCKSTAR

LEARN TO CODE AND MAKE
MUSIC WITH (((Sonic π)))

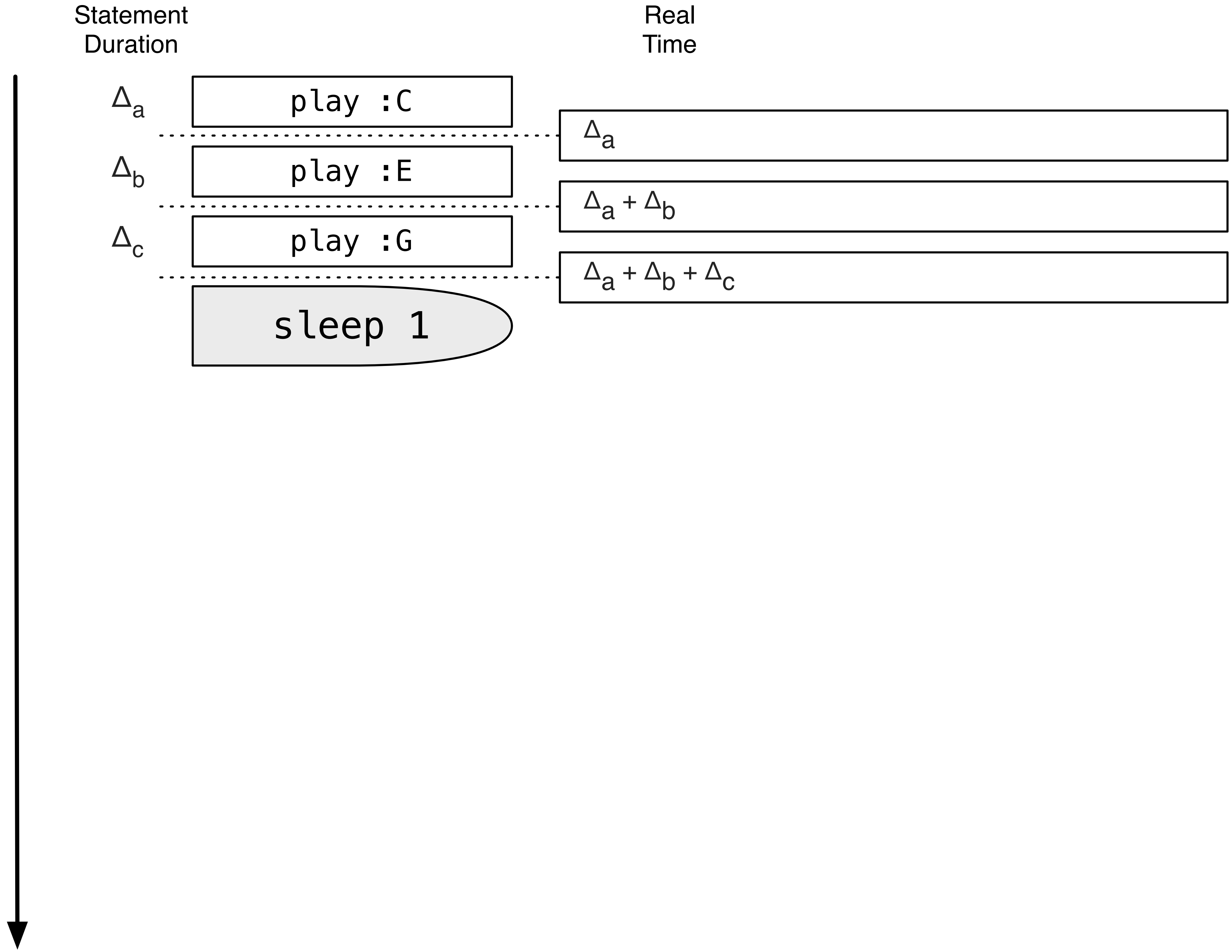
VISIT RASPBERRYPI.ORG TO FIND OUT MORE!

EXCITEDEDUCATOR.COM / ROB-S-WILLIAMS.COM

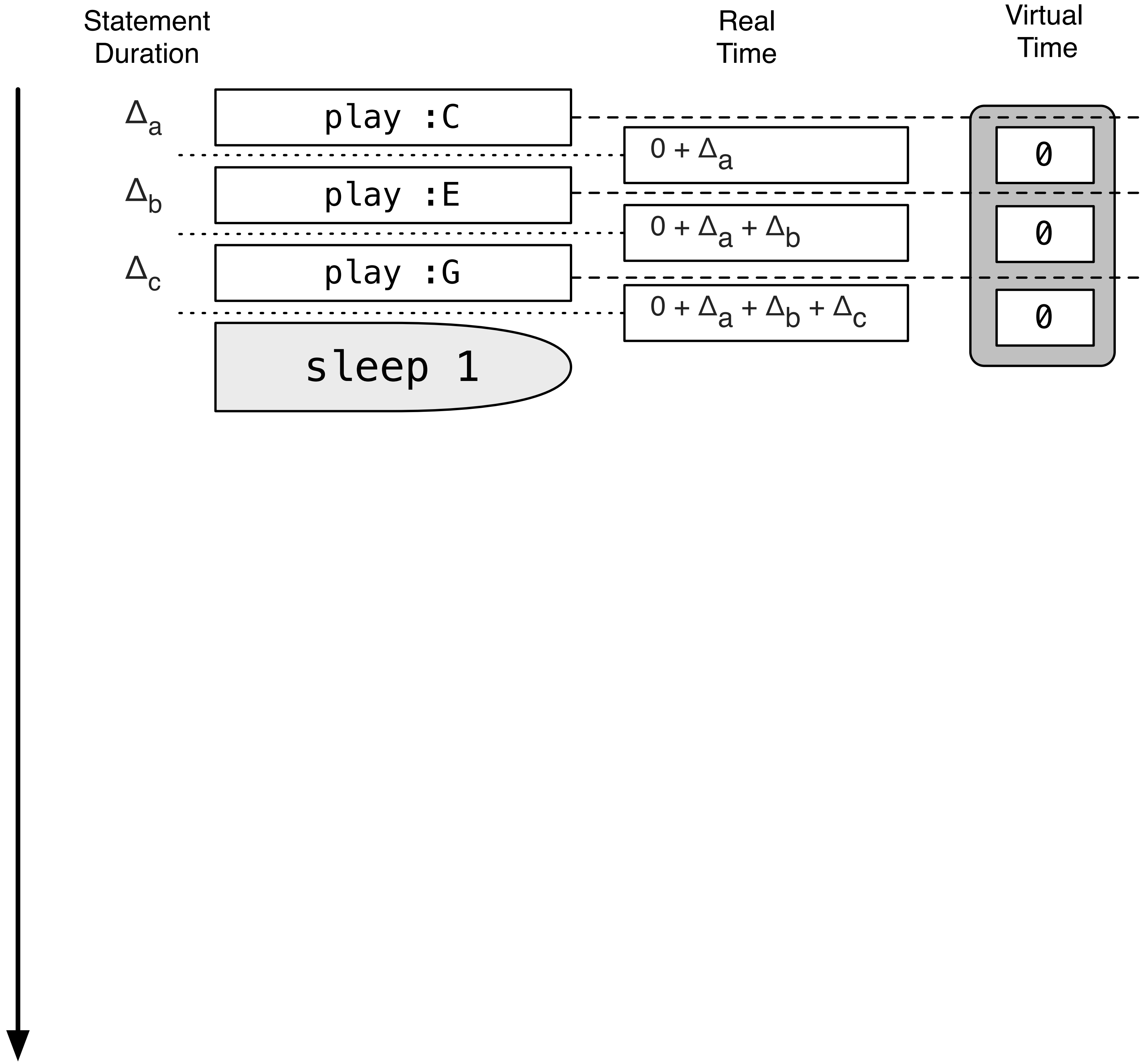

BE PART OF THE UK'S FIRST EVER
LIVE CODING SUMMER SCHOOL!
SONIC PI
LIVE & CODING
MUSIC
TECHNOLOGY
ART
AT CAMBRIDGE JUNCTION
MON 28 JUL – FRI 01 AUG
JUNCTION.CO.UK

Run
Stop
Save
Rec
Size
Size
Align
Output
control archie_ixi, rate: 3
control archie_reverb, room: 0.5
control archie_distort, distort: 0.3
control archie_level, amp: 3
control harvey_ixi, rate: 3
control harvey_distort, distort: 0.000000000000000001
control harvey_level, amp: 1
in_thread(name: :drum_first){loop{drum_first}}
in_thread(name: :drum_cymbals){loop{drum_cymbals}}
in_thread(name: :drum_snares){loop{drum_snares}}


Sonic Pi v2.0
Competition for Schools
Are you the next Daft Punk?
Make sure there's more than a screwdriver in your sonic toolbox.
Sonic Pi is a way to get creative with music and computing. We're hunting down the UK's best young musical coding talent from outer space: is it you? Create a 2-minute or 200-line piece of music on the theme **Space Wonders** with Sonic Pi v2.0 on a Raspberry Pi, and you could be in with a chance of winning one of hundreds of Raspberry Pi kits for yourself and your school, along with workshops with musical artists Juneau Projects, and with live coder-musician Sam Aaron!
Check out **raspberry.org/competitions/sonic-pi** to find out how to enter.
π)))
Sonic Pi



Statement Duration
Real Time
Δ_a
play :C
Δ_b
play :E
Δ_c
play :G
sleep 1
Δ_a
$\Delta_a + \Delta_b$
$\Delta_a + \Delta_b + \Delta_c$



Statement Duration
Real Time
Virtual Time
Δa
Δb
Δc
play :C
play :E
play :G
sleep 1
0 + Δa
0 + Δa + Δb
0 + Δa + Δb + Δc
0
0
0

A formal semantics for sleep

- Abstract interpretation “time system”
- Denotational semantics (via monads)
- Prove “time safety” = prove semantics sound wrt. time system

Paper has the full details

Simplified Sonic Pi *v*2.0 syntax

$$P ::= P; S \mid \emptyset$$

$$S ::= E \mid v = E$$

$$E ::= \texttt{sleep}\ \mathbb{R}_{\geqslant 0} \mid A^i \mid v$$

Time system

$$[-]_{\mathsf{v}} \;:\; \text{virtual time}$$

$$[\emptyset]_{\mathsf{v}} = 0$$

$$[P; v = E]_{\mathsf{v}} = [P]_{\mathsf{v}} + [E]_{\mathsf{v}}$$

$$[\texttt{sleep}\ t]_{\mathsf{v}} = t$$

$$[A^i]_{\mathsf{v}} = 0$$

$$[-]_{\mathsf{t}} \;:\; \text{actual time}$$

$$[\emptyset]_{\mathsf{t}} \approx 0$$

$$[P; \texttt{sleep}\ t]_{\mathsf{t}} \approx ([P]_{\mathsf{v}} + t)\ \mathit{max}\ [P]_{\mathsf{t}}$$

$$[P; v = A^i]_{\mathsf{t}} \approx [P]_{\mathsf{t}} + [A^i]_{\mathsf{t}}$$

e.g. $P; \texttt{sleep}\ 2$ **where** $[P]_{\mathrm{t}} = 1,\ [P]_{\mathrm{v}} = 0$

$$\therefore\ [P; \texttt{sleep}\ 2]_{\mathsf{t}} = (0 + 2)\ \mathit{max}\ 1 = 2$$

$$[P; \texttt{sleep}\ 2]_{\mathsf{v}} = 2$$

Time system

$[\text{—}]_{\mathsf{v}}$: virtual time

$$[\emptyset]_{\mathsf{v}} = 0$$
$$[P; v = E]_{\mathsf{v}} = [P]_{\mathsf{v}} + [E]_{\mathsf{v}}$$
$$[\texttt{sleep}\ t]_{\mathsf{v}} = t$$
$$[A^i]_{\mathsf{v}} = 0$$

$[\text{—}]_{\mathsf{t}}$: actual time

$$[\emptyset]_{\mathsf{t}} \approx 0$$
$$[P; \texttt{sleep}\ t]_{\mathsf{t}} \approx ([P]_{\mathsf{v}} + t)\ \mathit{max}\ [P]_{\mathsf{t}}$$
$$[P; v = A^i]_{\mathsf{t}} \approx [P]_{\mathsf{t}} + [A^i]_{\mathsf{t}}$$

e.g. $P; \texttt{sleep}\ 1$ **where** $[P]_{\mathrm{t}} = 2,\ [P]_{\mathrm{v}} = 0$

$$\therefore\ [P; \texttt{sleep}\ 1]_{\mathsf{t}} = (0 + 1)\ \mathit{max}\ 2 = 2$$
$$[P; \texttt{sleep}\ 1]_{\mathsf{v}} = 1$$

Time system

$$[-]_{\mathsf{v}} \ : \text{virtual time}$$

$$[\emptyset]_{\mathsf{v}} = 0$$

$$[P; v = E]_{\mathsf{v}} = [P]_{\mathsf{v}} + [E]_{\mathsf{v}}$$

$$[\texttt{sleep}\ t]_{\mathsf{v}} = t$$

$$[A^i]_{\mathsf{v}} = 0$$

$$[-]_{\mathsf{t}} \ : \text{actual time}$$

$$[\emptyset]_{\mathsf{t}} \approx 0$$

$$[P; \texttt{sleep}\ t]_{\mathsf{t}} \approx ([P]_{\mathsf{v}} + t)\ \mathit{max}\ [P]_{\mathsf{t}}$$

$$[P; v = A^i]_{\mathsf{t}} \approx [P]_{\mathsf{t}} + [A^i]_{\mathsf{t}}$$

Lemma 1. *For any program P then $[P]_{\mathsf{t}} \geqslant [P]_{\mathsf{v}}$.*

Denotational semantics

Denotational semantics

- State for *virtual time*
- Read only *actual time* (updated from OS)

$$\texttt{Temporal}\ a = (\text{start time, current time}) \rightarrow (\text{old vtime} \rightarrow (a, \text{new vtime}))$$

$$[\![P]\!]_{\text{top}} : \texttt{Temporal ()}$$

- Paper describes core semantics with Haskell

$$\texttt{Temporal}\ a = (\texttt{Time},\ \texttt{Time}) \rightarrow (\texttt{VTime} \rightarrow \texttt{IO}\ (a, \texttt{VTime}))$$

Time safety

soundness of the denotational semantics

- wrt. virtual time

$$\textbf{Lemma 2.}\quad [runTime\,[\![P]\!]]_{\mathsf{v}} \;=\; [P]_{\mathsf{v}}$$

- wrt. actual time (modulo constant sequential overhead)

$$\textbf{Lemma 3.}\quad [runTime\,[\![P]\!]]_{\mathsf{t}} \;\approx\; [P]_{\mathsf{t}}$$

Temporal monad for you?

- Reusable for other purposes
- Code online (http://github.com/dorchard/temporal-monad)
- Generalisations to applicative functor & monoid
- Over-run warnings (hard & soft)