# Temporal semantics for a live coding language

Sam Aaron, Dominic Orchard, Alan Blackwell

@samaaron @dorchard

**Fibonacci Crisis** 



FARM 2014, 6th September 2014, Gothenburg



### http://overtone.github.io



ins meta-ex.shader [:use [overtone.live] (:require [shadertone.tone :as t])) t/start-fullscreen "resources/shaders/fireball.gls[" 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 iChannel@ and a texture of the ;; previous frame in iChannel1 :textures [:overtone-audio :previous-frame] it/start-fullscreen "resources/shaders/menger-san.glsl"
;; this puts the FFT data in iChannel@ and a texture of the :: previous frame in iChannel1 t/start-fullscreen "resources/shaders/zoonwave.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 ["resoury 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

ins meta-ex.grumbles
iuse [overtone.live]
[meta-ex.kit.mixer]
[meta-ex.sets.ignite])
;; Inspired by an example in an early chapter of the SuperCollider book

(defonce grunble-g (group)

def ob (nkmx :s1) def ob 0 volume 0,55

~ n

do

grumble thead grumble g :freq-mul 2 :out-bus ob :and 2 grumble thead grumble-g :freq-mul 1.8 :out-bus ob :amp 2 grumble thead grumble-g :freq-mul 1.5 :out-bus ob :amp 2

do (grumble [:head grumble\_g] :freq-nul 1 :out-bus ob :ano 1
(grumble [:head grumble\_g] :freq-nul 0.5 :out-bus ob :ano 1))

grumble [:head grumble-g] :freq-nul 1 :out-bus ob :amp 3
grumble [:head grumble-g] :freq-nul 0.5 :out-bus ob :amp 2
ctl grumble-g :speed 1997)

arg-map

sin-ctl nkmx-sctl:sl :amp E:freg-mul 0 :mul 1 :add 0.5)

ctl nkmx-sctl:s1 :free-mul-7 0 :mul-7 1 :add-7 0.5

ctl nkmx-sctl:s1 (req-mu)-13-1/8 mul-13-1 :add-13-0.5 ;;(status) 8.11] Tasks: 248 total, 0 running 1.3] Load average: 1.72 1.61 1.59 6.5] Uptime: 17:27:12 1.30]

PID	USER/		I NI	VIRT	RES	SHR	5	CPUN	MEMPA	TIME+	Connand
4425	sam	3	1 0	2487M	- 1668	0	C	0.0	0.0	0:00.00	htop
1	1001		0 0	0	0	. 0	?	0.0	0.0	0:00.00	(launchd)
11	1001		8 8	0	0	0	?	0.0	0.0	0:00.00	(UserEventAgent)
12	AMPA.	1	8 8	0		0	7	0.0	0.0	0:00.00	(kextd)
- 13	root	1	8 8	. 0	0	0		0.0	8.0	0:00.00	(taskgated)
14	root		0 0	0 (	0			0.0	0.0	0:00.00	(notifyd)
15	root		8 8	0				0.0	0.0	0:00.00	(securityd)
16			0 0	0 / 0	0	0		0.0	0.0	0:00.00	(diskarbitrationd
17			8 8	0	0	0		0.0	0.0	0:00.00	(configd)
18			0 0	0	0	0		0.0	0.0	0:00.00	(powerd)
Elliplic	100	etun	Sear	ODE ATES	up at EST	500	10	SortBu	ETAIL	ESHIC	E STATE STATE

# meta-ex

\$51 \$1 \$\$\$ \$\$\$ \$\$\$\$\$\$\\\$\$\$\$\$\$ \$\$\$\$ \$\$\$\$ \\$\$\$\$\$\$\| \| \$\$\$\$\$\$\ >\$\$ - 55 \$5\\$5 \$5 \$5 \$5 \$5 \$5 | \$5 \_\_\_ / \$5 \$5 \\$\$5 | \$5 | \$5\$5\$\$\$\$ | \$5 \_\_\_ / \$5 \$\$ \\$\$\$\$\$5 \$5 55 55 / 5555\ 55555555| 55 \55\ 155 55 155 55 \$\$ \5 \$\$ \55 \| \$\$ | \$\$ 155 155 15555555 15555 15555555 \\$\$\$\$\$\$\$ \\$5 155 \\$\$

-> Connection established

Collaborative Programmable Music. v0.10-dev

Hello Sam. Do you feel it? I do. Creativity is rushing through your veins today!

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 'oluse 'overtone.live)2014-09-04 16:26:18.480 java[4344:d0b] Unknown modifier with keycode: 0

-UU-:@##--F2 shader.clj All (8,54) Gie anter (Clojure cider[meta-ex -UU-:@---F2 grumbles.clj All (1,3) Git:master (Clojure cider[meta-ex



a 2 measured me in the

· /. mararalters that it's part to

· A powe connecto Raspbe

feat now be

Log

#### Starting 1

of the most powerful ways to communicate with Id is full of strange arcane commands - much like The strange lerefore move to a more familiar graphical a computer. How IN the contents of a may c spell. We and many parts that may perhaps feel a bit more comfortable. To

Z

Phat there have the

9. Plug the HDAR CAR cable into your it's switched on and rital option). the small USB n, you should see the

nas 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







Vorkspace 1 Workspace 3 Workspace 4 Workspace 5 Workspace 1 Output 6 X	🛸 X11	Applications	Edit Window	Help				
Vorkspace 1 Workspace 2 Workspace 3 Workspace 4 Workspace 5 Workspace 4 Output	000			X	Sonic Pi - Kraft			
Vorkspace 1 Workspace 2 Workspace 3 Workspace 4 Workspace 5 Workspace 4 Output & *	0							
	Workspace 1	Workspace 2	Workspace 3	Workspace 4	Workspace 5	Workspace 🚽 I	Output	₽×
Errors 6×								
							Errors	5×
						<u>_</u>		<u> </u>





![](_page_12_Picture_0.jpeg)

![](_page_13_Picture_0.jpeg)

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

and a state

0

#### 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.

![](_page_15_Picture_8.jpeg)

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

Cambridge News | cambridge-news.co.uk | November 4, 2013 | 5

#### 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 reinspect 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.

![](_page_16_Picture_0.jpeg)

![](_page_17_Picture_0.jpeg)

#### LEARN TO CODE AND MAKE MUSIC WITH (((Sonic $\pi$ ))) VISIT RASPBERRYPLORG TO FIND OUT MORE!

DICITEDEDUCATOR.COM / ROB-S-WILLIAMS.COM

![](_page_18_Picture_0.jpeg)

![](_page_19_Picture_0.jpeg)

![](_page_20_Picture_0.jpeg)

![](_page_21_Picture_0.jpeg)

#### 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!

SOMPETITION FOR SCHOOLS

Check out **raspberry.org/competitions/sonic-pi** to find out how to enter.

![](_page_22_Picture_4.jpeg)

DI

o find out how to enter.

Check out raspberry.org/competitions/sonic-pi

д П))) В Sonic F

musical artists Juneau Projects, and with live coder-musician Sam Aaron!

![](_page_23_Figure_0.jpeg)

![](_page_24_Figure_0.jpeg)

# 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 v2.0 syntax

 $P ::= P; S \mid \emptyset$  $S ::= E \mid v = E$  $E ::= sleep \mathbb{R}_{\geq 0} \mid A^i \mid v$ 

## Time system

- $$\begin{split} [--]_{\mathsf{v}} : \text{virtual time} & [--]_{\mathsf{t}} : \text{actual time} \\ [\emptyset]_{\mathsf{v}} &= 0 & [\emptyset]_{\mathsf{t}} \approx 0 \\ [P; v = E]_{\mathsf{v}} &= [P]_{\mathsf{v}} + [E]_{\mathsf{v}} & [P; \texttt{sleep } t]_{\mathsf{t}} \approx ([P]_{\mathsf{v}} + t) \max [P]_{\mathsf{t}} \\ [\texttt{sleep } t]_{\mathsf{v}} &= t & [P; v = A^{i}]_{\mathsf{t}} \approx [P]_{\mathsf{t}} + [A^{i}]_{\mathsf{t}} \\ [A^{i}]_{\mathsf{v}} &= 0 \end{split}$$
- e.g. P; sleep 2 where  $[P]_t = 1$ ,  $[P]_v = 0$   $\therefore [P; sleep 2]_t = (0 + 2) \max 1 = 2$  $[P; sleep 2]_v = 2$

## Time system

- $$\begin{split} [--]_{\mathsf{v}} : \text{virtual time} & [--]_{\mathsf{t}} : \text{actual time} \\ [\emptyset]_{\mathsf{v}} &= 0 & [\emptyset]_{\mathsf{t}} \approx 0 \\ [P; v = E]_{\mathsf{v}} &= [P]_{\mathsf{v}} + [E]_{\mathsf{v}} & [P; \texttt{sleep } t]_{\mathsf{t}} \approx ([P]_{\mathsf{v}} + t) \max [P]_{\mathsf{t}} \\ [\texttt{sleep } t]_{\mathsf{v}} &= t & [P; v = A^{i}]_{\mathsf{t}} \approx [P]_{\mathsf{t}} + [A^{i}]_{\mathsf{t}} \\ [A^{i}]_{\mathsf{v}} &= 0 \end{split}$$
- e.g. P; sleep 1 where  $[P]_t = 2$ ,  $[P]_v = 0$   $\therefore [P; sleep 1]_t = (0 + 1) max 2 = 2$  $[P; sleep 1]_v = 1$

## Time system

$$\begin{split} [--]_{\mathsf{v}} &: \text{virtual time} & [--]_{\mathsf{t}} &: \text{actual time} \\ [\emptyset]_{\mathsf{v}} &= 0 & [\emptyset]_{\mathsf{t}} \approx 0 \\ [P; v = E]_{\mathsf{v}} &= [P]_{\mathsf{v}} + [E]_{\mathsf{v}} & [P; \texttt{sleep } t]_{\mathsf{t}} \approx ([P]_{\mathsf{v}} + t) \max [P]_{\mathsf{t}} \\ [\texttt{sleep } t]_{\mathsf{v}} &= t & [P; v = A^{i}]_{\mathsf{t}} \approx [P]_{\mathsf{t}} + [A^{i}]_{\mathsf{t}} \\ [A^{i}]_{\mathsf{v}} &= 0 \end{split}$$

#### **Lemma 1.** For any program P then $[P]_t \ge [P]_v$ .

### **Denotational semantics**

![](_page_30_Picture_1.jpeg)

### **Denotational semantics**

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

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

 $[\![P]\!]_{\rm top}: {\tt Temporal}$  ()

• Paper describes core semantics with Haskell

Temporal a = (Time, Time) → (VTime → IO (a, VTime))

![](_page_32_Picture_0.jpeg)

#### soundness of the denotational semantics

• wrt. virtual time

Lemma 2.  $[runTime [\![P]\!]]_v = [P]_v$ 

• wrt. actual time (modulo constant sequential overhead) Lemma 3.  $[runTime \llbracket P \rrbracket]_t \approx [P]_t$ 

## Temporal monad for you?

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