clive-core

Claude Heiland-Allen

2012–2019

Contents

1 BSD3.md

```
Copyright (c) 2011-2018, Claude Heiland-Allen

All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

  * Redistributions of source code must retain the above copyright
    notice, this list of conditions and the following disclaimer.

  * Redistributions in binary form must reproduce the above
    copyright notice, this list of conditions and the following
    disclaimer in the documentation and/or other materials provided
    with the distribution.

  * Neither the name of Claude Heiland-Allen nor the names of other
    contributors may be used to endorse or promote products derived
    from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

2 CC-BY-SA.md

```
## creative commons

# Attribution-ShareAlike 4.0 International

Creative Commons Corporation ("Creative Commons") is not a law firm and does not↙
    ↳ provide legal services or legal advice. Distribution of Creative Commons ↙
    ↳ public licenses does not create a lawyer-client or other relationship. ↙
```

```
    ↪ Creative Commons makes its licenses and related information available on ↙
    ↪ an "as-is" basis. Creative Commons gives no warranties regarding its ↙
    ↪ licenses, any material licensed under their terms and conditions, or any ↙
    ↪ related information. Creative Commons disclaims all liability for damages ↙
    ↪ resulting from their use to the fullest extent possible.

### Using Creative Commons Public Licenses

Creative Commons public licenses provide a standard set of terms and conditions ↙
    ↪ that creators and other rights holders may use to share original works of ↙
    ↪ authorship and other material subject to copyright and certain other ↙
    ↪ rights specified in the public license below. The following considerations↙
    ↪  are for informational purposes only, are not exhaustive, and do not form ↙
    ↪ part of our licenses.

* __Considerations for licensors:__ Our public licenses are intended for use by ↙
    ↪ those authorized to give the public permission to use material in ways ↙
    ↪ otherwise restricted by copyright and certain other rights. Our licenses ↙
    ↪ are irrevocable. Licensors should read and understand the terms and ↙
    ↪ conditions of the license they choose before applying it. Licensors should↙
    ↪  also secure all rights necessary before applying our licenses so that the↙
    ↪  public can reuse the material as expected. Licensors should clearly mark ↙
    ↪ any material not subject to the license. This includes other CC-licensed ↙
    ↪ material, or material used under an exception or limitation to copyright. ↙
    ↪ [More considerations for licensors](http://wiki.creativecommons.org/↙
    ↪ Considerations_for_licensors_and_licensees#Considerations_for_licensors).

* __Considerations for the public:__ By using one of our public licenses, a ↙
    ↪ licensor grants the public permission to use the licensed material under ↙
    ↪ specified terms and conditions. If the licensor's permission is not ↙
    ↪ necessary for any reason--for example, because of any applicable exception↙
    ↪  or limitation to copyright--then that use is not regulated by the license↙
    ↪ . Our licenses grant only permissions under copyright and certain other ↙
    ↪ rights that a licensor has authority to grant. Use of the licensed ↙
    ↪ material may still be restricted for other reasons, including because ↙
    ↪ others have copyright or other rights in the material. A licensor may make↙
    ↪  special requests, such as asking that all changes be marked or described.↙
    ↪  Although not required by our licenses, you are encouraged to respect ↙
    ↪ those requests where reasonable. [More considerations for the public](http↙
    ↪ ://wiki.creativecommons.org/Considerations_for_licensors_and_licensees#↙
    ↪ Considerations_for_licensees).

## Creative Commons Attribution-ShareAlike 4.0 International Public License

By exercising the Licensed Rights (defined below), You accept and agree to be ↙
    ↪ bound by the terms and conditions of this Creative Commons Attribution-↙
    ↪ ShareAlike 4.0 International Public License ("Public License"). To the ↙
    ↪ extent this Public License may be interpreted as a contract, You are ↙
    ↪ granted the Licensed Rights in consideration of Your acceptance of these ↙
    ↪ terms and conditions, and the Licensor grants You such rights in ↙
    ↪ consideration of benefits the Licensor receives from making the Licensed ↙
    ↪ Material available under these terms and conditions.

### Section 1 -- Definitions.

a. __Adapted Material__ means material subject to Copyright and Similar Rights ↙
    ↪ that is derived from or based upon the Licensed Material and in which the ↙
```

Licensed Material is translated, altered, arranged, transformed, or otherwise modified in a manner requiring permission under the Copyright and Similar Rights held by the Licensor. For purposes of this Public License, where the Licensed Material is a musical work, performance, or sound recording, Adapted Material is always produced where the Licensed Material is synched in timed relation with a moving image.

b. __Adapter's License__ means the license You apply to Your Copyright and Similar Rights in Your contributions to Adapted Material in accordance with the terms and conditions of this Public License.

c. __BY-SA Compatible License__ means a license listed at creativecommons.org/compatiblelicenses, approved by Creative Commons as essentially the equivalent of this Public License.

d. __Copyright and Similar Rights__ means copyright and/or similar rights closely related to copyright including, without limitation, performance, broadcast, sound recording, and Sui Generis Database Rights, without regard to how the rights are labeled or categorized. For purposes of this Public License, the rights specified in Section 2(b)(1)-(2) are not Copyright and Similar Rights.

e. __Effective Technological Measures__ means those measures that, in the absence of proper authority, may not be circumvented under laws fulfilling obligations under Article 11 of the WIPO Copyright Treaty adopted on December 20, 1996, and/or similar international agreements.

f. __Exceptions and Limitations__ means fair use, fair dealing, and/or any other exception or limitation to Copyright and Similar Rights that applies to Your use of the Licensed Material.

g. __License Elements__ means the license attributes listed in the name of a Creative Commons Public License. The License Elements of this Public License are Attribution and ShareAlike.

h. __Licensed Material__ means the artistic or literary work, database, or other material to which the Licensor applied this Public License.

i. __Licensed Rights__ means the rights granted to You subject to the terms and conditions of this Public License, which are limited to all Copyright and Similar Rights that apply to Your use of the Licensed Material and that the Licensor has authority to license.

j. __Licensor__ means the individual(s) or entity(ies) granting rights under this Public License.

k. __Share__ means to provide material to the public by any means or process that requires permission under the Licensed Rights, such as reproduction, public display, public performance, distribution, dissemination, communication, or importation, and to make material available to the public including in ways that members of the public may access the material from a place and at a time individually chosen by them.

l. __Sui Generis Database Rights__ means rights other than copyright resulting from Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, as amended and/or

succeeded, as well as other essentially equivalent rights anywhere in the world.

m. __You__ means the individual or entity exercising the Licensed Rights under this Public License. Your has a corresponding meaning.

Section 2 -- Scope.

a. ___License grant.___

1. Subject to the terms and conditions of this Public License, the Licensor hereby grants You a worldwide, royalty-free, non-sublicensable, non-exclusive, irrevocable license to exercise the Licensed Rights in the Licensed Material to:

   A. reproduce and Share the Licensed Material, in whole or in part; and

   B. produce, reproduce, and Share Adapted Material.

2. __Exceptions and Limitations.__ For the avoidance of doubt, where Exceptions and Limitations apply to Your use, this Public License does not apply, and You do not need to comply with its terms and conditions.

3. __Term.__ The term of this Public License is specified in Section 6(a).

4. __Media and formats; technical modifications allowed.__ The Licensor authorizes You to exercise the Licensed Rights in all media and formats whether now known or hereafter created, and to make technical modifications necessary to do so. The Licensor waives and/or agrees not to assert any right or authority to forbid You from making technical modifications necessary to exercise the Licensed Rights, including technical modifications necessary to circumvent Effective Technological Measures. For purposes of this Public License, simply making modifications authorized by this Section 2(a)(4) never produces Adapted Material.

5. __Downstream recipients.__

   A. __Offer from the Licensor -- Licensed Material.__ Every recipient of the Licensed Material automatically receives an offer from the Licensor to exercise the Licensed Rights under the terms and conditions of this Public License.

   B. __Additional offer from the Licensor -- Adapted Material.__ Every recipient of Adapted Material from You automatically receives an offer from the Licensor to exercise the Licensed Rights in the Adapted Material under the conditions of the Adapter's License You apply.

   C. __No downstream restrictions.__ You may not offer or impose any additional or different terms or conditions on, or apply any Effective Technological Measures to, the Licensed Material if doing so restricts exercise of the Licensed Rights by any recipient of the Licensed Material.

6. __No endorsement.__ Nothing in this Public License constitutes or may be construed as permission to assert or imply that You are, or that Your

use of the Licensed Material is, connected with, or sponsored, endorsed, or granted official status by, the Licensor or others designated to receive attribution as provided in Section 3(a)(1)(A)(i).

b. ___Other rights.___

1. Moral rights, such as the right of integrity, are not licensed under this Public License, nor are publicity, privacy, and/or other similar personality rights; however, to the extent possible, the Licensor waives and/or agrees not to assert any such rights held by the Licensor to the limited extent necessary to allow You to exercise the Licensed Rights, but not otherwise.

2. Patent and trademark rights are not licensed under this Public License.

3. To the extent possible, the Licensor waives any right to collect royalties from You for the exercise of the Licensed Rights, whether directly or through a collecting society under any voluntary or waivable statutory or compulsory licensing scheme. In all other cases the Licensor expressly reserves any right to collect such royalties.

Section 3 -- License Conditions.

Your exercise of the Licensed Rights is expressly made subject to the following conditions.

a. ___Attribution.___

1. If You Share the Licensed Material (including in modified form), You must:

   A. retain the following if it is supplied by the Licensor with the Licensed Material:

      i. identification of the creator(s) of the Licensed Material and any others designated to receive attribution, in any reasonable manner requested by the Licensor (including by pseudonym if designated);

      ii. a copyright notice;

      iii. a notice that refers to this Public License;

      iv. a notice that refers to the disclaimer of warranties;

      v. a URI or hyperlink to the Licensed Material to the extent reasonably practicable;

   B. indicate if You modified the Licensed Material and retain an indication of any previous modifications; and

   C. indicate the Licensed Material is licensed under this Public License, and include the text of, or the URI or hyperlink to, this Public License.

2. You may satisfy the conditions in Section 3(a)(1) in any reasonable manner based on the medium, means, and context in which You Share the Licensed Material. For example, it may be reasonable to satisfy the

conditions by providing a URI or hyperlink to a resource that includes the required information.

3. If requested by the Licensor, You must remove any of the information required by Section 3(a)(1)(A) to the extent reasonably practicable.

b. ___ShareAlike.___

In addition to the conditions in Section 3(a), if You Share Adapted Material You produce, the following conditions also apply.

1. The Adapter's License You apply must be a Creative Commons license with the same License Elements, this version or later, or a BY-SA Compatible License.

2. You must include the text of, or the URI or hyperlink to, the Adapter's License You apply. You may satisfy this condition in any reasonable manner based on the medium, means, and context in which You Share Adapted Material.

3. You may not offer or impose any additional or different terms or conditions on, or apply any Effective Technological Measures to, Adapted Material that restrict exercise of the rights granted under the Adapter's License You apply.

Section 4 -- Sui Generis Database Rights.

Where the Licensed Rights include Sui Generis Database Rights that apply to Your use of the Licensed Material:

a. for the avoidance of doubt, Section 2(a)(1) grants You the right to extract, reuse, reproduce, and Share all or a substantial portion of the contents of the database;

b. if You include all or a substantial portion of the database contents in a database in which You have Sui Generis Database Rights, then the database in which You have Sui Generis Database Rights (but not its individual contents) is Adapted Material, including for purposes of Section 3(b); and

c. You must comply with the conditions in Section 3(a) if You Share all or a substantial portion of the contents of the database.

For the avoidance of doubt, this Section 4 supplements and does not replace Your obligations under this Public License where the Licensed Rights include other Copyright and Similar Rights.

Section 5 -- Disclaimer of Warranties and Limitation of Liability.

a. __Unless otherwise separately undertaken by the Licensor, to the extent possible, the Licensor offers the Licensed Material as-is and as-available, and makes no representations or warranties of any kind concerning the Licensed Material, whether express, implied, statutory, or other. This includes, without limitation, warranties of title, merchantability, fitness for a particular purpose, non-infringement, absence of latent or other defects, accuracy, or the presence or absence of errors, whether or not known or discoverable. Where disclaimers of warranties are not allowed in full or in part, this disclaimer may not apply to You.__

b. __To the extent possible, in no event will the Licensor be liable to You on any legal theory (including, without limitation, negligence) or otherwise for any direct, special, indirect, incidental, consequential, punitive, exemplary, or other losses, costs, expenses, or damages arising out of this Public License or use of the Licensed Material, even if the Licensor has been advised of the possibility of such losses, costs, expenses, or damages. Where a limitation of liability is not allowed in full or in part, this limitation may not apply to You.__

c. The disclaimer of warranties and limitation of liability provided above shall be interpreted in a manner that, to the extent possible, most closely approximates an absolute disclaimer and waiver of all liability.

Section 6 -- Term and Termination.

a. This Public License applies for the term of the Copyright and Similar Rights licensed here. However, if You fail to comply with this Public License, then Your rights under this Public License terminate automatically.

b. Where Your right to use the Licensed Material has terminated under Section 6(a), it reinstates:

   1. automatically as of the date the violation is cured, provided it is cured within 30 days of Your discovery of the violation; or

   2. upon express reinstatement by the Licensor.

   For the avoidance of doubt, this Section 6(b) does not affect any right the Licensor may have to seek remedies for Your violations of this Public License.

c. For the avoidance of doubt, the Licensor may also offer the Licensed Material under separate terms or conditions or stop distributing the Licensed Material at any time; however, doing so will not terminate this Public License.

d. Sections 1, 5, 6, 7, and 8 survive termination of this Public License.

Section 7 -- Other Terms and Conditions.

a. The Licensor shall not be bound by any additional or different terms or conditions communicated by You unless expressly agreed.

b. Any arrangements, understandings, or agreements regarding the Licensed Material not stated herein are separate from and independent of the terms and conditions of this Public License.t stated herein are separate from and independent of the terms and conditions of this Public License.

Section 8 -- Interpretation.

a. For the avoidance of doubt, this Public License does not, and shall not be interpreted to, reduce, limit, restrict, or impose conditions on any use of the Licensed Material that could lawfully be made without permission under this Public License.

b. To the extent possible, if any provision of this Public License is deemed

```
    ↳ unenforceable, it shall be automatically reformed to the minimum extent ↙
    ↳ necessary to make it enforceable. If the provision cannot be reformed, it ↙
    ↳ shall be severed from this Public License without affecting the ↙
    ↳ enforceability of the remaining terms and conditions.

c. No term or condition of this Public License will be waived and no failure to ↙
    ↳ comply consented to unless expressly agreed to by the Licensor.

d. Nothing in this Public License constitutes or may be interpreted as a ↙
    ↳ limitation upon, or waiver of, any privileges and immunities that apply to↙
    ↳  the Licensor or You, including from the legal processes of any ↙
    ↳ jurisdiction or authority.
170
> Creative Commons is not a party to its public licenses. Notwithstanding, ↙
    ↳ Creative Commons may elect to apply one of its public licenses to material↙
    ↳  it publishes and in those instances will be considered the "Licensor." ↙
    ↳ Except for the limited purpose of indicating that material is shared under↙
    ↳  a Creative Commons public license or as otherwise permitted by the ↙
    ↳ Creative Commons policies published at [creativecommons.org/policies](http↙
    ↳ ://creativecommons.org/policies), Creative Commons does not authorize the ↙
    ↳ use of the trademark "Creative Commons" or any other trademark or logo of ↙
    ↳ Creative Commons without its prior written consent including, without ↙
    ↳ limitation, in connection with any unauthorized modifications to any of ↙
    ↳ its public licenses or any other arrangements, understandings, or ↙
    ↳ agreements concerning use of licensed material. For the avoidance of doubt↙
    ↳ , this paragraph does not form part of the public licenses.
>
> Creative Commons may be contacted at creativecommons.org
```

3 client/bela.mk

```
GCC ?= arm-linux-gnueabihf-gcc-6

../build/go.so: go.c dsp.h dsp/*.h
	$(GCC) -std=c99 -Wall -pedantic -Wextra -Wno-unused-parameter -O3 -march↙
	    ↳ =armv7-a -mtune=cortex-a8 -mfloat-abi=hard -mfpu=neon -ftree-↙
	    ↳ vectorize -I. -shared -fPIC -o ../build/go.so go.c -lm
```

4 client/clive-client.c

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

 5  #include <sys/types.h>
#include <sys/wait.h>
#include <unistd.h>

#include <linux/limits.h>
10  #include <sys/inotify.h>

int main(int argc, char **argv) {
  // build remake command line
  // FIXME this doesn't quote or escape or otherwise protect argv[1]
15  // FIXME don't pass untrusted input to this command!
  const char *templat =
```

```
    "git add go.c go.h ; "
    "git --no-pager diff --cached --color ; "
    "git commit -uno -m \"go $(date --iso=s)\" ; "
    "make --quiet -f "
    ;
const char *makefile = "native-sse.mk";
if (argc > 1)
{
  makefile = argv[1];
}
size_t bytes = strlen(templat) + strlen(makefile) + 1;
char *command = malloc(bytes);
if (! command) {
  perror("malloc()");
  return 1;
}
command[0] = 0;
strncat(command, templat, bytes - 1 - strlen(command));
strncat(command, makefile, bytes - 1 - strlen(command));
// watch for filesystem changes
int ino = inotify_init();
if (ino == -1) {
  perror("inotify_init()");
  return 1;
}
int wd = inotify_add_watch(ino, ".", IN_CLOSE_WRITE);
if (wd == -1) {
  perror("inotify_add_watch()");
  return 1;
}
ssize_t buf_bytes = sizeof(struct inotify_event) + NAME_MAX + 1;
char *buf = malloc(buf_bytes);
if (! buf)
{
  perror("malloc()");
  return 1;
}
// recompile
int ret = system(command);
if (ret == -1 || ! WIFEXITED(ret) || WEXITSTATUS(ret) != 0) {
  fprintf(stderr, "\x1b[31;1m%s: %d\x1b[0m\n", "SYSTEM ERROR", ret);
}
// main loop
while (1) {
  // read events (blocking)
  memset(buf, 0, buf_bytes);
  ssize_t r = read(ino, buf, buf_bytes);
  if (r == -1) {
    perror("read()");
    sleep(1);
  } else {
    char *bufp = buf;
    while (bufp < buf + r)
    {
      struct inotify_event *ev = (struct inotify_event *) bufp;
      bufp += sizeof(struct inotify_event) + ev->len;
      if (ev->mask & IN_CLOSE_WRITE) {
```

```
            fprintf(stderr, "\x1b[32;1mFILE CHANGED: '%s'\x1b[0m\n", ev->name);
75          if (0 == strcmp("go.c", ev->name) || 0 == strcmp("go.h", ev->name)) {
              // recompile
              int ret = system(command);
              if (ret == -1 || ! WIFEXITED(ret) || WEXITSTATUS(ret) != 0) {
                fprintf(stderr, "\x1b[31;1m%s: %d\x1b[0m\n", "SYSTEM ERROR", ret);
80            }
            }
          }
        }
      }
85  }
    // never reached
    close(ino);
    free(command);
    return 0;
90  }
```

5 client/dsp/delay.h

```
#ifndef DSP_DELAY_H
#define DSP_DELAY_H 1

#include "func.h"
5
typedef struct { int length, woffset; } DELAY;

static inline void delwrite(DELAY *del, sample x0) {
  float *buffer = (float *) (del + 1);
10  int l = del->length;
  l = (l > 0) ? l : 1;
  int w = del->woffset;
  buffer[w++] = x0;
  if (w >= l) { w -= l; }
15  del->woffset = w;
}

static inline sample delread1(DELAY *del, sample ms) {
  float *buffer = (float *) (del + 1);
20  int l = del->length;
  l = (l > 0) ? l : 1;
  int w = del->woffset;
  int d = ms / (sample)1000 * SR;
  d = (0 < d && d < l) ? d : 0;
25  int r = w - d;
  r = r < 0 ? r + l : r;
  return buffer[r];
}

30 static inline sample delread2(DELAY *del, sample ms) {
  float *buffer = (float *) (del + 1);
  int l = del->length;
  l = (l > 0) ? l : 1;
  int w = del->woffset;
35  sample d = ms / (sample)1000 * SR;
  int d0 = floor(d);
  int d1 = d0 + 1;
```

```
  sample t = d - d0;
  d0 = (0 < d0 && d0 < l) ? d0 : 0;
  d1 = (0 < d1 && d1 < l) ? d1 : d0;
  int r0 = w - d0;
  int r1 = w - d1;
  r0 = r0 < 0 ? r0 + l : r0;
  r1 = r1 < 0 ? r1 + l : r1;
  sample y0 = buffer[r0];
  sample y1 = buffer[r1];
  return (1 - t) * y0 + t * y1;
}

// https://en.wikipedia.org/wiki/Cubic_Hermite_spline#Interpolation_on_the_unit_interval_without_exact_derivatives

static inline sample delread4(DELAY *del, sample ms) {
  float *buffer = (float *) (del + 1);
  int l = del->length;
  l = (l > 0) ? l : 1;
  int w = del->woffset;
  sample d = ms / (sample) 1000 * SR;
  int d1 = floor(d);
  int d0 = d1 - 1;
  int d2 = d1 + 1;
  int d3 = d1 + 2;
  sample t = d - d1;
  d0 = (0 < d0 && d0 < l) ? d0 : 0;
  d1 = (0 < d1 && d1 < l) ? d1 : d0;
  d2 = (0 < d2 && d2 < l) ? d2 : d1;
  d3 = (0 < d3 && d3 < l) ? d3 : d2;
  int r0 = w - d0;
  int r1 = w - d1;
  int r2 = w - d2;
  int r3 = w - d3;
  r0 = r0 < 0 ? r0 + l : r0;
  r1 = r1 < 0 ? r1 + l : r1;
  r2 = r2 < 0 ? r2 + l : r2;
  r3 = r3 < 0 ? r3 + l : r3;
  sample y0 = buffer[r0];
  sample y1 = buffer[r1];
  sample y2 = buffer[r2];
  sample y3 = buffer[r3];
  sample a0 = -t*t*t + 2*t*t - t;
  sample a1 = 3*t*t*t - 5*t*t + 2;
  sample a2 = -3*t*t*t + 4*t*t + t;
  sample a3 = t*t*t - t*t;
  return (a0 * y0 + a1 * y1 + a2 * y2 + a3 * y3) / 2;
}

#endif
```

6 client/dsp/dtmf.h

```
static struct { int lo, hi; } dtmf[10] = {
  { 941, 1336 },
  { 697, 1209 },
  { 697, 1336 },
```

```
    { 697, 1477 },
    { 770, 1209 },
    { 770, 1336 },
    { 770, 1477 },
    { 852, 1209 },
    { 852, 1336 },
    { 852, 1477 }
};
```

7 client/dsp/effects.h

```
#ifndef DSP_EFFECTS_H
#define DSP_EFFECTS_H 1

#include "func.h"
#include "delay.h"
#include "osc.h"
#include "filter.h"

typedef struct { DELAY tape; float buf[16384]; } CHORUS;

static inline sample chorus(CHORUS *ch, int voices, sample delayms, sample depthms, sample phase, sample x) {
  ch->tape.length = 16384;
  sample c = 0;
  for (int i = 0; i < voices; ++i) {
    c += delread4(&ch->tape, delayms + depthms * cos(twopi * (phase + i / (sample) voices)));
  }
  c /= voices;
  delwrite(&ch->tape, x - c);
  return c;
}

typedef struct { PHASOR head; DELAY tape; float buf[16384]; } PITCHSHIFT;

static inline sample pitchshift(PITCHSHIFT *ps, sample delayms, sample windowms, sample transposesemi, sample x) {
  ps->tape.length = 16384;
  delwrite(&ps->tape, x);
  sample p0 = phasor(&ps->head, (1 - exp((sample)0.05776 * transposesemi)) * 1000 / windowms);
  sample p1 = wrap(p0 + (sample)0.5);
  return sin(pi * p0) * delread4(&ps->tape, windowms * p0 + delayms)
       + sin(pi * p1) * delread4(&ps->tape, windowms * p1 + delayms);
}

typedef struct { HIP hip[2]; LOP lop[3]; } COMPRESS;

void compress(sample out[2], COMPRESS *s, sample hiphz, sample lophz1, sample lophz2, sample db, const sample in[2]) {
  sample h[2] =
    { hip(&s->hip[0], in[0], hiphz)
    , hip(&s->hip[1], in[1], hiphz)
    };
  h[0] *= h[0];
  h[1] *= h[1];
```

```
  h[0] = lop(&s->lop[0], h[0], lophz1);
  h[1] = lop(&s->lop[1], h[1], lophz1);
  sample env = lop(&s->lop[2], sqrt(fmax(0, h[0] + h[1])), lophz2);
  sample env0 = env;
  env = rmstodb(env);
  if (env > db) {
    env = db + (env - db) / 4;
  } else {
    env = db;
  }
  env = (sample)0.25 * dbtorms(env) / dbtorms((100 - db) / 4 + db);
  sample gain = env / env0;
  if (isnan(gain) || isinf(gain)) { gain = 0; }
  gain = clamp(gain, 0, (sample)1.0e6);
  out[0] = tanh(in[0] * gain);
  out[1] = tanh(in[1] * gain);
}

// reverb

typedef struct {
  sample x[4];
} vec4;

static inline void vsub(vec4 *o, const vec4 *a, const vec4 *b) {
  for (int i = 0; i < 4; ++i) { o->x[i] = a->x[i] - b->x[i]; }
}

static inline void vmul(vec4 *o, const vec4 *a, sample b) {
  for (int i = 0; i < 4; ++i) { o->x[i] = a->x[i] * b; }
}

static inline sample vdot(const vec4 *a, const vec4 *b) {
  sample s = 0;
  for (int i = 0; i < 4; ++i) { s += a->x[i] * b->x[i]; }
  return s;
}

static inline void vref(vec4 *o, const vec4 *x, const vec4 *normal, sample distance) {
  vmul(o, normal, 2 * (vdot(x, normal) - distance) / vdot(normal, normal));
  vsub(o, x, o);
}

static inline sample vtime(const vec4 *x, const vec4 *y) {
  sample c = (sample)340.0 / (sample)1000.0;
  vec4 v;
  vsub(&v, x, y);
  sample d = vdot(&v, &v);
  return sqrt(d) / c;
}

typedef struct {
  sample times[9];
  sample decays[9];
  DELAY del;
  float delbuf[SR];
```

```
} EARLYREF;

static inline void early_ref_init(EARLYREF *eref, vec4 *size, vec4 *source, vec4 *listener) {
  vec4 normals[8] =
    { {{  1, 0, 0, 0 }}, {{ -1, 0, 0, 0 }}
    , {{ 0,  1, 0, 0 }}, {{ 0, -1, 0, 0 }}
    , {{ 0, 0,  1, 0 }}, {{ 0, 0, -1, 0 }}
    , {{ 0, 0, 0,  1 }}, {{ 0, 0, 0, -1 }}
    };
  sample distance[8] =
    { 0, size->x[0], 0, size->x[1], 0, size->x[2], 0, size->x[3] };
  vec4 sources[9];
  vmul(&sources[0], source, 1);
  for (int i = 1; i < 9; ++i) {
    vref(&sources[i], source, &normals[i - 1], distance[i - 1]);
  }
  for (int i = 0; i < 9; ++i) {
    eref->times[i] = vtime(&sources[i], listener);
    eref->decays[i] = pow(10, -340 * eref->times[i] / 10000);
  }
  eref->del.length = SR;
}

static inline sample early_ref(EARLYREF *eref, sample audio, sample brightness) {
  delwrite(&eref->del, audio);
  sample s = 0;
  for (int i = 1; i < 9; ++i) {
    s += eref->decays[i] * delread1(&eref->del, eref->times[i]);
  }
  s *= brightness;
  s += eref->decays[0] * delread1(&eref->del, eref->times[0]);
  return s;
}

static inline int cmp_sample(const void *a, const void *b) {
  const sample *x = (const sample *) a;
  const sample *y = (const sample *) b;
  if (*x > *y) { return 1; }
  if (*x < *y) { return -1; }
  return 0;
}

typedef struct {
  DELAY del;
  float delbuf[SR];
} REVERB_LINE;

typedef struct {
  sample times[80];
  REVERB_LINE lines[16];
} REVERB;

static inline void reverb_init(REVERB *r, const vec4 *size) {
  sample lx = size->x[0];
  sample ly = size->x[1];
```

```
  sample lz = size->x[2];
  sample lw = size->x[3];
  int k = 0;
  for (int nx = 0; nx < 3; ++nx) {
  for (int ny = 0; ny < 3; ++ny) {
  for (int nz = 0; nz < 3; ++nz) {
  for (int nw = 0; nw < 3; ++nw) {
    if (nx + ny + nz + nw == 0) continue;
    r->times[k++] = (sample)1000.0 / ((sample)(340.0/2.0) * sqrt((nx*nx)/(lx*lx)
        +(ny*ny)/(ly*ly)+(nz*nz)/(lz*lz)+(nw*nw)/(lw*lw)));
  }}}}
  qsort(r->times, 80, sizeof(sample), cmp_sample);
  // de-duplicate
  int wr = 0;
  int rd = 1;
  while (rd < 80) {
    if (r->times[wr] == r->times[rd]) {
      rd++;
    } else {
      r->times[wr] = r->times[rd];
      wr++;
      rd++;
    }
  }
  for (int i = 0; i < 16; ++i) {
    r->lines[i].del.length = SR;
  }
}

static inline void reverb(sample *out, REVERB *r, const sample *in, sample t60) {
  const float mat[16][16] =
  { { 1, -1, -1, -1, -1, 1, 1, 1, -1, 1, 1, 1, -1, 1, 1, 1 }
  , { -1, 1, -1, -1, 1, -1, 1, 1, 1, -1, 1, 1, 1, -1, 1, 1 }
  , { -1, -1, 1, -1, 1, 1, -1, 1, 1, 1, -1, 1, 1, 1, -1, 1 }
  , { -1, -1, -1, 1, 1, 1, 1, -1, 1, 1, 1, -1, 1, 1, 1, -1 }
  , { -1, 1, 1, 1, 1, -1, -1, -1, -1, 1, 1, 1, -1, 1, 1, 1 }
  , { 1, -1, 1, 1, -1, 1, -1, -1, 1, -1, 1, 1, 1, -1, 1, 1 }
  , { 1, 1, -1, 1, -1, -1, 1, -1, 1, 1, -1, 1, 1, 1, -1, 1 }
  , { 1, 1, 1, -1, -1, -1, -1, 1, 1, 1, 1, -1, 1, 1, 1, -1 }
  , { -1, 1, 1, 1, -1, 1, 1, 1, 1, -1, -1, -1, -1, 1, 1, 1 }
  , { 1, -1, 1, 1, 1, -1, 1, 1, -1, 1, -1, -1, 1, -1, 1, 1 }
  , { 1, 1, -1, 1, 1, 1, -1, 1, -1, -1, 1, -1, 1, 1, -1, 1 }
  , { 1, 1, 1, -1, 1, 1, 1, -1, -1, -1, -1, 1, 1, 1, 1, -1 }
  , { -1, 1, 1, 1, -1, 1, 1, 1, -1, 1, 1, 1, -1, -1, -1, -1 }
  , { 1, -1, 1, 1, 1, -1, 1, 1, 1, -1, 1, 1, -1, 1, -1, -1 }
  , { 1, 1, -1, 1, 1, 1, -1, 1, 1, 1, -1, 1, -1, -1, 1, -1 }
  , { 1, 1, 1, -1, 1, 1, 1, -1, 1, 1, 1, -1, -1, -1, -1, 1 }
  };
  float source[16];
  float output[16];
  for (int i = 0; i < 16; ++i) {
    source[i] = (sample)0.125 * in[i & 1] + delread1(&r->lines[i].del, r->times[
        i]) * pow(10, -3 * r->times[i] / fmax(t60, (sample)1e-6));
  }
  for (int i = 0; i < 16; ++i) {
    float s = 0;
```

```
      for (int j = 0; j < 16; ++j) {
        s += mat[i][j] * source[j];
      }
      output[i] = (sample)0.25 * s;
    }
    float s = 0;
    for (int i = 0; i < 8; ++i) {
      s += output[i];
    }
    s *= (sample)0.125;
    out[0] = s;
    s = 0;
    for (int i = 8; i < 16; ++i) {
      s += output[i];
    }
    s *= (sample)0.125;
    out[1] = s;
    for (int i = 0; i < 16; ++i) {
      delwrite(&r->lines[i].del, output[i]);
    }
  }

  #endif
```

8 client/dsp/filter.h

```
  #ifndef DSP_FILTER_H
  #define DSP_FILTER_H 1

  #include "func.h"

  // sample and hold

  typedef struct {
    sample trigger;
    sample value;
  } SAMPHOLD;

  sample samphold(SAMPHOLD *s, sample value, sample trigger) {
    if (trigger < s->trigger) {
          s->value = value;
    }
    s->trigger = trigger;
    return s->value;
  }

  // http://musicdsp.org/files/Audio-EQ-Cookbook.txt

  #define flatq 0.7071067811865476

  typedef struct { double b0, b1, b2, a1, a2, y1, y2; sample x1, x2; } BIQUAD;

  static inline sample biquad(BIQUAD *bq, sample x0) {
    double b0 = bq->b0, b1 = bq->b1, b2 = bq->b2, a1 = bq->a1, a2 = bq->a2;
    double x1 = bq->x1, x2 = bq->x2, y1 = bq->y1, y2 = bq->y2;
    double y0 = b0 * x0 + b1 * x1 + b2 * x2 - a1 * y1 - a2 * y2;
    bq->y2 = y1;
```

```
  bq->y1 = y0;
  bq->x2 = x1;
  bq->x1 = x0;
  return y0;
}

static inline BIQUAD *lowpass(BIQUAD *bq, sample hz, sample q) {
  double w0 = hz * twopi / SR;
  double a = fabs(sin(w0) / (2 * q));
  double c = cos(w0);
  double b0 = (1 - c) / 2, b1 = 1 - c, b2 = (1 - c) / 2;
  double a0 = 1 + a, a1 = -2 * c, a2 = 1 - a;
  bq->b0 = b0 / a0;
  bq->b1 = b1 / a0;
  bq->b2 = b2 / a0;
  bq->a1 = a1 / a0;
  bq->a2 = a2 / a0;
  return bq;
}

static inline BIQUAD *highpass(BIQUAD *bq, sample hz, sample q) {
  double w0 = hz * twopi / SR;
  double a = fabs(sin(w0) / (2 * q));
  double c = cos(w0);
  double b0 = (1 + c) / 2, b1 = -(1 + c), b2 = (1 + c) / 2;
  double a0 = 1 + a, a1 = -2 * c, a2 = 1 - a;
  bq->b0 = b0 / a0;
  bq->b1 = b1 / a0;
  bq->b2 = b2 / a0;
  bq->a1 = a1 / a0;
  bq->a2 = a2 / a0;
  return bq;
}

static inline BIQUAD *bandpass(BIQUAD *bq, sample hz, sample q) {
  double w0 = hz * twopi / SR;
  double a = fabs(sin(w0) / (2 * q));
  double c = cos(w0);
  double b0 = a, b1 = 0, b2 = -a;
  double a0 = 1 + a, a1 = -2 * c, a2 = 1 - a;
  bq->b0 = b0 / a0;
  bq->b1 = b1 / a0;
  bq->b2 = b2 / a0;
  bq->a1 = a1 / a0;
  bq->a2 = a2 / a0;
  return bq;
}

static inline BIQUAD *notch(BIQUAD *bq, sample hz, sample q) {
  double w0 = hz * twopi / SR;
  double a = fabs(sin(w0) / (2 * q));
  double c = cos(w0);
  double b0 = 1, b1 = -2 * c, b2 = 1;
  double a0 = 1 + a, a1 = -2 * c, a2 = 1 - a;
  bq->b0 = b0 / a0;
  bq->b1 = b1 / a0;
  bq->b2 = b2 / a0;
```

```
  bq->a1 = a1 / a0;
  bq->a2 = a2 / a0;
  return bq;
}

// based on pd's [vcf~] [lop~] [hip~]

typedef struct { double re, im; } VCF;

static inline sample vcf(VCF *s, sample x, sample hz, sample q) {
  double qinv = q > 0 ? 1 / q : 0;
  double ampcorrect = 2 - 2 / (q + 2);
  double cf = hz * twopi / SR;
  if (cf < 0) { cf = 0; }
  double r = qinv > 0 ? 1 - cf * qinv : 0;
  if (r < 0) { r = 0; }
  double oneminusr = 1 - r;
  double cre = r * cos(cf);
  double cim = r * sin(cf);
  double re2 = s->re;
  s->re = ampcorrect * oneminusr * x + cre * re2 - cim * s->im;
  s->im = cim * re2 + cre * s->im;
  return s->re;
}

typedef struct { double y; } LOP;

static inline sample lop(LOP *s, sample x, sample hz) {
  double c = clamp(twopi * hz / SR, 0, 1);
  return s->y = mix(x, s->y, 1 - c);
}

typedef struct { double y; } HIP;

static inline sample hip(HIP *s, sample x, sample hz) {
  double c = clamp(1 - twopi * hz / SR, 0, 1);
  double n = (1 + c) / 2;
  double y = x + c * s->y;
  double o = n * (y - s->y);
  s->y = y;
  return o;
}

// pd/extra/hilbert~.pd

typedef struct { double w[2][2][2]; } HILBERT;

static inline void hilbert(sample out[2], HILBERT *h, const sample in[2]) {
  static const double c[2][2][5] =
    { { {  1.94632, -0.94657,   0.94657,  -1.94632, 1 }
      , {  0.83774, -0.06338,   0.06338,  -0.83774, 1 }
      }
    , { { -0.02569,  0.260502, -0.260502,  0.02569, 1 }
      , {  1.8685,  -0.870686,  0.870686, -1.8685,  1 }
      }
    };
  for (int i = 0; i < 2; ++i) {
```

```
    double w = in[i] + c[i][0][0] * h->w[i][0][0] + c[i][0][1] * h->w[i][0][1];
    double x = c[i][0][2] * w + c[i][0][3] * h->w[i][0][0] + c[i][0][4] * h->w[i↙
        ↳ ][0][1];
    h->w[i][0][1] = h->w[i][0][0];
    h->w[i][0][0] = w;
    w = x + c[i][1][0] * h->w[i][1][0] + c[i][1][1] * h->w[i][1][1];
    out[i] = c[i][1][2] * w + c[i][1][3] * h->w[i][1][0] + c[i][1][4] * h->w[i↙
        ↳ ][1][1];
    h->w[i][1][1] = h->w[i][1][0];
    h->w[i][1][0] = w;
  }
}

typedef struct {
  double ya1;
  double wa1;
  double yb1;
  double wb1;
  double yc1;
  double wc1;
  double yd1;
} MOOG;

static inline sample moog(MOOG *m, sample audio, sample frequency, sample ↙
    ↳ resonance) {
  MOOG *muug = m;
  double ya1 = muug->ya1;
  double wa1 = muug->wa1;
  double yb1 = muug->yb1;
  double wb1 = muug->wb1;
  double yc1 = muug->yc1;
  double wc1 = muug->wc1;
  double yd1 = muug->yd1;
  double v = 2;
  double x = audio;
  double f = frequency;
  double r = resonance;
  double g = 1 - exp(-twopi * f / SR);
  double ya = ya1 + v * g * tanh((x - 4 * r * yd1) / v - wa1);
  double wa = tanh(ya / v);
  double yb = yb1 + v * g * (wa - wb1);
  double wb = tanh(yb / v);
  double yc = yc1 + v * g * (wb - wc1);
  double wc = tanh(yc / v);
  double yd = yd1 + v * g * (wc - tanh(yd1 / v));
  double y = yd;
  muug->ya1 = ya;
  muug->wa1 = wa;
  muug->yb1 = yb;
  muug->wb1 = wb;
  muug->yc1 = yc;
  muug->wc1 = wc;
  muug->yd1 = yd;
  return y;
}

// one-pole one-zero low pass filter designed in the Z plane
```

```
200
typedef struct {
  sample x;
  double y;
} LOP1;
205
static inline sample lop1(LOP1 *s, sample x, sample hz)
{
  const double w = twopi * clamp(fabs(hz / SR), 0, (sample)0.5);
  const double a = (1 - sin(w)) / cos(w);
210   const double b = (1 - a) / 2;
  const double y = b * (x + s->x) + a * s->y;
  s->x = x;
  s->y = y;
  return y;
215 }

// one-pole one-zero high pass filter designed in the Z plane

typedef struct {
220   sample x;
  double y;
} HIP1;

static inline sample hip1(HIP1 *s, sample x, sample hz)
225 {
  const double w = twopi * clamp(fabs(hz / SR), 0, (sample)0.5);
  const double a = (1 - sin(w)) / cos(w);
  const double b = (1 + a) / 2;
  const double y = b * (x - s->x) + a * s->y;
230   s->x = x;
  s->y = y;
  return y;
}

235 #endif
```

9 client/dsp/func.h

```
#ifndef DSP_FUNC_H
#define DSP_FUNC_H 1

#include <tgmath.h>
 5 #include <stdbool.h>
#include <stdlib.h>

#define   likely(x) __builtin_expect((x),1)
#define unlikely(x) __builtin_expect((x),0)
10
#define pi 3.141592653589793
#define twopi 6.283185307179586

static inline sample wrap(sample x) {
15   return x - floor(x);
}

static inline sample wrapto(sample x, sample n) {
```

```
  return wrap(x / n) * n;
}

static inline sample wrapat(sample x, sample n) {
  return wrap(x * n) / n;
}

static inline sample clamp(sample x, sample lo, sample hi) {
  return fmin(fmax(x, lo), hi);
}

static inline sample mix(sample x, sample y, sample t) {
  return (1 - t) * x + t * y;
}

static inline sample trisaw(sample x, sample t) {
  sample s = clamp(t, (sample)0.0000001, (sample)0.9999999);
  sample y = clamp(x, 0, 1);
  return y < s ? y / s : (1 - y) / (1 - s);
}

static inline sample noise() {
  return 2 * (rand() / (sample) RAND_MAX - (sample)0.5);
}

static inline sample ABS(sample x)
{
  return x < 0 ? -x : x;
}

static inline sample CLAMP(sample x, sample lo, sample hi)
{
  return lo < x ? x < hi ? x : hi : lo;
}

static inline sample WRAP(sample x)
{
  return x - floor(x);
}

static inline sample FMOD(sample x, sample y)
{
  return WRAP(x / y) * y;
}

static const double PI   = 3.141592653589793;
static const double PI_2 = 1.5707963267948966;
static const double PI_4 = 0.7853981633974483;

static inline double TAN(double x)
{
  bool recip = false;
  x = ABS(x);
  if (unlikely(x >= PI_2))
  {
    x = FMOD(x + PI_2, PI) - PI_2;
  }
```

```
  if (unlikely(x >= PI_4))
  {
    recip = true;
    x = PI_2 - x;
  }
  // 0 <= x <= pi/4
  // [5/6] Padé approximant
  double x2 = x * x;
  double a = x * (10395.0 + x2 * (-1260.0 + x2 * 21.0));
  double b = 10395.0 + x2 * (-4725.0 + x2 * (210.0 - x2));
  return recip ? b / a : a / b;
}

static inline double TANH(double x)
{
//  x = CLAMP(x, -4.0, 4.0);
  // [5/6] Padé approximant
  double x2 = x * x;
  double a = x * (10395.0 + x2 * (1260.0 + x2 * 21.0));
  double b = 10395.0 + x2 * (4725.0 + x2 * (210.0 + x2));
  return a / b;
}

// pd-0.45-5/src/d_math.c

#define log10overten 0.23025850929940458
#define tenoverlog10 4.3429448190325175

static inline sample mtof(sample f) {
  return (sample)8.17579891564 * exp((sample)0.0577622650 * fmin(f, 1499));
}

static inline sample ftom(sample f) {
  return (sample)17.3123405046 * log((sample)0.12231220585 * f);
}

static inline sample dbtorms(sample f) {
  return exp((sample)0.5 * (sample)log10overten * (fmin(f, 870) - 100));
}

static inline sample rmstodb(sample f) {
  return 100 + 2 * (sample)tenoverlog10 * log(f);
}

static inline sample dbtopow(sample f) {
  return exp((sample)log10overten * (fmin(f, 485) - 100));
}

static inline sample powtodb(sample f) {
  return 100 + (sample)tenoverlog10 * log(f);
}


static inline sample bytebeat(int x) {
  return ((x & 0xFF) - 0x80) / (sample) 0x80;
}
```

```
static inline sample bitcrush(sample x, sample bits) {
  sample n = pow(2, bits);
  return round(x * n) / n;
}

#endif
```

10 client/dsp.h

```
#ifndef DSP_H
#define DSP_H 1

#include "dsp/func.h"
#include "dsp/filter.h"
#include "dsp/delay.h"
#include "dsp/osc.h"
#include "dsp/effects.h"

#endif
```

11 client/dsp/osc.h

```
#ifndef DSP_OSC_H
#define DSP_OSC_H 1

#include "dsp/func.h"

// need accurate phase otherwise low frequency range is too quantized
typedef struct { double phase; } PHASOR;

static inline double phasor(PHASOR *p, double hz) {
  // can't use wrap as it is defined for sample which might be lower precision
  p->phase += hz / SR;
  p->phase -= floor(p->phase);
  return p->phase;
}

#define WAVELEVELS 11

typedef struct {
  float wave[1 << (WAVELEVELS + 1)];
} WAVETABLE;

static inline void wavetable_saw(WAVETABLE *w) {
  for (int level = 0; level <= WAVELEVELS; ++level) {
    int n = 1 << level;
    int m = n >> 1;
    for (int i = 0; i < n; ++i) {
      w->wave[n + i] = 0;
      for (int k = 1; k < m; ++k) {
        double phase = twopi * ((k * i) % n) / (double) n;
        double sign = (k & 1) ? 1 : -1;
        double window = clamp(2 * (m - k) / (double) m, 0, 1);
        w->wave[n + i] += window * sign * sin(phase) / k;
      }
      w->wave[n + i] *= 2 / pi;
```

```
    }
  }
}

typedef struct {
  double phase;
} WAVE;

static inline double wave(WAVETABLE *wt, WAVE *w, double phase) {
  double delta = fabs(wrap(phase - w->phase));
  w->phase = phase;
  double l = -log2(delta);
  int l1 = floor(l);
  int l0 = l1 - 1;
  double lt = l - l1;
  l0 = clamp(l0, 0, WAVELEVELS);
  l1 = clamp(l1, 0, WAVELEVELS);
  int n0 = 1 << l0;
  int n1 = 1 << l1;
  double t = wrap(phase);
  double i0 = t * n0;
  int i00 = floor(i0);
  int i01 = (i00 + 1) % n0;
  double i0t = i0 - i00;
  double i1 = t * n1;
  int i10 = floor(i1);
  int i11 = (i10 + 1) % n1;
  double i1t = i1 - i10;
  return
    mix
    ( mix(wt->wave[n0 + i00], wt->wave[n0 + i01], i0t)
    , mix(wt->wave[n1 + i10], wt->wave[n1 + i11], i1t)
    , lt
    );
}


// <http://www.firstpr.com.au/dsp/pink-noise/>
// <http://www.firstpr.com.au/dsp/pink-noise/phil_burk_19990905_patest_pink.c>

/*
        patest_pink.c

        Generate Pink Noise using Gardner method.
        Optimization suggested by James McCartney uses a tree
        to select which random value to replace.

    x x x x x x x x x x x x x x x x
     x   x   x   x   x   x   x   x
       x       x       x       x
           x               x
                   x

    Tree is generated by counting trailing zeros in an increasing index.
        When the index is zero, no random number is selected.

    This program uses the Portable Audio library which is under development.
```

```
        For more information see:   http://www.audiomulch.com/portaudio/

    Author: Phil Burk, http://www.softsynth.com
95
        Revision History:

    Copyleft 1999 Phil Burk - No rights reserved.
*/
100
/************************************************************/
/* Calculate pseudo-random 32 bit number based on linear congruential method. */
//static unsigned long randSeed = 22222;  /* Change this for different random sequences. */
static inline unsigned long pink_GenerateRandomNumber(unsigned long randSeed) {
105         randSeed = (randSeed * 196314165) + 907633515;
        return randSeed;
}

#define PINK_MAX_RANDOM_ROWS   (30)
110 #define PINK_RANDOM_BITS       (24)
#define PINK_RANDOM_SHIFT      ((sizeof(long)*8)-PINK_RANDOM_BITS)

typedef struct {
        long    pink_Rows[PINK_MAX_RANDOM_ROWS];
115     long    pink_RunningSum;   /* Used to optimize summing of generators. */
        int     pink_Index;        /* Incremented each sample. */
        int     pink_IndexMask;    /* Index wrapped by ANDing with this mask. */
        double  pink_Scalar;       /* Used to scale within range of -1.0 to +1.0 */
  unsigned long seed;
120 } PINK;

/* Setup PinkNoise structure for N rows of generators. */
void pink_init(PINK *pink, int numRows, unsigned long seed)
{
125     int i;
        long pmax;
        pink->pink_Index = 0;
        pink->pink_IndexMask = (1<<numRows) - 1;
/* Calculate maximum possible signed random value. Extra 1 for white noise always added. */
130     pmax = (numRows + 1) * (1<<(PINK_RANDOM_BITS-1));
        pink->pink_Scalar = 1.0f / pmax;
/* Initialize rows. */
        for( i=0; i<numRows; i++ ) pink->pink_Rows[i] = 0;
        pink->pink_RunningSum = 0;
135   pink->seed = seed;
}

/* Generate Pink noise values between -1.0 and +1.0 */
double pink(PINK *pink ) {
140     long newRandom;
        long sum;
        double output;
```

```
	/* Increment and mask index. */
145		pink->pink_Index = (pink->pink_Index + 1) & pink->pink_IndexMask;

	/* If index is zero, don't update any random values. */
		if( pink->pink_Index != 0 )
		{
150		/* Determine how many trailing zeros in PinkIndex. */
		/* This algorithm will hang if n==0 so test first. */
			int numZeros = 0;
			int n = pink->pink_Index;
			while( (n & 1) == 0 )
155			{
				n = n >> 1;
				numZeros++;
			}

160		/* Replace the indexed ROWS random value.
		 * Subtract and add back to RunningSum instead of adding all the random
		 * values together. Only one changes each time.
		 */
			pink->pink_RunningSum -= pink->pink_Rows[numZeros];
165			newRandom = ((long) (pink->seed = pink_GenerateRandomNumber(pink↙
			    ↳ ->seed))) >> PINK_RANDOM_SHIFT;
			pink->pink_RunningSum += newRandom;
			pink->pink_Rows[numZeros] = newRandom;
		}

170	/* Add extra white noise value. */
		newRandom = ((long) (pink->seed = pink_GenerateRandomNumber(pink->seed))↙
		    ↳ ) >> PINK_RANDOM_SHIFT;
		sum = pink->pink_RunningSum + newRandom;

	/* Scale to range of -1.0 to 0.9999. */
175		output = pink->pink_Scalar * sum;

		return output;
	}

180

	/* analogue adsr */

	#define SMALL 1e-6
	#define ANTIDENORMAL (sample)1e-18
185
	#define RELEASE 0
	#define ATTACK  1
	#define DECAY   2

190	typedef struct {
	  sample trigger_1;
	  sample peak_1;
	  sample peak_2;
	  sample threshold_1;
195	  sample threshold_2;
	  double out_1;
	  double out_2;
	  int phase;
```

```
} ADSR;

static inline sample adsr(ADSR *adsr, sample trigger, sample attack, sample peak
    , sample decay, sample sustain, sample release, sample threshold) {
  ADSR *x = adsr;
  /* restore history */
  sample trigger_1   = x->trigger_1;
  sample peak_1      = x->peak_1;
  sample peak_2      = x->peak_2;
  sample threshold_1 = x->threshold_1;
  sample threshold_2 = x->threshold_2;
  double out_1       = x->out_1;
  double out_2       = x->out_2;
  int    phase       = x->phase;
  /* useful semi-constant */
  double T = -1000.0 / SR;
  /* scale milliseconds to multipliers */
  if (attack  < SMALL) { attack  = SMALL; } attack  = exp(T/attack);
  if (decay   < SMALL) { decay   = SMALL; } decay   = exp(T/decay);
  if (release < SMALL) { release = SMALL; } release = exp(T/release);
  /* which phase are we in? */
  if       (trigger - trigger_1 > 0) {
      /* rising trigger */
      phase = ATTACK;
  } else if (trigger - trigger_1 < 0) {
      /* falling trigger */
      phase = RELEASE;
  } else if ((out_1 > peak_1 * threshold_1) -
             (out_2 > peak_2 * threshold_2) > 0) {
      /* rising above peak threshold */
      phase = DECAY;
  }
  /* behaviour depends on phase */
  sample target = 0;
  double rate = 0;
  switch (phase) {
  case (ATTACK):  target = peak;    rate = attack;  break;
  case (DECAY):   target = sustain; rate = decay;   break;
  case (RELEASE): target = 0;       rate = release; break;
  default:        target = 0;       rate = 0;       break; /* never */
  }
  /* exponential decay to a target */
  double out = rate * out_1 + (1 - rate) * target;
  /* kill denormal by quantization */
  out += ANTIDENORMAL;
  out -= ANTIDENORMAL;
  /* recycle history */
  trigger_1   = trigger;
  peak_2      = peak_1;
  peak_1      = peak;
  threshold_2 = threshold_1;
  threshold_1 = threshold;
  out_2       = out_1;
  out_1       = out;
  /* save history */
  x->trigger_1   = trigger_1;
  x->peak_1      = peak_1;
```

```
255     x->peak_2      = peak_2;
        x->threshold_1 = threshold_1;
        x->threshold_2 = threshold_2;
        x->out_1       = out_1;
        x->out_2       = out_2;
260     x->phase       = phase;
        return out;
      }

      #undef SMALL
265   #undef ANTIDENORMAL
      #undef RELEASE
      #undef ATTACK
      #undef DECAY

270   #endif
```

12 client/go.c

```
      #include "go.h"

      int go(S *s, int channels, const float *in, float *out) {
        if (s->reloaded) {
 5        s->reloaded = 0;
        }
        for (int c = 0; c < channels; ++c) {
          out[c] = 0;
        }
10      return 0;
      }
```

13 client/go.h

```
      #ifndef GO_H
      #define GO_H 1

      typedef float sample;
 5    #define SR 44100
      #include "dsp.h"

      typedef struct {
        int reloaded;
10
      } S;

      #endif
```

14 client/Makefile

```
      GCC ?= gcc-8

      all: clive-client

 5    clive-client: clive-client.c
              $(GCC) -std=c99 -Wall -pedantic -Wextra -Wno-unused-parameter -O3 -o ↙
                  ↳ clive-client clive-client.c
```

15 client/native-sse.mk

```
GCC ?= gcc-8

../build/go.so: go.c dsp.h dsp/*.h
	$(GCC) -std=c99 -Wall -pedantic -Wextra -Wno-unused-parameter -O3 -march=native -mfpmath=sse -I. -shared -fPIC -o ../build/go.so go.c -lm
```

16 client/rpi3b-64.mk

```
GCC ?= aarch64-linux-gnu-gcc-8

../build/go.so: go.c dsp.h dsp/*.h
	$(GCC) -std=c99 -Wall -pedantic -Wextra -Wno-unused-parameter -O3 -march=armv8-a+crc -I. -shared -fPIC -o ../build/go.so go.c -lm
```

17 doc/acl.bst

```
% BibTeX 'acl' style file for BibTeX version 0.99c, LaTeX version 2.09
% This version was made by modifying 'aaai-named' format based on the master
% file by Oren Patashnik (PATASHNIK@SCORE.STANFORD.EDU)

% Copyright (C) 1985, all rights reserved.
% Modifications Copyright 1988, Peter F. Patel-Schneider
% Further modifictions by Stuart Shieber, 1991, and Fernando Pereira, 1992.
% Copying of this file is authorized only if either
% (1) you make absolutely no changes to your copy, including name, or
% (2) if you do make changes, you name it something other than
% btxbst.doc, plain.bst, unsrt.bst, alpha.bst, and abbrv.bst.
% This restriction helps ensure that all standard styles are identical.

% There are undoubtably bugs in this style.  If you make bug fixes,
% improvements, etc.  please let me know.  My e-mail address is:
%       pfps@spar.slb.com

%   Citation format: [author-last-name, year]
%                    [author-last-name and author-last-name, year]
%                    [author-last-name {\em et al.}, year]
%
%   Reference list ordering: alphabetical by author or whatever passes
%       for author in the absence of one.
%
% This BibTeX style has support for short (year only) citations.  This
% is done by having the citations actually look like
%         \citename{name-info, }year
% The LaTeX style has to have the following
%     \let\@internalcite\cite
%     \def\cite{\def\citename##1{##1}\@internalcite}
%     \def\shortcite{\def\citename##1{}\@internalcite}
%     \def\@biblabel#1{\def\citename##1{##1}[#1]\hfill}
% which makes \shortcite the macro for short citations.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Changes made by SMS for thesis style
%   no emphasis on "et al."
%   "Ph.D." includes periods (not "PhD")
```

```
%   moved year to immediately after author's name
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ENTRY
  { address
    author
    booktitle
    chapter
    edition
    editor
    howpublished
    institution
    journal
    key
    month
    note
    number
    organization
    pages
    publisher
    school
    series
    title
    type
    volume
    year
  }
  {}
  { label extra.label sort.label }

INTEGERS { output.state before.all mid.sentence after.sentence after.block }

FUNCTION {init.state.consts}
{ #0 'before.all :=
  #1 'mid.sentence :=
  #2 'after.sentence :=
  #3 'after.block :=
}

STRINGS { s t }

FUNCTION {output.nonnull}
{ 's :=
  output.state mid.sentence =
    { ", " * write$ }
    { output.state after.block =
        { add.period$ write$
          newline$
          "\newblock " write$
        }
        { output.state before.all =
            'write$
            { add.period$ " " * write$ }
          if$
        }
      if$
      mid.sentence 'output.state :=
    }
```

```
    if$
    s
}

FUNCTION {output}
{ duplicate$ empty$
    'pop$
    'output.nonnull
  if$
}

FUNCTION {output.check}
{ 't :=
  duplicate$ empty$
    { pop$ "empty " t * " in " * cite$ * warning$ }
    'output.nonnull
  if$
}

FUNCTION {output.bibitem}
{ newline$

  "\bibitem[" write$
  label write$
  "]{" write$

  cite$ write$
  "}" write$
  newline$
  ""
  before.all 'output.state :=
}

FUNCTION {fin.entry}
{ add.period$
  write$
  newline$
}

FUNCTION {new.block}
{ output.state before.all =
    'skip$
    { after.block 'output.state := }
  if$
}

FUNCTION {new.sentence}
{ output.state after.block =
    'skip$
    { output.state before.all =
        'skip$
        { after.sentence 'output.state := }
      if$
    }
  if$
}
```

```
FUNCTION {not}
{   { #0 }
    { #1 }
  if$
}

FUNCTION {and}
{   'skip$
    { pop$ #0 }
  if$
}

FUNCTION {or}
{   { pop$ #1 }
    'skip$
  if$
}

FUNCTION {new.block.checka}
{ empty$
    'skip$
    'new.block
  if$
}

FUNCTION {new.block.checkb}
{ empty$
  swap$ empty$
  and
    'skip$
    'new.block
  if$
}

FUNCTION {new.sentence.checka}
{ empty$
    'skip$
    'new.sentence
  if$
}

FUNCTION {new.sentence.checkb}
{ empty$
  swap$ empty$
  and
    'skip$
    'new.sentence
  if$
}

FUNCTION {field.or.null}
{ duplicate$ empty$
    { pop$ "" }
    'skip$
  if$
}
```

```
FUNCTION {emphasize}
{ duplicate$ empty$
    { pop$ "" }
    { "{\em " swap$ * "}" * }
  if$
}

INTEGERS { nameptr namesleft numnames }

FUNCTION {format.names}
{ 's :=
  #1 'nameptr :=
  s num.names$ 'numnames :=
  numnames 'namesleft :=
    { namesleft #0 > }

    { s nameptr "{ff~}{vv~}{ll}{, jj}" format.name$ 't :=

      nameptr #1 >
        { namesleft #1 >
            { ", " * t * }
            { numnames #2 >
                { "," * }
                'skip$
              if$
              t "others" =
                { " et~al." * }
                { " and " * t * }
              if$
            }
          if$
        }
        't
      if$
      nameptr #1 + 'nameptr :=
      namesleft #1 - 'namesleft :=
    }
  while$
}

FUNCTION {format.authors}
{ author empty$
    { "" }
    { author format.names }
  if$
}

FUNCTION {format.editors}
{ editor empty$
    { "" }
    { editor format.names
      editor num.names$ #1 >
        { ", editors" * }
        { ", editor" * }
      if$
    }
  if$
```

```
}

FUNCTION {format.title}
{ title empty$
    { "" }

    { title "t" change.case$ }

  if$
}

FUNCTION {n.dashify}
{ 't :=
  ""
    { t empty$ not }
    { t #1 #1 substring$ "-" =
        { t #1 #2 substring$ "--" = not
            { "--" *
              t #2 global.max$ substring$ 't :=
            }
            {   { t #1 #1 substring$ "-" = }
                { "-" *
                  t #2 global.max$ substring$ 't :=
                }
              while$
            }
          if$
        }
        { t #1 #1 substring$ *
          t #2 global.max$ substring$ 't :=
        }
      if$
    }
  while$
}

FUNCTION {format.date}
{ year empty$
    { month empty$
        { "" }
        { "there's a month but no year in " cite$ * warning$
          month
        }
      if$
    }
    { month empty$
        { "" }
        { month }
      if$
    }
  if$
}

FUNCTION {format.btitle}
{ title emphasize
}
```

```
FUNCTION {tie.or.space.connect}
{ duplicate$ text.length$ #3 <
    { "~" }
    { " " }
  if$
  swap$ * *
}

FUNCTION {either.or.check}
{ empty$
    'pop$
    { "can't use both " swap$ * " fields in " * cite$ * warning$ }
  if$
}

FUNCTION {format.bvolume}
{ volume empty$
    { "" }
    { "volume" volume tie.or.space.connect
      series empty$
        'skip$
        { " of " * series emphasize * }
      if$
      "volume and number" number either.or.check
    }
  if$
}

FUNCTION {format.number.series}
{ volume empty$
    { number empty$
        { series field.or.null }
        { output.state mid.sentence =
            { "number" }
            { "Number" }
          if$
          number tie.or.space.connect
          series empty$
            { "there's a number but no series in " cite$ * warning$ }
            { " in " * series * }
          if$
        }
      if$
    }
    { "" }
  if$
}

FUNCTION {format.edition}
{ edition empty$
    { "" }
    { output.state mid.sentence =
        { edition "l" change.case$ " edition" * }
        { edition "t" change.case$ " edition" * }
      if$
    }
  if$
```

```
}

INTEGERS { multiresult }

FUNCTION {multi.page.check}
{ 't :=
  #0 'multiresult :=
    { multiresult not
      t empty$ not
      and
    }
    { t #1 #1 substring$
      duplicate$ "-" =
      swap$ duplicate$ "," =
      swap$ "+" =
      or or
        { #1 'multiresult := }
        { t #2 global.max$ substring$ 't := }
      if$
    }
  while$
  multiresult
}

FUNCTION {format.pages}
{ pages empty$
    { "" }
    { pages multi.page.check
        { "pages" pages n.dashify tie.or.space.connect }
        { "page" pages tie.or.space.connect }
      if$
    }
  if$
}

FUNCTION {format.year.label}
{ year extra.label *
}

FUNCTION {format.vol.num.pages}
{ volume field.or.null
  number empty$
    'skip$
    { "(" number * ")" * *
      volume empty$
        { "there's a number but no volume in " cite$ * warning$ }
        'skip$
      if$
    }
  if$
  pages empty$
    'skip$
    { duplicate$ empty$
        { pop$ format.pages }
        { ":" * pages n.dashify * }
      if$
    }
```

```
    if$
  }

  FUNCTION {format.chapter.pages}
  { chapter empty$
      'format.pages
      { type empty$
          { "chapter" }
          { type "l" change.case$ }
        if$
        chapter tie.or.space.connect
        pages empty$
          'skip$
          { ", " * format.pages * }
        if$
      }
    if$
  }

  FUNCTION {format.in.ed.booktitle}
  { booktitle empty$
      { "" }
      { editor empty$
          { "In " booktitle emphasize * }
          { "In " format.editors * ", " * booktitle emphasize * }
        if$
      }
    if$
  }

  FUNCTION {empty.misc.check}
  { author empty$ title empty$ howpublished empty$
    month empty$ year empty$ note empty$
    and and and and and

    key empty$ not and

      { "all relevant fields are empty in " cite$ * warning$ }
      'skip$
    if$
  }

  FUNCTION {format.thesis.type}
  { type empty$
      'skip$
      { pop$
        type "t" change.case$
      }
    if$
  }

  FUNCTION {format.tr.number}
  { type empty$
      { "Technical Report" }
      'type
    if$
    number empty$
```

```
    { "t" change.case$ }
    { number tie.or.space.connect }
  if$
}

FUNCTION {format.article.crossref}
{ key empty$
    { journal empty$
        { "need key or journal for " cite$ * " to crossref " * crossref *
          warning$
          ""
        }
        { "In {\em " journal * "\/}" * }
      if$
    }
    { "In " key * }
  if$
  " \cite{" * crossref * "}" *
}

FUNCTION {format.crossref.editor}
{ editor #1 "{vv~}{ll}" format.name$
  editor num.names$ duplicate$
  #2 >
    { pop$ " et~al." * }
    { #2 <
        'skip$
        { editor #2 "{ff }{vv }{ll}{ jj}" format.name$ "others" =
            { " et~al." * }
            { " and " * editor #2 "{vv~}{ll}" format.name$ * }
          if$
        }
      if$
    }
  if$
}

FUNCTION {format.book.crossref}
{ volume empty$
    { "empty volume in " cite$ * "'s crossref of " * crossref * warning$
      "In "
    }
    { "Volume" volume tie.or.space.connect
      " of " *
    }
  if$
  editor empty$
  editor field.or.null author field.or.null =
  or
    { key empty$
        { series empty$
            { "need editor, key, or series for " cite$ * " to crossref " *
              crossref * warning$
              "" *
            }
            { "{\em " * series * "\/}" * }
          if$
```

```
        }
        { key * }
      if$
    }
    { format.crossref.editor * }
  if$
  " \cite{" * crossref * "}" *
}

FUNCTION {format.incoll.inproc.crossref}
{ editor empty$
  editor field.or.null author field.or.null =
  or
    { key empty$
        { booktitle empty$
            { "need editor, key, or booktitle for " cite$ * " to crossref " *
              crossref * warning$
              ""
            }
            { "In {\em " booktitle * "\/}" * }
          if$
        }
        { "In " key * }
      if$
    }
    { "In " format.crossref.editor * }
  if$
  " \cite{" * crossref * "}" *
}

FUNCTION {article}
{ output.bibitem
  format.authors "author" output.check
  new.block
  format.year.label "year" output.check
  new.block
  format.title "title" output.check
  new.block
  crossref missing$
    { journal emphasize "journal" output.check
      format.vol.num.pages output
      format.date output
    }
    { format.article.crossref output.nonnull
      format.pages output
    }
  if$
  new.block
  note output
  fin.entry
}

FUNCTION {book}
{ output.bibitem
  author empty$
    { format.editors "author and editor" output.check }
    { format.authors output.nonnull
```

```
        crossref missing$
610       { "author and editor" editor either.or.check }
          'skip$
        if$
      }
    if$
615   new.block
    format.year.label "year" output.check
    new.block
    format.btitle "title" output.check
    crossref missing$
620     { format.bvolume output
        new.block
        format.number.series output
        new.sentence
        publisher "publisher" output.check
625     address output
      }
      { new.block
        format.book.crossref output.nonnull
      }
630   if$
    format.edition output
    format.date output
    new.block
    note output
635   fin.entry
}

FUNCTION {booklet}
{ output.bibitem
640   format.authors output
    new.block
    format.year.label "year" output.check
    new.block
    format.title "title" output.check
645   howpublished address new.block.checkb
    howpublished output
    address output
    format.date output
    new.block
650   note output
    fin.entry
}

FUNCTION {inbook}
655 { output.bibitem
    author empty$
      { format.editors "author and editor" output.check }
      { format.authors output.nonnull
        crossref missing$
660       { "author and editor" editor either.or.check }
          'skip$
        if$
      }
    if$
665   format.year.label "year" output.check
```

```
  new.block
  new.block
  format.btitle "title" output.check
  crossref missing$
    { format.bvolume output
      format.chapter.pages "chapter and pages" output.check
      new.block
      format.number.series output
      new.sentence
      publisher "publisher" output.check
      address output
    }
    { format.chapter.pages "chapter and pages" output.check
      new.block
      format.book.crossref output.nonnull
    }
  if$
  format.edition output
  format.date output
  new.block
  note output
  fin.entry
}

FUNCTION {incollection}
{ output.bibitem
  format.authors "author" output.check
  new.block
  format.year.label "year" output.check
  new.block
  format.title "title" output.check
  new.block
  crossref missing$
    { format.in.ed.booktitle "booktitle" output.check
      format.bvolume output
      format.number.series output
      format.chapter.pages output
      new.sentence
      publisher "publisher" output.check
      address output
      format.edition output
      format.date output
    }
    { format.incoll.inproc.crossref output.nonnull
      format.chapter.pages output
    }
  if$
  new.block
  note output
  fin.entry
}

FUNCTION {inproceedings}
{ output.bibitem
  format.authors "author" output.check
  new.block
  format.year.label "year" output.check
```

```
  new.block
  format.title "title" output.check
  new.block
  crossref missing$
    { format.in.ed.booktitle "booktitle" output.check
      format.bvolume output
      format.number.series output
      format.pages output
      address empty$
        { organization publisher new.sentence.checkb
          organization output
          publisher output
          format.date output
        }
        { address output.nonnull
          format.date output
          new.sentence
          organization output
          publisher output
        }
      if$
    }
    { format.incoll.inproc.crossref output.nonnull
      format.pages output
    }
  if$
  new.block
  note output
  fin.entry
}

FUNCTION {conference} { inproceedings }

FUNCTION {manual}
{ output.bibitem
  author empty$
    { organization empty$
        'skip$
        { organization output.nonnull
          address output
        }
      if$
    }
    { format.authors output.nonnull }
  if$
  format.year.label "year" output.check
  new.block
  new.block
  format.btitle "title" output.check
  author empty$
    { organization empty$
        { address new.block.checka
          address output
        }
        'skip$
      if$
    }
```

```
780       { organization address new.block.checkb
            organization output
            address output
          }
        if$
785     format.edition output
        format.date output
        new.block
        note output
        fin.entry
790   }

      FUNCTION {mastersthesis}
      { output.bibitem
        format.authors "author" output.check
795     new.block
        format.year.label "year" output.check
        new.block
        format.title "title" output.check
        new.block
800     "Master's thesis" format.thesis.type output.nonnull
        school "school" output.check
        address output
        format.date output
        new.block
805     note output
        fin.entry
      }

      FUNCTION {misc}
810   { output.bibitem
        format.authors output
        new.block
        format.year.label output
        new.block
815     title howpublished new.block.checkb
        format.title output
        howpublished new.block.checka
        howpublished output
        format.date output
820     new.block
        note output
        fin.entry
        empty.misc.check
      }
825
      FUNCTION {phdthesis}
      { output.bibitem
        format.authors "author" output.check
        new.block
830     format.year.label "year" output.check
        new.block
        format.btitle "title" output.check
        new.block
        "{Ph.D.} thesis" format.thesis.type output.nonnull
835     school "school" output.check
        address output
```

```
  format.date output
  new.block
  note output
  fin.entry
}

FUNCTION {proceedings}
{ output.bibitem
  editor empty$
    { organization output }
    { format.editors output.nonnull }
  if$
  new.block
  format.year.label "year" output.check
  new.block
  format.btitle "title" output.check
  format.bvolume output
  format.number.series output
  address empty$
    { editor empty$
        { publisher new.sentence.checka }
        { organization publisher new.sentence.checkb
          organization output
        }
      if$
      publisher output
      format.date output
    }
    { address output.nonnull
      format.date output
      new.sentence
      editor empty$
        'skip$
        { organization output }
      if$
      publisher output
    }
  if$
  new.block
  note output
  fin.entry
}

FUNCTION {techreport}
{ output.bibitem
  format.authors "author" output.check
  new.block
  format.year.label "year" output.check
  new.block
  format.title "title" output.check
  new.block
  format.tr.number output.nonnull
  institution "institution" output.check
  address output
  format.date output
  new.block
  note output
```

```
  fin.entry
}

FUNCTION {unpublished}
{ output.bibitem
  format.authors "author" output.check
  new.block
  format.year.label "year" output.check
  new.block
  format.title "title" output.check
  new.block
  note "note" output.check
  format.date output
  fin.entry
}

FUNCTION {default.type} { misc }

MACRO {jan} {"January"}

MACRO {feb} {"February"}

MACRO {mar} {"March"}

MACRO {apr} {"April"}

MACRO {may} {"May"}

MACRO {jun} {"June"}

MACRO {jul} {"July"}

MACRO {aug} {"August"}

MACRO {sep} {"September"}

MACRO {oct} {"October"}

MACRO {nov} {"November"}

MACRO {dec} {"December"}

MACRO {acmcs} {"ACM Computing Surveys"}

MACRO {acta} {"Acta Informatica"}

MACRO {cacm} {"Communications of the ACM"}

MACRO {ibmjrd} {"IBM Journal of Research and Development"}

MACRO {ibmsj} {"IBM Systems Journal"}

MACRO {ieeese} {"IEEE Transactions on Software Engineering"}

MACRO {ieeetc} {"IEEE Transactions on Computers"}

MACRO {ieeetcad}
```

```
 {"IEEE Transactions on Computer-Aided Design of Integrated Circuits"}

MACRO {ipl} {"Information Processing Letters"}

MACRO {jacm} {"Journal of the ACM"}

MACRO {jcss} {"Journal of Computer and System Sciences"}

MACRO {scp} {"Science of Computer Programming"}

MACRO {sicomp} {"SIAM Journal on Computing"}

MACRO {tocs} {"ACM Transactions on Computer Systems"}

MACRO {tods} {"ACM Transactions on Database Systems"}

MACRO {tog} {"ACM Transactions on Graphics"}

MACRO {toms} {"ACM Transactions on Mathematical Software"}

MACRO {toois} {"ACM Transactions on Office Information Systems"}

MACRO {toplas} {"ACM Transactions on Programming Languages and Systems"}

MACRO {tcs} {"Theoretical Computer Science"}

READ

FUNCTION {sortify}
{ purify$
  "l" change.case$
}

INTEGERS { len }

FUNCTION {chop.word}
{ 's :=
  'len :=
  s #1 len substring$ =
    { s len #1 + global.max$ substring$ }
    's
  if$
}

INTEGERS { et.al.char.used }

FUNCTION {initialize.et.al.char.used}
{ #0 'et.al.char.used :=
}

EXECUTE {initialize.et.al.char.used}

FUNCTION {format.lab.names}
{ 's :=
  s num.names$ 'numnames :=

  numnames #1 =
```

```
      { s #1 "{vv }{ll}" format.name$ }
      { numnames #2 =
          { s #1 "{vv }{ll }and " format.name$ s #2 "{vv }{ll}" format.name$ *
          }
          { s #1 "{vv }{ll }\bgroup et al.\egroup " format.name$ }
        if$
      }
    if$

}

FUNCTION {author.key.label}
{ author empty$
    { key empty$

        { cite$ #1 #3 substring$ }

        { key #3 text.prefix$ }
      if$
    }
    { author format.lab.names }
  if$
}

FUNCTION {author.editor.key.label}
{ author empty$
    { editor empty$
        { key empty$

            { cite$ #1 #3 substring$ }

            { key #3 text.prefix$ }
          if$
        }
        { editor format.lab.names }
      if$
    }
    { author format.lab.names }
  if$
}

FUNCTION {author.key.organization.label}
{ author empty$
    { key empty$
        { organization empty$

            { cite$ #1 #3 substring$ }

            { "The " #4 organization chop.word #3 text.prefix$ }
          if$
        }
        { key #3 text.prefix$ }
      if$
    }
    { author format.lab.names }
  if$
}
```

```
1065
FUNCTION {editor.key.organization.label}
{ editor empty$
    { key empty$
        { organization empty$
1070
            { cite$ #1 #3 substring$ }

            { "The " #4 organization chop.word #3 text.prefix$ }
          if$
1075      }
          { key #3 text.prefix$ }
        if$
    }
    { editor format.lab.names }
1080  if$
}

FUNCTION {calc.label}
{ type$ "book" =
1085  type$ "inbook" =
  or
    'author.editor.key.label
    { type$ "proceedings" =
        'editor.key.organization.label
1090    { type$ "manual" =
            'author.key.organization.label
            'author.key.label
          if$
        }
1095    if$
    }
  if$
  duplicate$

1100  "\protect\citename{" swap$ * "}" *
  year field.or.null purify$ *
  'label :=
  year field.or.null purify$ *

1105  sortify 'sort.label :=
}

FUNCTION {sort.format.names}
{ 's :=
1110  #1 'nameptr :=
  ""
  s num.names$ 'numnames :=
  numnames 'namesleft :=
    { namesleft #0 > }
1115    { nameptr #1 >
        { "   " * }
        'skip$
      if$

1120    s nameptr "{vv{ } }{ll{ }}{  ff{ }}{  jj{ }}" format.name$ 't :=
```

```
        nameptr numnames = t "others" = and
          { "et al" * }
          { t sortify * }
        if$
        nameptr #1 + 'nameptr :=
        namesleft #1 - 'namesleft :=
      }
    while$
}

FUNCTION {sort.format.title}
{ 't :=
  "A " #2
    "An " #3
      "The " #4 t chop.word
    chop.word
  chop.word
  sortify
  #1 global.max$ substring$
}

FUNCTION {author.sort}
{ author empty$
    { key empty$
        { "to sort, need author or key in " cite$ * warning$
          ""
        }
        { key sortify }
      if$
    }
    { author sort.format.names }
  if$
}

FUNCTION {author.editor.sort}
{ author empty$
    { editor empty$
        { key empty$
            { "to sort, need author, editor, or key in " cite$ * warning$
              ""
            }
            { key sortify }
          if$
        }
        { editor sort.format.names }
      if$
    }
    { author sort.format.names }
  if$
}

FUNCTION {author.organization.sort}
{ author empty$
    { organization empty$
        { key empty$
            { "to sort, need author, organization, or key in " cite$ * warning$
              ""
```

```
            }
            { key sortify }
          if$
        }
        { "The " #4 organization chop.word sortify }
      if$
    }
    { author sort.format.names }
  if$
}

FUNCTION {editor.organization.sort}
{ editor empty$
    { organization empty$
        { key empty$
            { "to sort, need editor, organization, or key in " cite$ * warning$
              ""
            }
            { key sortify }
          if$
        }
        { "The " #4 organization chop.word sortify }
      if$
    }
    { editor sort.format.names }
  if$
}

FUNCTION {presort}

{ calc.label
  sort.label
  "    "
  *
  type$ "book" =

  type$ "inbook" =
  or
    'author.editor.sort
    { type$ "proceedings" =
        'editor.organization.sort
        { type$ "manual" =
            'author.organization.sort
            'author.sort
          if$
        }
      if$
    }
  if$

  *

  "    "
  *
  year field.or.null sortify
  *
  "    "
```

```
    *
    title field.or.null
    sort.format.title
    *
    #1 entry.max$ substring$
    'sort.key$ :=
}

ITERATE {presort}

SORT

STRINGS { longest.label last.sort.label next.extra }

INTEGERS { longest.label.width last.extra.num }

FUNCTION {initialize.longest.label}
{ "" 'longest.label :=
  #0 int.to.chr$ 'last.sort.label :=
  "" 'next.extra :=
  #0 'longest.label.width :=
  #0 'last.extra.num :=
}

FUNCTION {forward.pass}
{ last.sort.label sort.label =
    { last.extra.num #1 + 'last.extra.num :=
      last.extra.num int.to.chr$ 'extra.label :=
    }
    { "a" chr.to.int$ 'last.extra.num :=
      "" 'extra.label :=
      sort.label 'last.sort.label :=
    }
  if$
}

FUNCTION {reverse.pass}
{ next.extra "b" =
    { "a" 'extra.label := }
    'skip$
  if$
  label extra.label * 'label :=
  label width$ longest.label.width >
    { label 'longest.label :=
      label width$ 'longest.label.width :=
    }
    'skip$
  if$
  extra.label 'next.extra :=
}

EXECUTE {initialize.longest.label}

ITERATE {forward.pass}

REVERSE {reverse.pass}
```

```
FUNCTION {begin.bib}

{ et.al.char.used
    { "\newcommand{\etalchar}[1]{$^{#1}$}" write$ newline$ }
    'skip$
  if$
  preamble$ empty$

    'skip$
    { preamble$ write$ newline$ }
  if$

  "\begin{thebibliography}{" "}" * write$ newline$

}

EXECUTE {begin.bib}

EXECUTE {init.state.consts}

ITERATE {call.type$}

FUNCTION {end.bib}
{ newline$
  "\end{thebibliography}" write$ newline$
}

EXECUTE {end.bib}
```

18 doc/getting-started.md

```
---
title: Getting started with Clive
author: Claude Heiland-Allen
date: 2018-09-03
classoption: aspectratio=149
fontfamily: lato
fontfamilyoptions: default
fontsize: 14pt
header-includes:
  - \definecolor{mathr}{rgb}{0.5, 0.25, 0.25}
  - \usecolortheme[named=mathr]{structure}
---

# Getting started with Clive

- Linux only (sorry)

- install dependencies, configure sudo and JACK

- clone the repository

- launching, exiting

- an example

## Dependencies
```

Debian

```
    # apt install \
        sudo git ca-certificates \
        build-essential pkg-config \
        libjack-jackd2-dev qjackctl \
        cpufrequtils ecasound \
        xterm htop geany \
        python-pygments
```

Configuring ‘sudo‘ for ‘cpufrequtils‘

Replace ‘claude‘ with your username and ‘latte‘ with your machine name:

```
    $ sudo visudo /etc/sudoers.d/cpufreq-set
    claude latte = (root) NOPASSWD: \
      /usr/bin/cpufreq-set
```

The file contents should be one line without ‘\‘, just split to fit slides!

This allows to change CPU frequency governor without password.

Configuring ‘JACK‘ for realtime

Provided by JACK packaging on Debian:

```
    $ cat /etc/security/limits.d/audio.conf
    @audio   -  rtprio     95
    @audio   -  memlock    unlimited
```

To check that you are in the audio group:

```
    $ groups
```

Download ‘clive‘ repository

```
    git clone https://code.mathr.co.uk/clive-core.git
    cd clive-core
    git config user.email "you@example.com"
    git config user.name "Your Name"
```

Launching

Launch ‘qjackctl‘ and configure your sound card.

Launch ‘clive‘:

```
    cd clive-core/src
    ./start.sh
```

After a short delay, there should be 2 terminal windows on the left and the ‘geany‘ text editor which can be resized to fit on the right.

Edit ‘go.c‘, press Ctrl-S to save, which will recompile and reload the code.

```
## Exiting

- Ctrl-C in the terminal running `./start.sh`

- Stop and rewind JACK transport

- `git checkout master`

## Example: a metronome

    git checkout origin/metronome
    git checkout -b metronome

## Example: a metronome

\fontsize{12pt}{13}\selectfont

```C
#define SR 48000
#include "dsp.h"

typedef struct {
  int reloaded;
  PHASOR clock, osc;
} S;

int go(S *s, int channels, const float *in, float *out) {
  if (s->reloaded) { s->reloaded = 0; }
  double env = phasor(&s->clock, 125/60.0) < 0.25;
  double osc = sin(twopi * phasor(&s->osc, 440));
  double o = env * osc;
  for (int c = 0; c < channels; ++c) { out[c] = o; }
  return 0;
}
```

Clive links

- <claude@mathr.co.uk>

- <https://mathr.co.uk/clive>
```

19 doc/.gitignore

```
*.aux
*.bbl
*.blg
*.log
*.out
*.pdf
```

20 doc/lac2018.sty

```
%
% lac2018.sty
%
```

```
\typeout{}
\typeout{LAC2018 Proceedings style -- January 2018}
%\typeout{only slightly different from}
%\typeout{Coling 2004 Proceedings style -- February 2004}
%\typeout{only slightly different from}
%\typeout{ACL-COLING 1998 Proceedings style -- March 31st 1998}
\typeout{}
%
% ---------------------------------------------------------------------
%
% This is the LaTeX style file for the Linux Audio Conference
% It is an almost unmodified copy of colacl.sty, the style file
% of Coling (International Conference on Computational Linguistics).
% Many thanks to Coling for their permission to use it!
% The original may be found here:
% http://www.issco.unige.ch/coling2004/coling2004downloads.html
%
% ---------------------------------------------------------------------
%
% This is the LaTeX style file for ACL-COLING 1998.  It is based on
% a series of similar files prepared for previous conferences by
% Fernando Pereira, Paul Jacobs, Stuart Shieber, Peter F.
% Patel-Schneider and others.  Various changes have been made, chiefly
% to save space in the final output or remove redundant definitions.
%
% colacl.sty is designed for use as a package or option with the
% standard LaTeX article class, and the BibTeX style acl.bst.
%
% Author/title and citation formatting differs slightly from standard
% LaTeX; see AUTHOR FORMATS and CITATION FORMATS below for more
% information.
%
% This file is supplied as a hopefully convenient implementation of
% some of the "instructions for authors" repeated below.  It is not
% guaranteed to work in any given LaTeX installation or in conjunction
% with any given class, package or style, and it is not intended as
% a LaTeX tutorial.
%
% ---------------------------------------------------------------------
% Instructions for authors
%
%   (i) Maximum length of full papers: 8 pages.
%
%  (ii) Paper size: A4.
%
% (iii) Margins: set so that text lies within a rectangle 9in (23cm)
%       high and 6.5in (16.5cm) wide.
%
%  (iv) Body of text to be set in two columns.  Full-width figures
%       (i.e. using \begin{figure*}) and tables may be used if
%       necessary.
%
%   (v) Use standard fonts, e.g. Computer Modern Roman, Times Roman, no
%       smaller than 10pt.
%
%  (vi) No page numbering.
%
```

```
% Items (iii), (iv) and (vi) are handled by this file, and should
% therefore not be overridden by resetting \textwidth, \textheight,
% \pagestyle etc. in your document, or by using styles or packages
% which have the same effect.
%
% ----------------------------------------------------------------------
% To convert papers prepared with colaclsub.sty to the final format
% for use with colacl.sty:
%
% (1) Remove commands specific to the original submission format
%     (\type, \subject, \contact, \conference, \makeidpage).
%
% (2) Replace \summary{...} with an abstract, using the normal
%     abstract environment, placed after \maketitle.
%
% A simple document template:
%
%   \documentclass[11pt,letterpaper]{article}
%   \usepackage{lac2012}
%   \title{...}
%   \author{...}              % see below for possible formats
%   \begin{document}
%   \maketitle
%   \begin{abstract}
%   ...                       % contents of abstract
%   \end{abstract}
%   ...                       % contents of article
%   \bibliographystyle{acl}  % use acl.bst
%   \bibliography{...}
%   \end{document}
%
% Users of obsolete LaTeX versions can try:
%
%   \documentstyle[lac2012]{article}    % or [11pt,lac2012]
%   \title{...}
%   ...
%
%
% ----------------------------------------------------------------------
% AUTHOR FORMATS
%
% Author information can be set in various styles.
%
% For several authors from the same institution:
% \author{Author 1 \and ... \and Author n \\
%         Address line \\ ... \\ Address line}
% if the names do not fit well on one line use
%         Author 1 \\ {\bf Author 2} \\ ... \\ {\bf Author n} \\
%
% For authors from different institutions:
% \author{Author 1 \\ Address line \\  ... \\ Address line
%         \And  ... \And
%         Author n \\ Address line \\ ... \\ Address line}
%
% To start a separate "row" of authors use \AND, as in
% \author{Author 1 \\ Address line \\  ... \\ Address line
%         \AND
```

```
%          Author 2 \\ Address line \\ ... \\ Address line \And
%          Author 3 \\ Address line \\ ... \\ Address line}
%
% If the title and author information does not fit in the area allocated,
% place \setlength\titlebox{<new height>} after \usepackage{colacl},
% where <new height> can be something larger than 2.0in
%
% ---------------------------------------------------------------------
% CITATION FORMATS
%
% Three possible citation formats:
% "\cite{...}"       produces a citation like "(Author, 1999)"
% "\shortcite{...}"  produces a citation like "(1999)"
% "\newcite{...}"    produces a citation like "Author (1999)"
%
% All three take an optional argument which can be used to add page
% references, etc.:
% "\newcite[1--6]{...}" produces a citation like "Author (1999, 1--6)"
%
% ---------------------------------------------------------------------
% IF IT DOESN'T WORK
%
% The error message "File `colacl.sty' not found." indicates that this
% file has not been installed in a location which is visible to your
% LaTeX.  Try putting it in the same directory as your paper, and
% running LaTeX there.  Consult your `Local Guide' documentation or
% your system administrator to find out how LaTeX searches for input
% files.
%
% "\documentclass..." is a LaTeX2e declaration.  An error message
% "Undefined control sequence." followed by a line ending in
% "\documentclass" indicates that you have used this with an obsolete
% LaTeX installation.  Use the "\documentstyle" variant shown above.
%
% As a last resort, forget about colacl.sty and simply copy the
% following lines (uncommented, obviously) into your document before
% the "\begin{document}":
%
% \setlength\topmargin{0.0in}
% \setlength\oddsidemargin{-0.0in}
% \setlength\textheight{9.0in}
% \setlength\textwidth{6.5in}
% \setlength\columnsep{0.25in}
% \setlength\headheight{0pt}
% \setlength\headsep{0pt}
% \thispagestyle{empty}
% \pagestyle{empty}
% \flushbottom
% \twocolumn
% \sloppy
%
% Some interactions with other packages may still occur.  In order to
% remove the page number from the first page, you may have to place the
% "\thispagestyle{empty}" command immediately after "\maketitle".
%
% ---------------------------------------------------------------------
% NOTE:  Some laser printers have a serious problem printing TeX output.
```

```
% These printing devices, commonly known as "write-white" laser
% printers, tend to make characters too light.  To get around this
% problem, a darker set of fonts must be created for these devices.
%
% ---------------------------------------------------------------------
% Physical page layout - slightly modified from IJCAI by pj
% Physical page layout - slightly modified by rg for LAC
\newlength\myavierh
\setlength\myavierh{297mm}
%\message{The page height is \the\paperheight}
\ifdim \paperheight = \myavierh
\message{ !! Using ISO-A4 paper}
%% A4paper
\setlength\topmargin{0.0in}
\setlength\oddsidemargin{-0.0in}
\setlength\textheight{9.5in}   %%sm was 9.0in
\setlength\textwidth{6.5in}
\setlength\columnsep{0.25in}
\newlength\titlebox
\setlength\titlebox{2.0in}              % was 2.25in
\setlength\headheight{0pt}
\setlength\headsep{0pt}
\setlength\footskip{0pt}                % irrelevant when no footers.

\else
%% US-Letter
\message{ !! Using US-Letter paper}
\setlength{\pdfpagewidth}{8.5in}
\setlength{\pdfpageheight}{11in}
\setlength\topmargin{0.0in}
\setlength\oddsidemargin{-0.2in}
\setlength\textheight{8.8in}
\setlength\textwidth{6.9in}
\setlength\columnsep{0.4in}
\newlength\titlebox
\setlength\titlebox{2.1in}
\setlength\headheight{0pt}
\setlength\headsep{0pt}
\setlength\footskip{0pt}      % irrelevant when no footers.
\fi

\pagestyle{empty}                       % no page numbers
\thispagestyle{empty}                   % no page numbers
\flushbottom
\twocolumn
\sloppy

% We're never going to need a table of contents, so just flush it to
% save space --- suggested by drstrip@sandia-2
\def\addcontentsline#1#2#3{}

% ---------------------------------------------------------------------
% Title stuff, taken from deproc.

\def\maketitle{%
  \par%
  \begingroup%
```

```
      \def\thefootnote{\fnsymbol{footnote}}%
      \def\@makefnmark{\rlap{$^{\@thefnmark}$\hss}}%
      % no paragraph breaks in \thanks
      \long\def\@makefntext##1{%
                  \parindent 1em\noindent%
                  \hbox to 1em{$^{\@thefnmark}$}##1}
      \twocolumn[\@maketitle] \@thanks%
  \endgroup%
  \setcounter{footnote}{0}%
  \let\maketitle\relax\let\@maketitle\relax%
  \gdef\@thanks{}\gdef\@author{}\gdef\@title{}%
  \let\thanks\relax}

% some vertical space removed here: skip above and below title
%
\def\@maketitle{%
  \vbox to \titlebox{%
    \hsize\textwidth\linewidth\hsize%
    \vskip 0.125in minus 0.05in%
    \centering{\Large\bf \@title \par}%
    \vskip 0.2in plus 0.1fil minus 0.1in
    {\def\and{\unskip\enspace{\rm and}\enspace}%
     \def\And{\end{tabular}\hss \egroup \hskip 1in plus 2fil
              \hbox to 0pt\bgroup\hss \begin{tabular}[t]{c}\bf}%
     \def\AND{\end{tabular}\hss\egroup \hfil\hfil\egroup
              \vskip 0.25in plus 1fil minus 0.125in
              \hbox to \linewidth\bgroup\large \hfil\hfil
              \hbox to 0pt\bgroup\hss \begin{tabular}[t]{c}\bf}
    \hbox to \linewidth \bgroup\large \hfil\hfil
    \hbox to 0pt\bgroup\hss \begin{tabular}[t]{c}\bf\@author
                            \end{tabular}\hss\egroup
    \hfil\hfil\egroup}
  \vskip 0.3in plus 2fil minus 0.1in
}}

% ------------------------------------------------------------------------
% abstract, changed for LAC2005

% \renewenvironment{abstract}{\section*{\centerline{Abstract}}}{}
\renewenvironment{abstract}{\section*{Abstract}}{}

% ------------------------------------------------------------------------
% keywords, added for LAC2005

\def\keywords{\section*{Keywords}}

% ------------------------------------------------------------------------
% bibliography and citations

% most of cite format is from aclsub.sty by SMS

% don't box citations, separate with ; and a space
% Replaced for multiple citations (pj)
% don't box citations and also add space, semicolon between multiple
% citations
%
\def\@citex[#1]#2{\if@filesw\immediate\write\@auxout{\string\citation{#2}}\fi
```

```
  \def\@citea{}\@cite{\@for\@citeb:=#2\do
290       {\@citea\def\@citea{; }\@ifundefined
          {b@\@citeb}{{\bf ?}\@warning
           {Citation `\@citeb' on page \thepage \space undefined}}%
     {\csname b@\@citeb\endcsname}}}{#1}}

295  % Allow short (name-less) citations, when used in
     % conjunction with a bibliography style that creates labels like
     %       \citename{<names>, }<year>
     %
     \let\@internalcite\cite
300  \def\cite{\def\citename##1{##1, }\@internalcite}
     \def\longcite{\def\citename##1{##1, }\@internalcite}

     \def\shortcite{\def\citename##1{}\@internalcite}
     \def\newcite{\def\citename##1{{##1}\ (}\@internalciteb}
305
     % Macros for \newcite, which leaves name in running text, and is
     % otherwise like \shortcite.
     %
     \def\@citexb[#1]#2{\if@filesw\immediate\write\@auxout{\string\citation{#2}}\fi
310    \def\@citea{}\@newcite{\@for\@citeb:=#2\do
         {\@citea\def\@citea{;\penalty\@m\ }\@ifundefined
            {b@\@citeb}{{\bf ?}\@warning
            {Citation `\@citeb' on page \thepage \space undefined}}%
     % gjr: hbox causes too many bad linebreaks
315  %\hbox{\csname b@\@citeb\endcsname}}}{#1}}
          {\csname b@\@citeb\endcsname}}}{#1}}

     \def\@internalciteb{%
       \@ifnextchar [{\@tempswatrue\@citexb}{\@tempswafalse\@citexb[]}}
320
     \def\@newcite#1#2{{#1\if@tempswa, #2\fi)}}

     % gjr: no labels in this bibliography style
     % \def\@biblabel#1{\def\citename##1{##1}[#1]\hfill}
325  \def\@biblabel#1{}

     % use square brackets after all
     %%% More changes made by SMS (originals in latex.tex)
     % Use parentheses instead of square brackets in the text.
330  % \def\@cite#1#2{({#1\if@tempswa , #2\fi})}

     % Don't put a label in the bibliography at all.  Just use the unlabeled format
     % instead.
     % gjr: removed \@mkboth -- no headers here.
335  % gjr: reduced vertical space between entries (plus was .33em)
     %

     \def\thebibliography#1{\section*{References}\list
      {[\arabic{enumi}]}{\settowidth\labelwidth{[#1]}\leftmargin\labelwidth
340   \advance\leftmargin\labelsep
      \usecounter{enumi}}
      \def\newblock{\hskip .11em plus .11em minus .07em}
      \sloppy\clubpenalty4000\widowpenalty4000
      \sfcode`\.=1000\relax}
345  \let\endthebibliography=\endlist
```

```
% \def\thebibliography#1{%
%   \section*{References}
%   \list{}{\setlength{\labelwidth}{0pt}
%            \setlength{\leftmargin}{\parindent}
%            \setlength{\itemsep}{0.11ex plus 0.11ex}
%            \setlength{\parsep}{0ex}
%            \setlength{\itemindent}{-\parindent}}
%   \def\newblock{\hskip .11em plus .11em minus -.07em}
%   \sloppy\clubpenalty4000\widowpenalty4000
%   \sfcode`\.=1000\relax}
% \let\endthebibliography=\endlist

% Allow for a bibliography of sources of attested examples
\def\thesourcebibliography#1{%
  \section*{Sources of Attested Examples}
  \list{}{\setlength{\labelwidth}{0pt}
          \setlength{\leftmargin}{\parindent}
          \setlength{\itemsep}{0.11ex plus 0.11ex}
          \setlength{\parsep}{0ex}
          \setlength{\itemindent}{-\parindent}}
  \def\newblock{\hskip .11em plus .11em minus -.07em}
  \sloppy\clubpenalty4000\widowpenalty4000
  \sfcode`\.=1000\relax}
\let\endthesourcebibliography=\endlist

\def\@lbibitem[#1]#2{\item[]\if@filesw
      { \def\protect##1{\string ##1\space}\immediate
        \write\@auxout{\string\bibcite{#2}{#1}}\fi\ignorespaces}}

\def\@bibitem#1{\item\if@filesw \immediate\write\@auxout
       {\string\bibcite{#1}{\the\c@enumi}}\fi\ignorespaces}

% ---------------------------------------------------------------------
% Section headings with less space

\def\section{%
    \@startsection{section}{1}{\z@}%
                  {-2.0ex plus -0.5ex minus -0.3ex}%
                  {0.8ex plus 0.3ex minus 0.2ex}%
                  {\large\bf\raggedright}}
\def\subsection{%
    \@startsection{subsection}{2}{\z@}%
                  {-1.4ex plus -0.4ex minus -0.2ex}%
                  {0.6ex plus 0.2ex minus 0.1ex}%
                  {\normalsize\bf\raggedright}}
\def\subsubsection{%
    \@startsection{subsubsection}{3}{\z@}%
                  {-0.8ex plus -0.3ex minus -0.1ex}%
                  {0.4ex plus 0.1ex minus 0.1ex}%
                  {\normalsize\bf\raggedright}}
\def\paragraph{%
    \@startsection{paragraph}{4}{\z@}%
                  {-0.8ex plus -0.3ex minus -0.1ex}%
                  {-1em}%
                  {\normalsize\bf}}
\def\subparagraph{%
```

```
    \@startsection{subparagraph}{5}{\parindent}%
                  {0.4ex plus 0.3ex minus 0.1ex}%
                  {-1em}%
                  {\normalsize\bf}}


% ------------------------------------------------------------------
% Footnotes

%\footnotesep 6.65pt %
%\skip\footins 9pt plus 4pt minus 2pt
%\def\footnoterule{\kern-3pt \hrule width 5pc \kern 2.6pt }
%\setcounter{footnote}{0}

% ------------------------------------------------------------------
% Lists and paragraphs

\setlength\parindent{1em}

\leftmargin 2em \leftmargini\leftmargin \leftmarginii 2em
\leftmarginiii 1.5em \leftmarginiv 1.0em \leftmarginv .5em \leftmarginvi .5em
\labelwidth\leftmargini\advance\labelwidth-\labelsep \labelsep 5pt

% ------------------------------------------------------------------
% Floats (figures, tables etc.)
%
% Allow a larger proportion of the column/page to be taken up with
% floats than the standard classes.  Also discourage the creation of
% columns/pages containing only floats.

% Maximum fraction of the page that can be occupied by floats:
%
\renewcommand\topfraction{.9}
\renewcommand\bottomfraction{.5}
\renewcommand\dbltopfraction{.9}        % 2-column floats

% Minimum fraction of page that can be occupied by text:
%
\renewcommand\textfraction{.1}

% Maximum fraction of a page that can be occupied by floats before a
% separate float page is produced:
%
\renewcommand\floatpagefraction{0.9}
\renewcommand\dblfloatpagefraction{.9}  % 2-column floats

% ------------------------------------------------------------------
%
% Since we're using two columns, lines are short and we can get away
% with less vertical space between lines, within lists and around
% various kinds of display.
%
% Normally, these parameters are set in the size option to the class
% file (standard definitions are in classes.dtx).  Here we want to
% accommodate 10pt, 11pt and 12pt, so we wrap the definitions in
% \ifcase.
%
```

```
460
    %  \normalsize
    %
    \ifcase\@ptsize%
        \renewcommand{\normalsize}{%                                    10pt
465         \@setsize\normalsize{11.3pt}\xpt\@xpt%
            \abovedisplayskip 10\p@\@plus2\p@\@minus5\p@%
            \abovedisplayshortskip\z@\@plus3\p@%
            \belowdisplayshortskip 4\p@\@plus3\p@\@minus3\p@%
            \belowdisplayskip\abovedisplayskip%
470         \let\@listi\@listI}%
    \or%
        \renewcommand{\normalsize}{%                                    11pt
            \@setsize\normalsize{12.6pt}\xipt\@xipt%
            \abovedisplayskip11\p@\@plus2\p@\@minus4\p@%
475         \abovedisplayshortskip\z@\@plus3\p@%
            \belowdisplayshortskip5\p@\@plus3\p@\@minus2\p@%
            \belowdisplayskip\abovedisplayskip%
            \let\@listi\@listI}%
    \or%
480     \renewcommand{\normalsize}{%                                    12pt
            \@setsize\normalsize{13pt}\xiipt\@xiipt%
            \abovedisplayskip 11\p@ \@plus3\p@ \@minus5\p@%
            \abovedisplayshortskip \z@ \@plus3\p@%
            \belowdisplayshortskip 5\p@ \@plus3\p@ \@minus2\p@%
485         \belowdisplayskip\abovedisplayskip%
            \let\@listi\@listI}%
    \fi

    % \small
490 %
    \ifcase\@ptsize%
        \renewcommand{\small}{%                                         10pt
            \@setsize\small{10.5pt}\ixpt\@ixpt%
            \abovedisplayskip 8\p@ \@plus3\p@ \@minus3\p@%
495         \abovedisplayshortskip \z@ \@plus2\p@%
            \belowdisplayshortskip 3\p@ \@plus2\p@ \@minus2\p@%
            \belowdisplayskip\abovedisplayskip%
            \def\@listi{\leftmargin\leftmargini%
                        \topsep 3.5\p@ \@plus1.5\p@ \@minus1.5\p@%
500                     \parsep 1.5\p@ \@plus\p@ \@minus\p@%
                        \itemsep \parsep}}%
    \or%
        \renewcommand{\small}{%                                         11pt
            \@setsize\small{11.3pt}\xpt\@xpt%
505         \abovedisplayskip 9\p@ \@plus2\p@ \@minus4\p@%
            \abovedisplayshortskip \z@ \@plus3\p@%
            \belowdisplayshortskip 5\p@ \@plus2.5\p@ \@minus2.5\p@%
            \belowdisplayskip\abovedisplayskip%
            \def\@listi{\leftmargin\leftmargini%
510                     \topsep 5\p@ \@plus2\p@ \@minus2\p@%
                        \parsep 2\p@ \@plus2\p@ \@minus\p@%
                        \itemsep \parsep}}%
    \or%
        \renewcommand{\small}{%                                         12pt
515         \@setsize\small{12pt}\xipt\@xipt%
            \abovedisplayskip 9\p@ \@plus3\p@ \@minus4\p@%
```

```
        \abovedisplayshortskip \z@ \@plus3\p@%
        \belowdisplayshortskip 5\p@ \@plus2.5\p@ \@minus2\p@%
        \belowdisplayskip\abovedisplayskip%
        \def\@listi{\leftmargin\leftmargini%
                    \topsep 5.5\p@ \@plus2.5\p@ \@minus2.5\p@%
                    \parsep 4\p@ \@plus1.5\p@ \@minus\p@%
                    \itemsep \parsep}}%
  \fi

  % \footnotesize
  %
  \ifcase\@ptsize
      \renewcommand{\footnotesize}{%                                  10pt
        \@setsize\footnotesize{9.3pt}\viiipt\@viiipt%
        \abovedisplayskip 5\p@ \@plus2\p@ \@minus3\p@%
        \abovedisplayshortskip \z@ \@plus\p@%
        \belowdisplayshortskip 2.5\p@\@plus\p@\@minus2\p@%
        \belowdisplayskip\abovedisplayskip%
        \def\@listi{\leftmargin\leftmargini%
                    \topsep 2.5\p@ \@plus\p@ \@minus\p@%
                    \parsep 1.5\p@ \@plus\p@ \@minus\p@%
                    \itemsep \parsep}}%
  \or%
      \renewcommand{\footnotesize}{%                                  11pt
        \@setsize\footnotesize{10.3pt}\ixpt\@ixpt%
        \abovedisplayskip 7\p@ \@plus2\p@ \@minus4\p@%
        \abovedisplayshortskip \z@ \@plus\p@%
        \belowdisplayshortskip 3\p@ \@plus2\p@ \@minus2\p@%
        \belowdisplayskip\abovedisplayskip%
        \def\@listi{\leftmargin\leftmargini%
                    \topsep 3\p@ \@plus2\p@ \@minus2\p@%
                    \parsep 2\p@ \@plus\p@ \@minus\p@%
                    \itemsep \parsep}}%
  \or%
      \renewcommand{\footnotesize}{%                                  12pt
        \@setsize\footnotesize{11pt}\xpt\@xpt%
        \abovedisplayskip 9\p@ \@plus2\p@ \@minus4\p@%
        \abovedisplayshortskip \z@ \@plus3\p@%
        \belowdisplayshortskip 5\p@ \@plus3\p@ \@minus3\p@%
        \belowdisplayskip\abovedisplayskip%
        \def\@listi{\leftmargin\leftmargini%
                    \topsep 4.5\p@ \@plus2\p@ \@minus2\p@%
                    \parsep 3\p@ \@plus\p@ \@minus\p@%
                    \itemsep \parsep}}%
  \fi


  % \large
  %
  \ifcase\@ptsize%
      \renewcommand{\large}{\@setsize\large{13pt}\xiipt\@xiipt}%  10pt
  \or%
      \renewcommand{\large}{\@setsize\large{13pt}\xiipt\@xiipt}%  11pt
  \or%
      \renewcommand{\large}{\@setsize\large{16pt}\xivpt\@xivpt}%  12pt
  \fi
```

```
% \Large
%
\ifcase\@ptsize%
    \renewcommand{\Large}{\@setsize\Large{16pt}\xivpt\@xivpt}%  10pt
 \or%
    \renewcommand{\Large}{\@setsize\Large{16pt}\xivpt\@xivpt}%  11pt
 \or%
    \renewcommand{\Large}{\@setsize\Large{16pt}\xivpt\@xivpt}%  12pt
 \fi

% Leave \scriptsize, \tiny, \huge, \Huge unchanged?

%
% Float separations, single and double-column
%
\ifcase\@ptsize%
    \setlength\floatsep{10\p@ \@plus 2\p@ \@minus 2\p@}%        10pt
    \setlength\textfloatsep{16\p@ \@plus 2\p@ \@minus 4\p@}%
    \setlength\intextsep{10\p@ \@plus 2\p@ \@minus 2\p@}%
    \setlength\dblfloatsep{10\p@ \@plus 2\p@ \@minus 2\p@}%
    \setlength\dbltextfloatsep{16\p@ \@plus 2\p@ \@minus 4\p@}%
 \or%
    \setlength\floatsep{10\p@ \@plus 2\p@ \@minus 2\p@}%        11pt
    \setlength\textfloatsep{16\p@ \@plus 2\p@ \@minus 4\p@}%
    \setlength\intextsep{10\p@ \@plus 2\p@ \@minus 2\p@}%
    \setlength\dblfloatsep{10\p@ \@plus 2\p@ \@minus 2\p@}%
    \setlength\dbltextfloatsep{16\p@ \@plus 2\p@ \@minus 4\p@}%
 \or%
    \setlength\floatsep{12\p@ \@plus 3\p@ \@minus 3\p@}%        12pt
    \setlength\textfloatsep{18\p@ \@plus 2\p@ \@minus 4\p@}%
    \setlength\intextsep{12\p@ \@plus 3\p@ \@minus 3\p@}%
    \setlength\dblfloatsep{12\p@ \@plus 2\p@ \@minus 4\p@}%
    \setlength\dbltextfloatsep{18\p@ \@plus 2\p@ \@minus 4\p@}%
 \fi

%
% Top-level list in \normalsize text
%
\ifcase\@ptsize%
    \def\@listi{\leftmargin\leftmargini%                        10pt
                \topsep   6\p@ \@plus2\p@ \@minus2\p@%
                \parsep   2\p@ \@plus0.5\p@ \@minus\p@%
                \itemsep 2.5\p@ \@plus\p@ \@minus0.5\p@}%
 \or%
    \def\@listi{\leftmargin\leftmargini%                        11pt
                \topsep   8\p@ \@plus2\p@ \@minus2\p@%
                \parsep   3\p@ \@plus1.5\p@ \@minus\p@%
                \itemsep  3\p@ \@plus1.5\p@ \@minus\p@}%
 \or%
    \def\@listi{\leftmargin\leftmargini%                        12pt
                \topsep   9\p@ \@plus3\p@    \@minus4\p@%
                \parsep   4\p@   \@plus2\p@ \@minus\p@%
                \itemsep  4\p@   \@plus2\p@ \@minus\p@}%
 \fi
\let\@listI\@listi
```

```
%
% Embedded lists unchanged.
%

% ---------------------------------------------------------------------
% End of colacl.sty
% ---------------------------------------------------------------------
```

21 doc/lightning-talk.md

```
---
title: Live-coding Audio in C
author: Claude Heiland-Allen
date: 2018-06-09
classoption: aspectratio=149
fontfamily: lato
fontfamilyoptions: default
fontsize: 14pt
header-includes:
  - \definecolor{mathr}{rgb}{0.5, 0.25, 0.25}
  - \usecolortheme[named=mathr]{structure}
---

## JACK Audio

- Process callback delegates to a function pointer

- Passes a pointer to a preallocated memory area

- Main thread can change the function pointer

## Dynamic Reloading

- `dlopen()`, `dlsym()`, `-ldl`

- Race condition: don't unload running code

- Double buffering: `dl` caches aggressively

## Detecting File Changes

- `inotify`

- if `go.c` changes: compile it (by calling `make`)

- if `go.so` changes: make a copy and load it

## Where

- performance tonight!

- <claude@mathr.co.uk>

- <https://mathr.co.uk/clive>
```

22 doc/Makefile

```
all: paper.pdf talk.pdf getting-started.pdf talk-thsf.pdf lightning-talk.pdf

clean:
	-rm paper.aux paper.bbl paper.blg paper.log paper.out

%.pdf: %.tex %.bib
	pdflatex $*
	bibtex   $*
	pdflatex $*
	pdflatex $*

%.pdf: %.md
	pandoc --slide-level 2 -t beamer $*.md -o $*.pdf

#paper.md.tex: paper.md
#	pandoc -f markdown+backtick_code_blocks+citations -t latex --listings --highlight-style=pygments -o paper.md.tex paper.md
```

23 doc/paper.bib

```
@inproceedings{PD,
author={Miller Puckette},
year={1996},
title={Pure Data},
booktitle={International Computer Music Conference},
pages={224-227},
address={San Francisco},
organisation={International Computer Music Association},
url={http://msp.ucsd.edu/Publications/icmc96.ps}
}

@article{SC3,
author={James McCartney},
year={2002},
title={Rethinking the Computer Music Language: SuperCollider},
journal={Computer Music Journal},
volume={26(4)},
pages={61-68}
}

@article{DL,
author={Adam J. Richter and Michael Kerrisk},
year={2015},
title={{dlclose}, {dlopen}, {dlmopen} - open and close a shared object},
journal={The Linux man-pages project},
url={http://man7.org/linux/man-pages/man3/dlopen.3.html}
}

@article{INOTIFY,
author={Michael Kerrisk and Heinrich Schuchardt},
year={2014},
title={{inotify} - monitoring filesytem events},
journal={The Linux man-pages project},
url={http://man7.org/linux/man-pages/man7/inotify.7.html}
```

```
}

@article{DAZFTZ,
author={Shawn Casey},
year={2008},
title={x87 and {SSE} Floating Point Assists in {IA-32}: Flush-To-Zero ({FTZ}) and Denormals-Are-Zero ({DAZ})},
journal={Intel Software Developer Zone},
url={https://software.intel.com/en-us/articles/x87-and-sse-floating-point-assists-in-ia-32-flush-to-zero-ftz-and-denormals-are-zero-daz/}
}

@article{NEBULULLABY,
author={{Various Artists}},
year={2016},
title={Nebulullaby: An Interstellar Cloud Of Dust (digital release)},
journal={Nebularosa},
url={http://nebularosa.net/nebulullaby/}
}

@article{BELA,
author={{Augmented Instruments Laboratory}},
year={2017},
title={Bela},
journal={{C4DM}, Queen Mary University of London},
url={https://bela.io/}
}

@article{FRAG,
author={Mikael Hvidtfeldt Christensen},
year={2013},
title={Fragmentarium},
journal={Syntopia},
url={https://syntopia.github.io/Fragmentarium/}
}
```

24 doc/paper.tex

```
\documentclass[11pt,a4paper]{article}
\usepackage{lac2018}
\sloppy
\newenvironment{contentsmall}{\small}

\usepackage{listings}
\lstset{breaklines=true}
\lstset{language=C}
\lstset{basicstyle=\small\ttfamily}
\lstset{extendedchars=true}
\lstset{tabsize=1}
\lstset{columns=fixed}
\lstset{showspaces=false}
\lstset{showstringspaces=true}
\lstset{numbers=left}
\lstset{xleftmargin=2em}
\lstset{stepnumber=1}
\lstset{numbersep=5pt}
\lstset{numberstyle=\footnotesize\ttfamily}
```

```
\lstset{captionpos=t}
\lstset{breaklines=true}
\lstset{breakatwhitespace=true}
\lstset{breakautoindent=true}
\lstset{linewidth=\columnwidth}

\usepackage[hidelinks]{hyperref}

\title{Live-coding Audio in C}

%see lac2018.sty for how to format multiple authors!
\author
{Claude HEILAND-ALLEN
\\ unaffiliated
\\ claude@mathr.co.uk
}

\begin{document}
\maketitle

\begin{abstract}
\begin{contentsmall}

  An audio live-coding skeleton is presented, implemented in C and adaptable to
  languages that expose the C ABI from shared libraries. The aim is to support
  a two-phase edit-commit coding cycle allowing long-lived signal processing
  graphs to be modified.

  The skeleton watches a directory for changes, recompiling changed sources and
  reloading changed shared objects. The shared object defines an audio
  processing callback. The skeleton maintains a memory region between reloads,
  allowing callback state to be preserved so that audio processing can continue
  uninterrupted.

\end{contentsmall}
\end{abstract}

\keywords{
\begin{contentsmall}
live-coding, audio processing, code reloading
\end{contentsmall}
}

\section{Motivation}\label{motivation}

(At least) two popular music software packages allow live-coding.
Pure-data~\cite{PD} allows the manipulation of audio processing graphs at ↙
    ↳ runtime,
but live-coding requires careful preparation as there is no edit/commit
cycle: every change is live as soon as it is made. Moreover each edit
recompiles the whole DSP chain, which is not a realtime-safe operation
and can lead to audio dropouts.

Supercollider~\cite{SC3} takes a
different approach, running selected code only when a certain key
combination is pressed. However, Supercollider DSP code runs on a server
distinct from its language, and synths (compiled by a client such as
```

```
\lstinline!sclang! and sent to the \lstinline!scsynth! server) can't
be modified once they are running, beyond setting control variables
or re-ordering synths within a group -- the code within each synth is
immutable.

The main goal of Clive is to be able to edit digital synthesis algorithms
while they are running, with a two phase edit/commit cycle: code can be
edited freely, until it is saved to disk, at which point it is re-compiled
and (if successful) re-loaded into an active audio processing engine.  The
reasons for using the C programming language are familiarity to the author,
precise control of memory layout of data structures allowing code reload
with state preservation, speed of compilation (relative to higher-level
languages like C++), and predictable runtime performance.

Performance with Clive typically involves pre-preparation, at least of
common unit generators like oscillators and filters, perhaps higher-level
effects like chorus or dynamic range compression, extending to more-complete
compositions with complex data flow.  The live-coding aspect involves editing
a file in the performer's favourite text editor, with the act of saving with
Ctrl-S or other shortcut being timed to allow the new code to start executing
after the latency of compilation.

\section{Implementation}\label{implementation}

\subsection{The skeleton}\label{the-skeleton}

The live engine starts a JACK client, whose audio processing function is
a thin wrapper around a \lstinline!callback! function pointer. The
function pointer is initially set to a function that fills its output
buffers with silence. The main program watches the current directory for
changes, recompiling changed sources and reloading a shared library
(named \lstinline!go.so!) on changes. The shared library defines a
symbol \lstinline!go! matching the \lstinline!callback_t! interface. The
callback is set to the address of this function. The callback is passed
a pointer to a memory area that is preserved between code reloads.

\begin{lstlisting}[language=C, caption={The skeleton}, label=lskeleton]
typedef void (*callback_t)(void *, float *, int, float *, int);
volatile callback_t callback = silence_callback;
void *data;
int process(jack_nframes_t n, void *arg) {
 float in[nin];
 float out[nout];
 // ... get JACK buffers
 for (int i = 0; i < n; ++i) {
  // ... copy buffers to in
  callback(data, in, nin, out, nout);
  // ... copy out to buffers
 }
 return 0;
}
int main() {
 data = calloc(1, bytes);
 load_library("go.so");
 // ... launch JACK client
 // the following is pseudo-code
 watch w = watch_directory(".");
```

```
 event e;
 while (( e = wait_event(w) )) {
  if (has_changed(e, "go.so"))
   load_library("go.so");
  if (has_changed(e, "go.c"))
   recompile("go.c");
 }
 return 0;
}
\end{lstlisting}

\subsection{Dynamically reload
libraries}\label{dynamically-reload-libraries}

The POSIX.1-2001 \lstinline!dl! interface is used to load shared libraries
at runtime~\cite{DL}. The library is loaded with \lstinline!dlopen!, the
callback function symbol address is found with \lstinline!dlsym!, and when
no longer needed the code can be unloaded with \lstinline!dlclose!. The type
cast gymnastics on line 9 are to avoid a warning in strict ISO C99 mode.

\begin{lstlisting}[language=C, caption={Using libdl}, label=llibdl]
void *old_library = 0;
void *new_library = 0;
void load_library(const char *name) {
 if (! name)
  return;
 if (! (new_library = dlopen(name, RTLD_NOW)))
  return;
 callback_t *new_callback;
 *(void **) (&new_callback) = dlsym(new_library, "go");
 if (new_callback) {
  set_callback(new_callback);
  // unload old library
  if (old_library)
   dlclose(old_library);
  old_library = new_library;
  new_library = 0;
 } else {
  dlclose(new_library);
  new_library = 0;
 }
}
\end{lstlisting}

\subsection{Mitigate races}\label{mitigate-races}

There is a race condition, especially problematic on multi-core
architectures: the old code must not be unloaded while the JACK callback
is still running it, otherwise the process will crash. To mitigate this
risk (though not prevent it entirely), a flag is added that makes the
main process spin until the current DSP block's JACK processing is
finished. The old library is probably safe to unload after
\lstinline!set_callback! has returned. To avoid a total stop to the live
performace in the unlikely event of this crash, a shell wrapper script
relaunches the engine automatically.

\begin{lstlisting}[language=C, caption={Race mitigation}, label=lrace]
```

```
volatile bool processing = false;
int process(jack_nframes_t n, void *arg) {
 processing = true;
 // ...
 processing = false;
 return 0;
}
void set_callback(callback_t new_callback) {
 while (processing)
  ;
 callback = new_callback;
}
\end{lstlisting}

\subsection{Defeat caches}\label{defeat-caches}

Unfortunately \lstinline!libdl! performs rather aggressive caching,
meaning that a library of the same name is not reloaded if it is already
open, even if its contents have changed.

A first try was to unload the old code and load the new code, hoping
that it would be quick enough to avoid audio dropouts (the callback
function pointer was set to silence before unloading to avoid a crash).
The dropouts were too severe to make this viable in a performance
situation.

A second attempt was to use two symlinks for double-buffering, but even
this does not defeat the cache. A real file copy appears to be
necessary, and finally the current version copies alternately to two
separate files in a double-buffering scheme.

\begin{lstlisting}[language=C, caption={Defeating caches by copying}, label=↙
    ↳ lcache]
const char *copy[2] =
 { "cp -f ./go.so ./go.a.so"
 , "cp -f ./go.so ./go.b.so" };
const char *library[2] =
 { "./go.a.so", "./go.b.so" };
int which = 0;
void copy_library() {
 if (! system(copy[which]))
  return library[which];
 else
  return 0;
}
void load_library(const char *name) {
 // ...
 if (new_callback) {
  which = 1 - which;
  // ...
 }
}
int main() {
 // ...
  if (has_changed(e, "go.so"))
   load_library(copy_library());
}
```

```
\end{lstlisting}

\subsection{Notify reloads}\label{notify-reloads}

It can be useful when live programming audio to perform initialization
once only. To that end, clive mandates that the first word of the memory
area is an \lstinline!int!, that is set to 1 when the code is reloaded.
The callback code is responsible for clearing it, necessary to detect
subsequent reloads.

\begin{lstlisting}[language=C, caption={Reload notification}, label=lreload]
volatile bool reloaded = false;
int process(jack_nframes_t n, void *arg) {
 if (reloaded) {
  int *p = data;
  *p = 1;
  reloaded = 0;
 }
 // ...
 return 0;
}
void set_callback(callback_t new_callback) {
 // ...
 reloaded = true;
}
\end{lstlisting}

\subsection{Watch for changes}\label{watch-for-changes}

On Linux, the filesystem can be monitored for changes using
\lstinline!inotify!~\cite{INOTIFY}. The function
\lstinline!inotify_add_watch! is used to check for file close events, of
files that were opened for writing. The \lstinline!inotify_event!
structure contains the name of the file that changed, which is compared
to two cases: if the source \lstinline!go.c! has changed, recompile it;
if the library \lstinline!go.so! has changed, reload it. Hopefully the
first case triggers the second case, otherwise the live coder should
check the terminal for compilation error messages.

It is important to start watching before the main loop, rather than
stopping and starting repeatedly within the loop, lest events are missed
between loop iterations. To support languages other than C, it is
necessary to modify line 27 of Listing \ref{linotify} to reference the
new source file, and add the approriate compilation commands to the
\lstinline!Makefile!.

\begin{lstlisting}[language=C, caption={Using inotify}, label=linotify]
int main() {
 // ...
 int ino = inotify_init();
 if (ino == -1) return 1;
 int wd = inotify_add_watch(ino, ".", IN_CLOSE_WRITE);
 if (wd == -1) return 1;
 ssize_t buf_bytes = sizeof(struct inotify_event) + NAME_MAX + 1;
 char *buf = malloc(buf_bytes);
 while (true) {
  load_library(copy_library());
```

```
  // read events (blocking)
  bool library_changed = false;
  while (! library_changed) {
   memset(buf, 0, buf_bytes);
   ssize_t r = read(ino, buf, buf_bytes);
   if (r == -1)
    sleep(1);
   else {
    char *bufp = buf;
    while (bufp < buf + r) {
     struct inotify_event *ev = (struct inotify_event *) bufp;
     bufp += sizeof(struct inotify_event) + ev->len;
     if (ev->mask & IN_CLOSE_WRITE) {
      if (0 == strcmp("go.so", ev->name))
       // reload
       library_changed = true;
      if (0 == strcmp("go.c", ev->name))
       // recompile
       system("make --quiet");
     }
    }
   }
  }
  close(ino);
  return 0;
}
\end{lstlisting}

\subsection{Squash denormals}\label{squash-denormals}

A common problem in audio processing on \lstinline!x86_64! architectures
is the large increased cost of processing denormal floating point
numbers when recursive algorithms decay to zero. This is achieved in
clive by setting the \lstinline!DAZ! (denormals-are-zero) and
\lstinline!FTZ! (flush-to-zero) CPU flags~\cite{DAZFTZ}, and ensuring
that the shared library code is compiled with the \lstinline!gcc! flag
\lstinline!-mfpmath=sse! which makes calculations use the SSE unit.

\begin{lstlisting}[language=C, caption={Setting DAZ and FTZ}, label=ldenormal]
int process(jack_nframes_t n, void *arg) {
#ifdef __x86_64__
 fenv_t fe;
 fegetenv(&fe);
 unsigned int old_mxcsr = fe.__mxcsr;
 fe.__mxcsr |= 0x8040;
 fesetenv(&fe);
#endif
 // ...
#ifdef __x86_64__
 fe.__mxcsr = old_mxcsr;
 fesetenv(&fe);
#endif
 return 0;
}
\end{lstlisting}
```

```
\section{Example}\label{example}

Thus far in this paper all the code presented has been from the live
engine. To remedy this, here is a short callback example, presenting
a simple metronome tone:

\begin{lstlisting}[language=C, caption={Metronome example}, label=lexample]
#include <math.h>
#define SR 48000
#define PI 3.141592653589793
double wrap(double x) {
  return x - floor(x);
}
typedef struct { double phase; } PHASOR;
double phasor(PHASOR *s, double hz) {
  return s->phase = wrap(s->phase + hz / SR);
}
typedef struct {
  int reloaded;
  PHASOR beat;
  PHASOR tone;
} S;
extern void go(S *s, float *in, int nin, float *out, int nout) {
  double beat = phasor(&s->beat, 120 / 60.0) < 0.25;
  double tone = sin(2 * PI * phasor(&s->tone, 1000));
  double metronome = beat * tone;
  for (int c = 0; c < nout; ++c)
    out[c] = metronome;
}
\end{lstlisting}

\section{Evaluation and future work}\label{evaluation}

Clive's code swapping is quantized to JACK process callback block boundaries,
which is not musically useful. A workaround can be to scatter the callback
code with sample-and-hold unit generators so that the previous values are kept
until some future trigger (from an already-running clock oscillator perhaps)
synchronizes them with the desired sound. The need in Clive to specify unit
generator storage separately from the invocation of the generator makes this
doubly painful.

Clive processes one sample at a time, which is not necessarily the most
efficient way to process audio. It may be useful to add unit generators to
support reblocking, alongside vector operations, to Clive's library. This
reblocking would be necessary if FFT operations were to be supported,
which would also benefit from reblocking with overlap.

The Bela~\cite{BELA} platform is interesting for low-latency audio processing.
Clive could be adapted to cross-compile into a device directory mounted on
the host, with the engine split into two parts (device to reload library
code, and host to recompile source code). Bela also supports electronic
connections (both analogue and digital pins), which would be nice to expose
to Clive callbacks.

The OpenGL graphics library is interesting for programmable shaders. Clive
could be adapted to load and compile GLSL source code for live-coding visuals,
perhaps focusing on fragment shaders similar to Fragmentarium~\cite{FRAG}.
```

```
A way for Clive's audio processing callbacks to send control data to OpenGL
visualisations would allow tight audio-visual synchronisation.

\section{Conclusion}\label{conclusion}

A live coding system for audio processing in C has been presented. The
author has performed with it many times since its first version in 2011,
including both seated events and club-style Algorave situations, as well
as sessions at home.

The code and some session audio+diff-casts can be found at
\begin{itemize}
\item \url{https://code.mathr.co.uk/clive}
\item \url{https://mathr.co.uk/clive}
\end{itemize}
respectively.

\bibliographystyle{acl}
\bibliography{paper}

\end{document}
```

25 doc/README.md

```
# note

these documents describe clive as of the v0 tag

significant changes have been made since then, these docs need updating
```

26 doc/talk.md

```
---
title: Live-coding Audio in C
author: Claude Heiland-Allen
date: 2018-09-02
classoption: aspectratio=149
fontfamily: lato
fontfamilyoptions: default
fontsize: 14pt
header-includes:
  - \definecolor{mathr}{rgb}{0.5, 0.25, 0.25}
  - \usecolortheme[named=mathr]{structure}
---

## Contents

- Overview (principle, strengths, weaknesses)

- JACK Audio (client, process callback)

- Dynamic Reloading (races, caching)

- Detecting File Changes (inotify events)
```

```
- Future Work

# Overview

## How Clive Works

- watches a directory for file changes

- recompiles changed sources

- reloads recompiled library

- maintains memory area between reloads

## Clive Strengths

- two-phase edit/commit cycle

- no explicit "DSP graph of UGens" separation

- compilation is realtime safe

- reloading is realtime safe (almost)

- processing uses single-sample callbacks (simple)

## Clive Weaknesses

- explicit state maintenance, append-only

- tiny UGen library

- compilation is high latency

- reloading is at JACK block boundaries

- processing uses single-sample callbacks (slow)

# JACK Audio

## JACK Client Setup

\fontsize{12pt}{13}\selectfont

```C
client = jack_client_open("live", JackNoStartServer, 0);
jack_set_process_callback(client, processcb, 0);
out_port = jack_port_register(client, "live:output_1",
      JACK_DEFAULT_AUDIO_TYPE, JackPortIsOutput, 0);
jack_activate(client);
jack_connect(client,
      jack_port_name(out_port), "system:playback_1");
```

JACK Process Callback

\fontsize{12pt}{13}\selectfont
```

```C
int processcb(jack_nframes_t nframes, void *arg) {
  jack_default_audio_sample_t *in, *out;
  in  = jack_port_get_buffer(in_port,  nframes);
  out = jack_port_get_buffer(out_port, nframes);
  for (jack_nframes_t i = 0; i < nframes; ++i)
    out[i] = state.func(state.data, in[i]);
  return 0;
}
```

Clive Concept

- change `state.func` at runtime

- preserve contents of `state.data`

Dynamic Code Reloading

Using `libdl`

\fontsize{12pt}{13}\selectfont

```C
void *old_dl = 0;
void *new_dl = 0;
if ((new_dl = dlopen("go.so", RTLD_NOW))) {
  callback *new_cb;
  new_cb = dlsym(new_dl, "go");
  if (new_cb) {
    while (inprocesscb) ; // race condition
    state.func = new_cb;
    state.reload = 1;
    while (inprocesscb) ; // race condition
    if (old_dl) dlclose(old_dl);
    old_dl = new_dl;
  } else dlclose(new_dl);
}
```

Race Condition

- don't unload running code (otherwise... boom!)

```C
volatile int inprocesscb = 0;
int processcb(/* ... */) {
  inprocesscb = 1;
  // ...
  inprocesscb = 0;
  return 0;
}
```

Cache Circumvention

```
- `dlopen` caches based on filenames (or inodes?)

- need to **copy** the `go.so` file to a new location

- double buffering works (two copies are enough)

# Detecting File Changes

## Getting INotify Events

\fontsize{12pt}{13}\selectfont

```C
int ino = inotify_init();
int wd = inotify_add_watch(ino, ".", IN_CLOSE_WRITE);
ssize_t buf_bytes =
  sizeof(struct inotify_event) + NAME_MAX + 1;
char *buf = malloc(buf_bytes);
while (1) {
  memset(buf, 0, buf_bytes);
  ssize_t r = read(ino, buf, buf_bytes); // blocking
  if (r == -1) sleep(1);
  else /* parse events */ ;
}
```

Parsing INotify Events

\fontsize{12pt}{13}\selectfont

```C
char *bufp = buf;
while (bufp < buf + r) {
  struct inotify_event *ev = bufp;
  bufp += sizeof(*ev) + ev->len;
  if (ev->mask & IN_CLOSE_WRITE) {
    if (0 == strcmp("go.so", ev->name))
      /* reload */ ;
    if (0 == strcmp("go.c", ev->name))
      /* recompile */ ;
  }
}
```

Future Work

Future Work

- embiggen UGen library

- block-based processing (including FFT)

- low-latency embedded DSP

    (cross-compile on host, run on device)

Thanks
```

```
195
    ## Clive

    - <claude@mathr.co.uk>

200 - <https://mathr.co.uk/clive>
```

27 doc/talk-thsf.md

```
    ---
    title: Live-coding Audio in C
    author: Claude Heiland-Allen
    date: 2018-05-12
  5 classoption: aspectratio=149
    fontfamily: lato
    fontfamilyoptions: default
    fontsize: 14pt
    header-includes:
 10   - \definecolor{mathr}{rgb}{0.5, 0.25, 0.25}
      - \usecolortheme[named=mathr]{structure}
    ---

    ## Contents
 15
    - Overview

    - JACK Audio

 20 - Dynamic Reloading

    - Detecting File Changes

    - Future Work
 25
    # Overview

    ## How Clive Works

 30 - watches a directory for file changes

    - recompiles changed sources

    - reloads recompiled library
 35
    - maintains memory area between reloads

    ## Clive Strengths

 40 - two-phase edit/commit cycle

    - no explicit "DSP graph of UGens" separation

    - compilation is realtime safe
 45
    - reloading is (almost) realtime safe

    - processing uses single-sample callbacks (simple)
```

```
50  ## Clive Weaknesses

    - explicit state maintenance, append-only

    - tiny UGen library
55
    - compilation is high latency

    - reloading is at JACK block boundaries

60  - processing uses single-sample callbacks (slow)

    # JACK Audio

    ## JACK Client Setup
65
    \fontsize{12pt}{13}\selectfont

    ```C
    client = jack_client_open("live", JackNoStartServer, 0);
70  jack_set_process_callback(client, processcb, 0);
    out_port = jack_port_register(client, "live:output_1",
          JACK_DEFAULT_AUDIO_TYPE, JackPortIsOutput, 0);
    jack_activate(client);
    jack_connect(client,
75        jack_port_name(out_port), "system:playback_1");
    ```

    ## JACK Process Callback

80 \fontsize{12pt}{13}\selectfont

    ```C
    int processcb(jack_nframes_t nframes, void *arg) {
      jack_default_audio_sample_t *in, *out;
85    in  = jack_port_get_buffer(in_port,  nframes);
      out = jack_port_get_buffer(out_port, nframes);
      for (jack_nframes_t i = 0; i < nframes; ++i)
        out[i] = state.func(state.data, in[i]);
      return 0;
90  }
    ```

    ## Clive Concept

95 - change `state.func` at runtime

    - preserve contents of `state.data`

    # Dynamic Code Reloading
100
    ## Using `libdl`

    \fontsize{12pt}{13}\selectfont

105 ```C
```

```
void *old_dl = 0;
void *new_dl = 0;
if ((new_dl = dlopen("go.so", RTLD_NOW))) {
  callback *new_cb;
  new_cb = dlsym(new_dl, "go");
  if (new_cb) {
    while (inprocesscb) ; // race condition
    state.func = new_cb;
    state.reload = 1;
    if (old_dl) dlclose(old_dl);
    old_dl = new_dl;
  } else dlclose(new_dl);
}
```

Race Condition

- don't unload running code (otherwise... boom!)

```C
volatile inprocesscb = 0;
int processcb(/* ... */) {
  inprocesscb = 1;
  // ...
  inprocesscb = 0;
  return 0;
}
```

Cache Circumvention

- `dlopen` caches based on filenames

- need to copy the `go.so` file to a new location

- double buffering works (two copies are enough)

Detecting File Changes

Getting INotify Events

\fontsize{12pt}{13}\selectfont

```C
int ino = inotify_init();
int wd = inotify_add_watch(ino, ".", IN_CLOSE_WRITE);
ssize_t buf_bytes =
  sizeof(struct inotify_event) + NAME_MAX + 1;
char *buf = malloc(buf_bytes);
while (1) {
  memset(buf, 0, buf_bytes);
  ssize_t r = read(ino, buf, buf_bytes); // blocking
  if (r == -1) sleep(1);
  else /* parse events */ ;
}
```

```
## Parsing INotify Events

\fontsize{12pt}{13}\selectfont

```C
char *bufp = buf;
while (bufp < buf + r) {
  struct inotify_event *ev = bufp;
  bufp += sizeof(*ev) + ev->len;
  if (ev->mask & IN_CLOSE_WRITE) {
    if (0 == strcmp("go.so", ev->name))
      /* reload */ ;
    if (0 == strcmp("go.c", ev->name))
      /* recompile */ ;
  }
}
```

Future Work

Embiggen UGen Library

- oscillators (bandlimited wavetables, ...)

- filters (Butterworth, ...)

- spatialisation (ambisonics, ...)

- sequencing (varistep, ...)

Block-based Processing

- more CPU efficient

- more awkward

- FFT-based spectral processing

Embedded DSP

- low power audio processing devices

- process external inputs with very low latency

- compile on host and transfer over USB/network

- also support sensors/electronics

- rapid prototyping of embedded instruments

Thanks

Clive

- <claude@mathr.co.uk>

- <https://mathr.co.uk/clive>
```

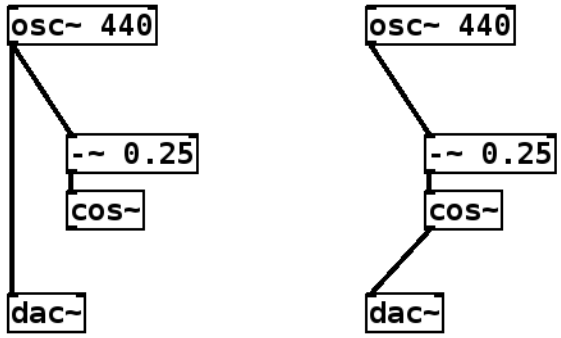
220

- <https://code.mathr.co.uk/clive>

28 doc/two-phase-pd-1.pd

```
#N canvas 3 58 680 503 10;
#X obj 64 180 dac~;
#X obj 64 68 osc~ 440;
#X obj 87 118 -~ 0.25;
#X obj 87 140 cos~;
#X obj 204 180 dac~;
#X obj 204 68 osc~ 440;
#X obj 227 118 -~ 0.25;
#X obj 227 140 cos~;
#X connect 1 0 0 0;
#X connect 1 0 2 0;
#X connect 2 0 3 0;
#X connect 5 0 6 0;
#X connect 6 0 7 0;
#X connect 7 0 4 0;
```

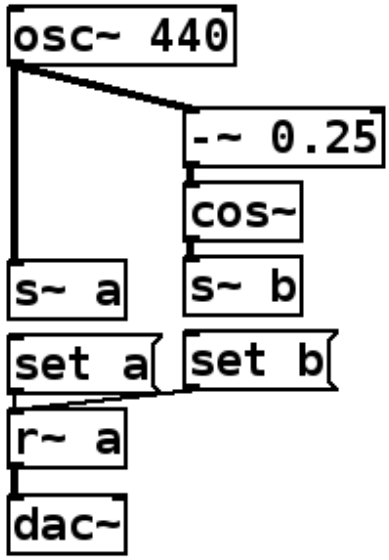
29 doc/two-phase-pd-1.png



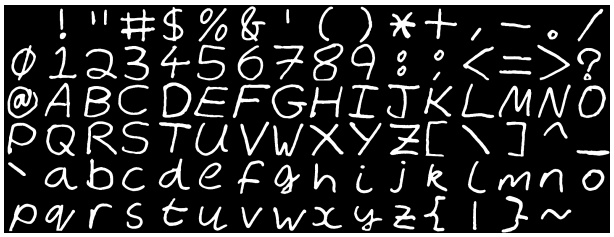
30 doc/two-phase-pd-2.pd

```
#N canvas 3 58 680 503 10;
#X obj 64 193 dac~;
#X obj 64 68 osc~ 440;
#X obj 109 94 -~ 0.25;
#X obj 109 113 cos~;
#X obj 64 133 s~ a;
#X obj 109 132 s~ b;
#X obj 64 171 r~ a;
#X msg 109 151 set b;
#X msg 64 152 set a;
#X connect 1 0 2 0;
#X connect 1 0 4 0;
#X connect 2 0 3 0;
#X connect 3 0 5 0;
#X connect 6 0 0 0;
#X connect 7 0 6 0;
#X connect 8 0 6 0;
```

31 doc/two-phase-pd-2.png



32 extra/ascii.png



33 extra/diffc2html.py

```
from pygments import highlight
from pygments.lexers.compiled import CLexer
from pygments.formatters import HtmlFormatter

print "<pre>"
try:
  while True:
    line = raw_input()
    if (line[:4] == 'diff') or (line[:5] == 'index') or (line[:3] == '---') or (line[:3] == '+++'):
      print ('<span class="dh">' + line + '</span>')
    elif line[:2] == '@@':
      ix = line[2:].index('@') + 4
      print ('<span class="dl">' + line[:ix] + highlight(line[ix:], CLexer(), HtmlFormatter(nowrap=True))[:-1] + '</span>')
    elif line[:1] == ' ':
      print ('<span class="du"> ' + highlight(line[1:], CLexer(), HtmlFormatter(nowrap=True))[:-1] + '</span>')
```

```
    elif line[:1] == '+':
      print ('<span class="di">+' + highlight(line[1:], CLexer(), HtmlFormatter(↙
          ↳ nowrap=True))[:-1] + '</span>')
    elif line[:1] == '-':
      print ('<span class="dd">-' + highlight(line[1:], CLexer(), HtmlFormatter(↙
          ↳ nowrap=True))[:-1] + '</span>')
    else:
      print ('<span class="dw">' + line + '</span>')
except EOFError:
  pass
print "</pre>"
```

34 extra/git-term-log.sh

```
#!/bin/sh
git log --pretty=format:"%H" --reverse --after 2012-01-01 |
while read commit
do
  git show --color --word-diff=color --pretty=format:"%n%Cblue%H%Creset %C(↙
      ↳ yellow)%ci%Creset%n" $commit
done |
most
```

35 extra/Makefile

```
GCC ?= gcc-8

visuals: visuals.c ../client/dsp.h ../client/dsp/*.h ascii.pgm
	$(GCC) -xc++ -std=c++11 -Wall -pedantic -Wextra -Wno-unused-parameter -↙
	    ↳ O3 -march=native -I../client/ -o visuals visuals.c -lGL -lGLEW -↙
	    ↳ lglfw -lv4l2 `pkg-config --cflags --libs jack` -lm -ljpeg

ascii.pgm: ascii.png
	pngtopnm < ascii.png > ascii.pgm
```

36 extra/session2html.sh

```
#!/bin/sh
if [ "x${1}" = "x" ]
then
  echo "usage: ${0} session-branch [session-start]"
  exit 1
fi
branch="${1}"
date="$(echo "${2:-${1}}" | sed "s/session-\(....-..-..\)-\(..\)\(..\)\(..\)/\1 ↙
    ↳ \2:\3:\4/")"
tmp="$(mktemp -p '' -d session2html.XXXXXXXXXX)"
(git log --pretty=format:"%H" --reverse --after "${date}" "${branch}" && echo) >↙
    ↳ "${tmp}/commits.txt"
cat "${tmp}/commits.txt" |
while read commit
do
  echo "<div class='commit' id='commit-$commit'>" &&
  git show --pretty=format:"diff @ %ci" "$commit" |
  python extra/diffc2html.py &&
  echo "</div>"
```

```
done > "${tmp}/diffs.html" &&
cat "${tmp}/commits.txt" |
(
  osecs="$(date -d "${date}" +%s)"
  while read commit
  do
    secs="$(date -d "$(git show --pretty=%ci $commit | head -n 1)" +%s)"
    delta="$((secs - osecs))"
    echo "${delta}"
  done
) > "${tmp}/times.txt"
(
  cat << EOF &&
<!DOCTYPE html>
<html>
<head>
<title>${branch}</title>
<style type='text/css'>
body {
  text-align: center;
  background-color: grey;
}
h1 {
  background-color: white;
  width: 40ex;
  margin: 1em auto;
  padding: 0.25em 1em;
  border 1px solid;
}
audio { width: 100%; }
h2 { text-align: center; }
p { text-align: justify; }
.commit {
  text-align: left;
  background-color: white;
  width: 80ex;
  margin: 2em auto;
  padding: 0 1em;
  border: 1px solid;
  display: none;
}
.commit .k, .commit .kr, .commit .kt, .commit .p { font-weight: bold; }
.commit .c1 { font-style: italic; }
.commit .dh:first-child { font-weight: bold; color: black; }
.commit .dh { width: 32ex; display: inline-block; float: right; color: silver; }
.commit .dl { color: blue; }
.commit .du { color: grey; }
.commit .di { color: green; }
.commit .dd { color: red; }
</style>
<script type="text/javascript">
var times;
var commits;
function search(t) {
  var lo = 0;
  var hi = times.length - 1;
  while (lo + 1 < hi) {
```

```
      var mi = Math.floor((lo + hi) / 2);
      if (times[mi] <= t) {
        lo = mi;
      } else {
        hi = mi;
      }
    }
    return commits[lo];
}
function go() {
    var current = null;
    var player = document.getElementById("player");
    player.ontimeupdate =
      function() {
        var commit = document.getElementById("commit-" + search(player.currentTime↙
            ↳ ));
        if (!!current) {
          current.style.display = "none";
        }
        if (!!commit) {
          commit.style.display = "block";
        }
        current = commit;
      };
}
times = [
0,
EOF
    cat "${tmp}/times.txt" |
    sed 's/$/,/'
    cat << EOF
1000000];
commits = [
"intro",
EOF
    cat "${tmp}/commits.txt" |
    sed 's/^/"/' |
    sed 's/$/",/'
    cat << EOF
"outro"];
</script>
</head>
<body onload="javascript:go()">
<h1>${branch}</h1>
<audio id="player" autoplay controls>
<source src="${branch}.ogg" type="audio/ogg; codecs=vorbis">
</audio>
<div class='commit' id='commit-intro'>
<h2><a href="https://code.mathr.co.uk/clive" title="clive" target="_blank">clive↙
    ↳ </a></h2>
<p>a C audio live-coding skeleton</p>
</div>
EOF
    cat "${tmp}/diffs.html" &&
    cat << EOF
<div class='commit' id='commit-outro'>
<h2>the end</h2>
```

```
</div>
</body>
</html>
EOF
) > "${branch}.html"
```

37 extra/visuals.c

```
#ifndef _GNU_SOURCE
#define _GNU_SOURCE
#endif

#include <assert.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#include <sys/types.h>
#include <sys/time.h>
#include <sys/mman.h>
#include <fcntl.h>
#include <libv4l2.h>
#include <linux/videodev2.h>
#include <setjmp.h>
#include <jpeglib.h>

#include <GL/glew.h>
#include <GLFW/glfw3.h>
#define GLM_FORCE_RADIANS
#include <glm/gtc/matrix_transform.hpp>

#include <jack/jack.h>

using std::isnan;
using std::isinf;
typedef float sample;
#define SR 48000
#include "dsp.h"

jack_client_t *j_client = 0;
jack_port_t *j_in[2] = { 0, 0 };

float hue_shift[3] = { 1, 1, 1 };

unsigned char *webcam_buffer = 0;
unsigned char **webcam_buffer_ptrs = 0;
struct webcam_buffer {
  void *start;
  size_t length;
};
struct webcam_buffer *webcam_buffers = 0;

GLint u_text_size = -1;
GLint u_hue_shift = -1;
GLint u_hue_rotate = -1;
GLint u_hue_unrotate = -1;
GLint u_fly_size = -1;
```

```
50
    const char *text_vert =
    "#version 330 core\n"
    "uniform vec2 screen_size;\n"
    "in vec2 pos;\n"
55  "in vec2 tc;\n"
    "out vec2 coord;\n"
    "void main() {\n"
    "  gl_Position = vec4(pos, 0.0, 1.0);\n"
    "  coord = screen_size * tc;\n"
60  "}\n"
    ;

    const char *text_frag =
    "#version 330 core\n"
65  "uniform sampler2D webcam;\n"
    "uniform sampler2D text;\n"
    "uniform sampler2D ascii;\n"
    "uniform vec2 webcam_size;\n"
    "uniform vec2 screen_size;\n"
70  "uniform vec2 text_size;\n"
    "uniform vec3 hue_shift;\n"
    "uniform mat3 hue_rotate;\n"
    "uniform mat3 hue_unrotate;\n"
    "uniform float fly_size;\n"
75  "const float fly_factor = 2.0;\n"
    "in vec2 coord;\n"
    "out vec4 colour;\n"
    "void main() {\n"
    "  float screen_aspect = screen_size.x / screen_size.y;\n"
80  "  float text_aspect = text_size.x / text_size.y;\n"
    "  float webcam_aspect = webcam_size.x / webcam_size.y;\n"
    "  vec2 webcam_coord = coord / screen_size;\n"
    "  webcam_coord -= vec2(0.5);\n"
    "  if (webcam_aspect < screen_aspect) {\n"
85  "    webcam_coord.y *= webcam_aspect / screen_aspect;\n"
    "  } else {\n"
    "    webcam_coord.x /= webcam_aspect / screen_aspect;\n"
    "  }\n"
    "  vec2 webcam_coord1 = webcam_coord;\n"
90  "  float r = sqrt(3.0)/2.0 * (4.0 / 3.0) / (16.0 / 9.0);\n"
    "  webcam_coord.y *= r;\n"
    "  webcam_coord1.y *= r;\n"
    "  vec2 middle = round(fly_size * vec2(webcam_coord.x + webcam_coord.y / 2.0, ↙
     ↳ webcam_coord.y));\n"
    "  middle.x -= middle.y / 2.0;\n"
95  "  vec2 middle0 = middle;\n"
    "  vec2 middle1 = middle + vec2(1.0, 0.0);\n"
    "  vec2 middle2 = middle + vec2(-1.0, 0.0);\n"
    "  vec2 middle3 = middle + vec2(-0.5, 1.0);\n"
    "  vec2 middle4 = middle + vec2(0.5, 1.0);\n"
100 "  vec2 middle5 = middle + vec2(-0.5, -1.0);\n"
    "  vec2 middle6 = middle + vec2(0.5, -1.0);\n"
    "  middle0 /= fly_size;\n"
    "  middle1 /= fly_size;\n"
    "  middle2 /= fly_size;\n"
105 "  middle3 /= fly_size;\n"
```

```
"  middle4 /= fly_size;\n"
"  middle5 /= fly_size;\n"
"  middle6 /= fly_size;\n"
"  float d = 1.0/0.0;\n"
"  if (distance(webcam_coord, middle0) < d) {\n"
"    d = distance(webcam_coord, middle0);\n"
"    middle = middle0;\n"
"  }\n"
"  if (distance(webcam_coord, middle1) < d) {\n"
"    d = distance(webcam_coord, middle1);\n"
"    middle = middle1;\n"
"  }\n"
"  if (distance(webcam_coord, middle2) < d) {\n"
"    d = distance(webcam_coord, middle2);\n"
"    middle = middle2;\n"
"  }\n"
"  if (distance(webcam_coord, middle3) < d) {\n"
"    d = distance(webcam_coord, middle3);\n"
"    middle = middle3;\n"
"  }\n"
"  if (distance(webcam_coord, middle4) < d) {\n"
"    d = distance(webcam_coord, middle4);\n"
"    middle = middle4;\n"
"  }\n"
"  if (distance(webcam_coord, middle5) < d) {\n"
"    d = distance(webcam_coord, middle5);\n"
"    middle = middle5;\n"
"  }\n"
"  if (distance(webcam_coord, middle6) < d) {\n"
"    d = distance(webcam_coord, middle6);\n"
"    middle = middle6;\n"
"  }\n"
"  ivec2 imiddle = ivec2(floor(fly_size * middle * 2.0));\n"
"  int icell = int(floor(float(imiddle.x) / 2.0)) + ((1 + int(floor(float(↙
    ↳ imiddle.y) / 2.0))) & 1);\n"
"  icell += 1000;\n"
"  icell = int((float(icell)/3.0 - floor(float(icell)/3.0)) * 3.0) % 3;\n"
"  vec3 cell_mask = vec3(1.0);\n"
"  cell_mask[icell] = 0.5;\n"
"  webcam_coord = webcam_coord + fly_factor * (webcam_coord - middle);\n"
"  webcam_coord1 = webcam_coord1 + fly_factor * webcam_coord1;\n"
"  webcam_coord.y /= r;\n"
"  webcam_coord1.y /= r;\n"
"  webcam_coord += vec2(0.5);\n"
"  webcam_coord1 += vec2(0.5);\n"
"  vec3 webcam_colour = textureGrad(webcam, webcam_coord, dFdx(webcam_coord1), ↙
    ↳ dFdy(webcam_coord1)).rgb;\n"
"  float scale;\n"
"  vec2 offset = vec2(0.0);\n"
"  if (text_aspect < screen_aspect) {\n"
"    scale = text_size.y / screen_size.y;\n"
"    offset = vec2((scale * screen_size.x - text_size.x) / 2.0, 0.0);\n"
"  } else {\n"
"    scale = text_size.x / screen_size.x;\n"
"    offset = vec2(0.0, (scale * screen_size.y - text_size.y) / 2.0);\n"
"  }\n"
"  vec2 index = scale * coord - offset;\n"
```

```
    "  ivec2 ix = ivec2(int(floor(index.x)), int(floor(index.y)));\n"
    "  vec2 index1 = index;\n"
    "  index -= vec2(ix);\n"
    "  int glyph      = max(0, int(round(255.0 * texelFetch(text, ix, 0).r)) - 32);\
    ↪ n"
165 "  int line_glyph = max(0, int(round(255.0 * texelFetch(text, ivec2(1, ix.y), 0)
    ↪ .r)) - 32);\n"
    "  vec2 glyph_coord = vec2(float(glyph % 16), float(glyph / 16));\n"
    "  glyph_coord += index;\n"
    "  glyph_coord *= vec2(1.0 / 16.0, 1.0 / 6.0);\n"
    "  vec2 glyph_coord1 = index1 * vec2(1.0 / 16.0, 1.0 / 6.0);\n"
170 "  vec3 base = vec3(0.25);\n"
    "  if (line_glyph == 0) { base = vec3(1.0); } else\n"
    "  if (line_glyph == 32) { base = vec3(0.25, 0.25, 1.0); } else\n"
    "  if (line_glyph == 11) { base = vec3(0.25, 1.0, 0.25); } else\n"
    "  if (line_glyph == 13) { base = vec3(1.0, 0.25, 0.25); }\n"
175 "  webcam_colour = hue_rotate * webcam_colour;\n"
    "  vec3 webcam_colour1 = 0.5 * (1.0 - cos(hue_shift * 3.141592653 *
    ↪ webcam_colour));\n"
    "  webcam_colour = mix(webcam_colour, webcam_colour1, cell_mask);\n"
    "  webcam_colour = hue_unrotate * webcam_colour;\n"
    "  webcam_colour = clamp(webcam_colour, 0.0, 1.0);\n"
180 "  vec3 glyph_colour = base *\n"
    "    textureGrad(ascii, glyph_coord, dFdx(glyph_coord1), dFdy(glyph_coord1)).r;\
    ↪ n"
    "  colour = vec4(2.0 * glyph_colour + webcam_colour, 2.0) * 0.5;\n"
    "}\n"
    ;
185
  void debug_program(GLuint program, const char *name) {
    if (program) {
      GLint linked = GL_FALSE;
      glGetProgramiv(program, GL_LINK_STATUS, &linked);
190   if (linked != GL_TRUE) {
        fprintf(stderr, "%s: OpenGL shader program link failed\n", name);
      }
      GLint length = 0;
      glGetProgramiv(program, GL_INFO_LOG_LENGTH, &length);
195   char *buffer = (char *) malloc(length + 1);
      glGetProgramInfoLog(program, length, 0, buffer);
      buffer[length] = 0;
      if (buffer[0]) {
        fprintf(stderr, "%s: OpenGL shader program info log\n", name);
200     fprintf(stderr, "%s\n", buffer);
      }
      free(buffer);
    } else {
      fprintf(stderr, "%s: OpenGL shader program creation failed\n", name);
205 }
  }

  void debug_shader(GLuint shader, GLenum type, const char *name) {
    const char *tname = 0;
210 switch (type) {
      case GL_VERTEX_SHADER:   tname = "vertex";   break;
      case GL_FRAGMENT_SHADER: tname = "fragment"; break;
      default:                 tname = "unknown";  break;
```

```
  }
  if (shader) {
    GLint compiled = GL_FALSE;
    glGetShaderiv(shader, GL_COMPILE_STATUS, &compiled);
    if (compiled != GL_TRUE) {
      fprintf(stderr, "%s: OpenGL %s shader compile failed\n", name, tname);
    }
    GLint length = 0;
    glGetShaderiv(shader, GL_INFO_LOG_LENGTH, &length);
    char *buffer = (char *) malloc(length + 1);
    glGetShaderInfoLog(shader, length, 0, buffer);
    buffer[length] = 0;
    if (buffer[0]) {
      fprintf(stderr, "%s: OpenGL %s shader info log\n", name, tname);
      fprintf(stderr, "%s\n", buffer);
    }
    free(buffer);
  } else {
    fprintf(stderr, "%s: OpenGL %s shader creation failed\n", name, tname);
  }
}

void compile_shader(GLint program, GLenum type, const char *name, const GLchar *source) {
  GLuint shader = glCreateShader(type);
  glShaderSource(shader, 1, &source, 0);
  glCompileShader(shader);
  debug_shader(shader, type, name);
  glAttachShader(program, shader);
  glDeleteShader(shader);
}

GLint compile_program(const char *name, const GLchar *vert, const GLchar *frag) {
  GLint program = glCreateProgram();
  if (vert) { compile_shader(program, GL_VERTEX_SHADER  , name, vert); }
  if (frag) { compile_shader(program, GL_FRAGMENT_SHADER, name, frag); }
  glLinkProgram(program);
  debug_program(program, name);
  return program;
}

void key_press_handler(GLFWwindow *window, int key, int scancode, int action, int mods) {
  (void) scancode;
  (void) mods;
  if (action == GLFW_PRESS) {
    switch (key) {
      case GLFW_KEY_Q: case GLFW_KEY_ESCAPE: glfwSetWindowShouldClose(window, GL_TRUE); break;
    }
  }
}

GLFWwindow *create_window(int major, int minor, int width, int height) {
  glfwWindowHint(GLFW_CONTEXT_VERSION_MAJOR, major);
  glfwWindowHint(GLFW_CONTEXT_VERSION_MINOR, minor);
```

```
  glfwWindowHint(GLFW_OPENGL_FORWARD_COMPAT, GL_TRUE);
  glfwWindowHint(GLFW_OPENGL_PROFILE, GLFW_OPENGL_CORE_PROFILE);
  glfwWindowHint(GLFW_RESIZABLE, GL_FALSE);
  GLFWwindow *window = glfwCreateWindow(width, height, "clive", 0, 0);
  if (! window) {
    fprintf(stderr, "couldn't create window with OpenGL core %d.%d context\n", major, minor);
  }
  return window;
}

GLFWwindow *create_context(int width, int height) {
  if (! glfwInit()) {
    fprintf(stderr, "couldn't initialize glfw\n");
    return 0;
  }
  int major, minor;
  GLFWwindow *window = create_window(major = 3, minor = 3, width, height);
  if (! window) {
    fprintf(stderr, "couldn't create window\n");
    return 0;
  }
  glfwMakeContextCurrent(window);
  glewExperimental = GL_TRUE;
  glewInit();
  glGetError();
  return window;
}

void initialize_gl(int screen_width, int screen_height, int webcam_width, int webcam_height, int text_buffer_width, int text_buffer_height) {
  GLuint program = compile_program("text", text_vert, text_frag);
  glUseProgram(program);
  GLint ipos = glGetAttribLocation(program, "pos");
  GLint itc = glGetAttribLocation(program, "tc");
  GLint u_screen_size = glGetUniformLocation(program, "screen_size");
  GLint u_webcam_size = glGetUniformLocation(program, "webcam_size");
  u_text_size = glGetUniformLocation(program, "text_size");
  GLint u_webcam = glGetUniformLocation(program, "webcam");
  GLint u_ascii = glGetUniformLocation(program, "ascii");
  GLint u_text = glGetUniformLocation(program, "text");
  u_hue_shift = glGetUniformLocation(program, "hue_shift");
  u_hue_rotate = glGetUniformLocation(program, "hue_rotate");
  u_hue_unrotate = glGetUniformLocation(program, "hue_unrotate");
  u_fly_size = glGetUniformLocation(program, "fly_size");
  glUniform1i(u_webcam, 0);
  glUniform1i(u_text, 1);
  glUniform1i(u_ascii, 2);
  glUniform2f(u_screen_size, screen_width, screen_height);
  glUniform2f(u_webcam_size, webcam_width, webcam_height);
  GLuint vao;
  glGenVertexArrays(1, &vao);
  glBindVertexArray(vao);
  GLuint vbo;
  glGenBuffers(1, &vbo);
  glBindBuffer(GL_ARRAY_BUFFER, vbo);
  GLfloat vbo_data[] =
```

```
      { -1, -1, 0, 1
      , -1,  1, 0, 0
      ,  1, -1, 1, 1
      ,  1,  1, 1, 0
      };
    glBufferData(GL_ARRAY_BUFFER, 16 * sizeof(GLfloat), vbo_data, GL_STATIC_DRAW);
    glVertexAttribPointer(ipos, 2, GL_FLOAT, GL_FALSE, 4 * sizeof(GLfloat), 0);
    glVertexAttribPointer(itc, 2, GL_FLOAT, GL_FALSE, 4 * sizeof(GLfloat), ((char *)0) + 2 * sizeof(GLfloat));
    glEnableVertexAttribArray(ipos);
    glEnableVertexAttribArray(itc);
    glActiveTexture(GL_TEXTURE0);
    GLuint t_webcam = 0;
    glGenTextures(1, &t_webcam);
    glBindTexture(GL_TEXTURE_2D, t_webcam);
    glTexImage2D(GL_TEXTURE_2D, 0, GL_RGB, webcam_width, webcam_height, 0, GL_RGB, GL_UNSIGNED_BYTE, 0);
    glGenerateMipmap(GL_TEXTURE_2D);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR_MIPMAP_LINEAR);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP_TO_EDGE);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP_TO_EDGE);
    glActiveTexture(GL_TEXTURE1);
    GLuint t_text = 0;
    glGenTextures(1, &t_text);
    glBindTexture(GL_TEXTURE_2D, t_text);
    glTexImage2D(GL_TEXTURE_2D, 0, GL_RED, text_buffer_width, text_buffer_height, 0, GL_RED, GL_UNSIGNED_BYTE, 0);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_NEAREST);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP_TO_EDGE);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP_TO_EDGE);
    glActiveTexture(GL_TEXTURE2);
    GLuint t_ascii = 0;
    glGenTextures(1, &t_ascii);
    glBindTexture(GL_TEXTURE_2D, t_ascii);
    unsigned char *ascii_data = (unsigned char *) malloc(2048 * 768);
    FILE *ascii_file = fopen("ascii.pgm", "rb");
    fscanf(ascii_file, "P5\n2048 768\n255");
    int c = fgetc(ascii_file);
    assert(c == '\n');
    fread(ascii_data, 2048 * 768, 1, ascii_file);
    fclose(ascii_file);
    glTexImage2D(GL_TEXTURE_2D, 0, GL_RED, 2048, 768, 0, GL_RED, GL_UNSIGNED_BYTE, ascii_data);
    free(ascii_data);
    glGenerateMipmap(GL_TEXTURE_2D);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR_MIPMAP_LINEAR);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP_TO_EDGE);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP_TO_EDGE);
}

int initialize_webcam(const char *dev, int *width, int *height) {
    int webcam = v4l2_open(dev, O_RDWR | O_NONBLOCK, 0);
```

```
  struct v4l2_capability cap;
  memset(&cap, 0, sizeof(cap));
375 if (-1 == v4l2_ioctl(webcam, VIDIOC_QUERYCAP, &cap)) {
    fprintf(stderr, "not a video device\n");
    exit(1);
  }
  if (! (cap.capabilities & V4L2_CAP_VIDEO_CAPTURE)) {
380   fprintf(stderr, "not a video capture device\n");
    exit(1);
  }
  if (! (cap.capabilities & V4L2_CAP_READWRITE)) {
    fprintf(stderr, "no support for read i/o\n");
385   exit(1);
  }
  struct v4l2_format format;
  memset(&format, 0, sizeof(format));
  format.type = V4L2_BUF_TYPE_VIDEO_CAPTURE;
390 if (-1 == v4l2_ioctl(webcam, VIDIOC_G_FMT, &format)) {
    fprintf(stderr, "couldn't get video format\n");
    exit(1);
  }
  *width = format.fmt.pix.width;
395 *height = format.fmt.pix.height;
  format.fmt.pix.pixelformat = V4L2_PIX_FMT_MJPEG;
  if (-1 == v4l2_ioctl(webcam, VIDIOC_S_FMT, &format)) {
    fprintf(stderr, "couldn't set video format\n");
    exit(1);
400 }
  if (format.fmt.pix.pixelformat != V4L2_PIX_FMT_MJPEG) {
    fprintf(stderr, "video format mismatch\n");
    exit(1);
  }
405 struct v4l2_requestbuffers req;
  memset(&req, 0, sizeof(req));
  req.count = 4;
  req.type = V4L2_BUF_TYPE_VIDEO_CAPTURE;
  req.memory = V4L2_MEMORY_MMAP;
410 if (-1 == v4l2_ioctl(webcam, VIDIOC_REQBUFS, &req)) {
    fprintf(stderr, "couldn't set video reqbufs\n");
    exit(1);
  }
  webcam_buffers = (struct webcam_buffer *) calloc(req.count, sizeof(*↙
      ↳ webcam_buffers));
415 struct v4l2_buffer buf;
  unsigned int n_buffers;
  for (n_buffers = 0; n_buffers < req.count; ++n_buffers) {
    memset(&buf, 0, sizeof(buf));
    buf.type        = V4L2_BUF_TYPE_VIDEO_CAPTURE;
420   buf.memory      = V4L2_MEMORY_MMAP;
    buf.index       = n_buffers;
    if (-1 == v4l2_ioctl(webcam, VIDIOC_QUERYBUF, &buf)) {
      fprintf(stderr, "couldn't query video buffer\n");
      exit(1);
425   }
    webcam_buffers[n_buffers].length = buf.length;
    webcam_buffers[n_buffers].start = v4l2_mmap(0, buf.length, PROT_READ | ↙
        ↳ PROT_WRITE, MAP_SHARED, webcam, buf.m.offset);
```

```
    if (MAP_FAILED == webcam_buffers[n_buffers].start) {
      fprintf(stderr, "couldn't mmap video buffer\n");
      exit(1);
    }
  }
  for (unsigned int i = 0; i < n_buffers; ++i) {
    memset(&buf, 0, sizeof(buf));
    buf.type = V4L2_BUF_TYPE_VIDEO_CAPTURE;
    buf.memory = V4L2_MEMORY_MMAP;
    buf.index = i;
    if (-1 == v4l2_ioctl(webcam, VIDIOC_QBUF, &buf)) {
      fprintf(stderr, "couldn't queue video buffer\n");
      exit(1);
    }
  }
  enum v4l2_buf_type type = V4L2_BUF_TYPE_VIDEO_CAPTURE;
  if (-1 == v4l2_ioctl(webcam, VIDIOC_STREAMON, &type)) {
    fprintf(stderr, "couldn't start video streaming\n");
    exit(1);
  }
  webcam_buffer = (unsigned char *) malloc(*width * *height * 3);
  webcam_buffer_ptrs = (unsigned char **) malloc(*height * sizeof(*↙
    ↳ webcam_buffer_ptrs));
  for (int i = 0; i < *height; ++i)
    webcam_buffer_ptrs[i] = webcam_buffer + i * *width * 3;
  return webcam;
}

struct jerr {
  struct jpeg_error_mgr e;
  jmp_buf j;
};

void jerr_handler(j_common_ptr cinfo)
{
  struct jerr *jerr = (struct jerr *) cinfo->err;
  longjmp(jerr->j, 1);
}

int read_webcam(int webcam, int width, int height) {
  fd_set fds;
  FD_ZERO(&fds);
  FD_SET(webcam, &fds);
  struct timeval tv;
  tv.tv_sec = 0;
  tv.tv_usec = 1000000 / 60;
  int r = select(webcam + 1, &fds, 0, 0, &tv);
  if (r != 1) {
    return 0;
  }
  struct v4l2_buffer buf;
  memset(&buf, 0, sizeof(buf));
  buf.type = V4L2_BUF_TYPE_VIDEO_CAPTURE;
  buf.memory = V4L2_MEMORY_MMAP;
  if (-1 == v4l2_ioctl(webcam, VIDIOC_DQBUF, &buf)) {
    return 0;
  }
```

```
485   struct jpeg_decompress_struct cinfo;
      struct jerr jerr;
      cinfo.err = jpeg_std_error(&jerr.e);
      jerr.e.error_exit = jerr_handler;
      if (setjmp(jerr.j)) {
490     jpeg_destroy_decompress(&cinfo);
        v4l2_ioctl(webcam, VIDIOC_QBUF, &buf);
        return 0;
      }
      jpeg_create_decompress(&cinfo);
495   jpeg_mem_src(&cinfo, (const unsigned char*) webcam_buffers[buf.index].start, ↙
        ↳ buf.length);
      jpeg_read_header(&cinfo, 0);
      jpeg_start_decompress(&cinfo);
      if (cinfo.output_width == (unsigned int) width && cinfo.output_height == (↙
        ↳ unsigned int) height && cinfo.output_components == 3)
        while (cinfo.output_scanline < cinfo.output_height)
500       jpeg_read_scanlines(&cinfo, webcam_buffer_ptrs + cinfo.output_scanline, ↙
            ↳ cinfo.output_height);
      else
        fprintf(stderr, "ERR webcam format mismatch\n");
      jpeg_finish_decompress(&cinfo);
      jpeg_destroy_decompress(&cinfo);
505
      v4l2_ioctl(webcam, VIDIOC_QBUF, &buf);
      return 1;
    }

510 void deinitialize_webcam(int webcam) {
      enum v4l2_buf_type type = V4L2_BUF_TYPE_VIDEO_CAPTURE;
      if (-1 == v4l2_ioctl(webcam, VIDIOC_STREAMOFF, &type)) {
        fprintf(stderr, "couldn't stop video streaming\n");
        exit(1);
515   }
      for (int i = 0; i < 4; ++i) {
        v4l2_munmap(webcam_buffers[i].start, webcam_buffers[i].length);
      }
      v4l2_close(webcam);
520   free(webcam_buffer);
    }

    void errorcb(const char *desc) {
      fprintf(stderr, "JACK error: %s\n", desc);
525 }

    void shutdowncb(void *arg) {
      exit(1);
    }
530
    void atexitcb(void) {
      jack_client_close(j_client);
    }

535 struct state {
      LOP lop[14];
      HIP hip[8];
```

```
};
struct state state;

int processcb(jack_nframes_t nframes, void *arg) {
  jack_default_audio_sample_t *in[2];
  for (int c = 0; c < 2; ++c) {
    in[c] = (jack_default_audio_sample_t *) jack_port_get_buffer(j_in[c], nframes);
  }
  float lo = 0, md = 0, hi = 0;
  float lof = 400;
  float hif = 1000;
  for (jack_nframes_t i = 0; i < nframes; ++i) {
    float lo0 = lop(&state.lop[0], lop(&state.lop[1], in[0][i], lof), lof);
    float lo1 = lop(&state.lop[2], lop(&state.lop[3], in[1][i], lof), lof);
    float md0 = lop(&state.lop[4], lop(&state.lop[5], hip(&state.hip[0], hip(&state.hip[1], in[0][i], lof), lof), hif), hif);
    float md1 = lop(&state.lop[6], lop(&state.lop[7], hip(&state.hip[2], hip(&state.hip[3], in[1][i], lof), lof), hif), hif);
    float hi0 = hip(&state.hip[4], hip(&state.hip[5], in[0][i], hif), hif);
    float hi1 = hip(&state.hip[6], hip(&state.hip[7], in[1][i], hif), hif);
    lo = lop(&state.lop[8], lop(&state.lop[9], lo0 * lo0 + lo1 * lo1, 8), 8);
    md = lop(&state.lop[10], lop(&state.lop[11], md0 * md0 + md1 * md1, 8), 8);
    hi = lop(&state.lop[12], lop(&state.lop[13], hi0 * hi0 + hi1 * hi1, 8), 8);
  }
  hue_shift[0] = 1 + 16 * sqrtf(lo);
  hue_shift[1] = 1 + 16 * sqrtf(md);
  hue_shift[2] = 1 + 16 * sqrtf(hi);
  return 0;
}

void initialize_audio(void) {
  jack_set_error_function(errorcb);
  if (!(j_client = jack_client_open("visuals", JackNoStartServer, 0))) {
    fprintf(stderr, "jack server not running?\n");
    exit(1);
  }
  atexit(atexitcb);
  jack_set_process_callback(j_client, processcb, 0);
  jack_on_shutdown(j_client, shutdowncb, 0);
  // multi-channel processing
  for (int c = 0; c < 2; ++c) {
    char portname[100];
    snprintf(portname, 100, "input_%d", c + 1);
    j_in[c] = jack_port_register(j_client, portname, JACK_DEFAULT_AUDIO_TYPE, JackPortIsInput, 0);
  }
  if (jack_activate(j_client)) {
    fprintf (stderr, "cannot activate JACK client");
    exit(1);
  }
}

int main(int argc, char **argv) {
  const char *webcam_dev = "/dev/video0";
  if (argc > 1) {
    webcam_dev = argv[1];
```

```
  }
  memset(&state, 0, sizeof(state));
  initialize_audio();
  int text_buffer_width = 128, text_buffer_height = 64;
595 char *text_buffer = (char *) malloc(text_buffer_width * text_buffer_height);
  int screen_width = 1280, screen_height = 720;
  int webcam_width = 0, webcam_height = 0;
  int webcam = initialize_webcam(webcam_dev, &webcam_width, &webcam_height);
  if (webcam < 0) {
600   return 1;
  }
  GLFWwindow *window = create_context(screen_width, screen_height);
  if (! window) {
    return 1;
605 }
  glfwSetWindowPos(window, 1366, 0);
  glfwSetKeyCallback(window, key_press_handler);
  initialize_gl(screen_width, screen_height, webcam_width, webcam_height, ↙
      ↳ text_buffer_width, text_buffer_height);
  int frame = 0;
610 while (! glfwWindowShouldClose(window)) {
    // read webcam
    if (read_webcam(webcam, webcam_width, webcam_height)) {
      glActiveTexture(GL_TEXTURE0);
      glTexSubImage2D(GL_TEXTURE_2D, 0, 0, 0, webcam_width, webcam_height, ↙
          ↳ GL_RGB, GL_UNSIGNED_BYTE, webcam_buffer);
615   glGenerateMipmap(GL_TEXTURE_2D);
    }
    // read git diff
    FILE *diff_file = popen("git diff HEAD~1", "r");
    if (diff_file) {
620   memset(text_buffer, ' ', text_buffer_width * text_buffer_height);
      int text_width = 0;
      int text_height = 0;
      char *line = 0;
      size_t len = 0;
625   ssize_t read;
      for (int y = 1; y < text_buffer_height - 1; ++y) {
        if ((read = getline(&line, &len, diff_file)) == -1) {
          break;
        }
630     for (int x = 1; x < text_buffer_width - 1 && x < read; ++x) {
          text_buffer[y * text_buffer_width + x] = line[x - 1];
          text_width = text_width > x ? text_width : x;
        }
        text_height = y;
635   }
      text_width += 2;
      text_height += 2;
      free(line);
      pclose(diff_file);
640   glActiveTexture(GL_TEXTURE1);
      glTexSubImage2D(GL_TEXTURE_2D, 0, 0, 0, text_buffer_width, ↙
          ↳ text_buffer_height, GL_RED, GL_UNSIGNED_BYTE, text_buffer);
      glUniform2f(u_text_size, text_width, text_height);
    }
    // read hue shift
```

```
645     glUniform3fv(u_hue_shift, 1, hue_shift);
        float theta = twopi * frame / 600;
        glm::mat4 rot4 = glm::rotate(glm::mat4(1.0), theta, glm::vec3(1.0f,1.0f,1.0f↙
            ↳ ));
        glm::mat4 unrot4 = glm::rotate(glm::mat4(1.0), -theta, glm::vec3(1.0f,1.0f↙
            ↳ ,1.0f));
        glm::mat3 rot(rot4);
650     glm::mat3 unrot(unrot4);
        glUniformMatrix3fv(u_hue_rotate, 1, GL_FALSE, &rot[0][0]);
        glUniformMatrix3fv(u_hue_unrotate, 1, GL_FALSE, &unrot[0][0]);
        glUniform1f(u_fly_size, 24 * pow(0.5, 0.5 + 0.5 * cos(twopi * frame / 2500))↙
            ↳ );
        // draw
655     glDrawArrays(GL_TRIANGLE_STRIP, 0, 4);
        glfwSwapBuffers(window);
        glfwPollEvents();
        frame++;
      }
660   glfwTerminate();
      deinitialize_webcam(webcam);
      free(text_buffer);
      return 0;
    }
```

38 .gitignore

```
client/clive-client
server/clive-server
extra/visuals
*.so
5   *.wav
*.ogg
*.mp3
*.pgm
*.html
10  *.txt
*.s
```

39 GPL3.md

```
### GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

5   Copyright (C) 2007 Free Software Foundation, Inc.
<https://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this
license document, but changing it is not allowed.
10
### Preamble

The GNU General Public License is a free, copyleft license for
software and other kinds of works.
15
The licenses for most software and other practical works are designed
to take away your freedom to share and change the works. By contrast,
```

```
the GNU General Public License is intended to guarantee your freedom
to share and change all versions of a program--to make sure it remains
free software for all its users. We, the Free Software Foundation, use
the GNU General Public License for most of our software; it applies
also to any other work released this way by its authors. You can apply
it to your programs, too.

When we speak of free software, we are referring to freedom, not
price. Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
them if you wish), that you receive source code or can get it if you
want it, that you can change the software or use pieces of it in new
free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you
these rights or asking you to surrender the rights. Therefore, you
have certain responsibilities if you distribute copies of the
software, or if you modify it: responsibilities to respect the freedom
of others.

For example, if you distribute copies of such a program, whether
gratis or for a fee, you must pass on to the recipients the same
freedoms that you received. You must make sure that they, too, receive
or can get the source code. And you must show them these terms so they
know their rights.

Developers that use the GNU GPL protect your rights with two steps:
(1) assert copyright on the software, and (2) offer you this License
giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains
that there is no warranty for this free software. For both users' and
authors' sake, the GPL requires that modified versions be marked as
changed, so that their problems will not be attributed erroneously to
authors of previous versions.

Some devices are designed to deny users access to install or run
modified versions of the software inside them, although the
manufacturer can do so. This is fundamentally incompatible with the
aim of protecting users' freedom to change the software. The
systematic pattern of such abuse occurs in the area of products for
individuals to use, which is precisely where it is most unacceptable.
Therefore, we have designed this version of the GPL to prohibit the
practice for those products. If such problems arise substantially in
other domains, we stand ready to extend this provision to those
domains in future versions of the GPL, as needed to protect the
freedom of users.

Finally, every program is threatened constantly by software patents.
States should not allow patents to restrict development and use of
software on general-purpose computers, but in those that do, we wish
to avoid the special danger that patents applied to a free program
could make it effectively proprietary. To prevent this, the GPL
assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and
modification follow.
```

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major

Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or

similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other

parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.
- c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
- d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.
- e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of

product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work,

for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

```
Moreover, your license from a particular copyright holder is
reinstated permanently if the copyright holder notifies you of the
violation by some reasonable means, this is the first time you have
received notice of violation of this License (for any work) from that
copyright holder, and you cure the violation prior to 30 days after
your receipt of the notice.

Termination of your rights under this section does not terminate the
licenses of parties who have received copies or rights from you under
this License. If your rights have been terminated and not permanently
reinstated, you do not qualify to receive new licenses for the same
material under section 10.

#### 9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run
a copy of the Program. Ancillary propagation of a covered work
occurring solely as a consequence of using peer-to-peer transmission
to receive a copy likewise does not require acceptance. However,
nothing other than this License grants you permission to propagate or
modify any covered work. These actions infringe copyright if you do
not accept this License. Therefore, by modifying or propagating a
covered work, you indicate your acceptance of this License to do so.

#### 10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically
receives a license from the original licensors, to run, modify and
propagate that work, subject to this License. You are not responsible
for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an
organization, or substantially all assets of one, or subdividing an
organization, or merging organizations. If propagation of a covered
work results from an entity transaction, each party to that
transaction who receives a copy of the work also receives whatever
licenses to the work the party's predecessor in interest had or could
give under the previous paragraph, plus a right to possession of the
Corresponding Source of the work from the predecessor in interest, if
the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the
rights granted or affirmed under this License. For example, you may
not impose a license fee, royalty, or other charge for exercise of
rights granted under this License, and you may not initiate litigation
(including a cross-claim or counterclaim in a lawsuit) alleging that
any patent claim is infringed by making, using, selling, offering for
sale, or importing the Program or any portion of it.

#### 11. Patents.

A "contributor" is a copyright holder who authorizes use under this
License of the Program or a work on which the Program is based. The
work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned
```

or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

```
Nothing in this License shall be construed as excluding or limiting
any implied license or other defenses to infringement that may
otherwise be available to you under applicable patent law.

#### 12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License. If you cannot convey a
covered work so as to satisfy simultaneously your obligations under
this License and any other pertinent obligations, then as a
consequence you may not convey it at all. For example, if you agree to
terms that obligate you to collect a royalty for further conveying
from those to whom you convey the Program, the only way you could
satisfy both those terms and this License would be to refrain entirely
from conveying the Program.

#### 13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have
permission to link or combine any covered work with a work licensed
under version 3 of the GNU Affero General Public License into a single
combined work, and to convey the resulting work. The terms of this
License will continue to apply to the part which is the covered work,
but the special requirements of the GNU Affero General Public License,
section 13, concerning interaction through a network will apply to the
combination as such.

#### 14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions
of the GNU General Public License from time to time. Such new versions
will be similar in spirit to the present version, but may differ in
detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program
specifies that a certain numbered version of the GNU General Public
License "or any later version" applies to it, you have the option of
following the terms and conditions either of that numbered version or
of any later version published by the Free Software Foundation. If the
Program does not specify a version number of the GNU General Public
License, you may choose any version ever published by the Free
Software Foundation.

If the Program specifies that a proxy can decide which future versions
of the GNU General Public License can be used, that proxy's public
statement of acceptance of a version permanently authorizes you to
choose that version for the Program.

Later license versions may give you additional or different
permissions. However, no additional obligations are imposed on any
author or copyright holder as a result of your choosing to follow a
later version.

#### 15. Disclaimer of Warranty.
```

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

> <one line to give the program's name and a brief idea of what it does.>
> Copyright (C) <year> <name of author>
>
> This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.
>
> This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

```
645
        You should have received a copy of the GNU General Public License
        along with this program.  If not, see <https://www.gnu.org/licenses/>.

    Also add information on how to contact you by electronic and paper
650 mail.

    If the program does terminal interaction, make it output a short
    notice like this when it starts in an interactive mode:

655         <program>  Copyright (C) <year>  <name of author>
            This program comes with ABSOLUTELY NO WARRANTY; for details type `show w↙
                ↳ '.
            This is free software, and you are welcome to redistribute it
            under certain conditions; type `show c' for details.

660 The hypothetical commands \`show w' and \`show c' should show the
    appropriate parts of the General Public License. Of course, your
    program's commands might be different; for a GUI interface, you would
    use an "about box".

665 You should also get your employer (if you work as a programmer) or
    school, if any, to sign a "copyright disclaimer" for the program, if
    necessary. For more information on this, and how to apply and follow
    the GNU GPL, see <https://www.gnu.org/licenses/>.

670 The GNU General Public License does not permit incorporating your
    program into proprietary programs. If your program is a subroutine
    library, you may consider it more useful to permit linking proprietary
    applications with the library. If this is what you want to do, use the
    GNU Lesser General Public License instead of this License. But first,
675 please read <https://www.gnu.org/licenses/why-not-lgpl.html>.
```

40 launch/local-native-sse.sh

```
    #!/bin/bash
    for c in $(cat /proc/cpuinfo | grep "^processor" | sed "s/processor\t: //")
    do
      sudo cpufreq-set -c "${c}" -g performance
 5  done
    make -C client
    make -C server
    mkdir -p build
    SESSION="session-$(date -u +%F-%H%M%S)"
10  git checkout -b "${SESSION}"
    (
    ecasound -q -G:jack,record -f:f32,2,48000 -i:jack -o "${SESSION}.wav" &
    sleep 5
    xterm -geometry 80x10+0+0   -T server -e bash -c 'cd server ; while true ; do ./↙
        ↳ clive-server ; done' &
15  xterm -geometry 80x30+0+154 -T client -e bash -c 'cd client ; while true ; do ./↙
        ↳ clive-client native-sse.mk ; done' &
    xterm -geometry 80x8+0+588  -T htop   -e bash -c 'while true ; do htop ; done' &
    { cd client ; while true ; do geany -mist dsp.h dsp/*.h go.c ; done } &
    wait
    )
```

41 launch/remote-bela.sh

```
#!/bin/bash
REMOTEHOST="${1:-root@192.168.6.2}"
REMOTEPATH="${2:-Bela/projects/clive-core}"
#ssh-add
for c in $(cat /proc/cpuinfo | grep "^processor" | sed "s/processor\t: //")
do
  sudo cpufreq-set -c "${c}" -g performance
done
make -C client
mkdir -p build
rm -f build/go*.so
sshfs "${REMOTEHOST}:${REMOTEPATH}/build" ./build
SESSION="session-$(date -u +%F-%H%M%S)"
git checkout -b "${SESSION}"
(
#ssh "${REMOTEHOST}" "ecasound -q -G:jack,record,send -f:f32,2,48000 -i:jack -o ↙
    ↪ '${REMOTEPATH}/${SESSION}.wav'" &
sleep 5
xterm -geometry 80x10+0+0   -T server -e ssh "${REMOTEHOST}" "${REMOTEPATH}/↙
    ↪ server/clive-server.sh" &
xterm -geometry 80x30+0+154 -T client -e bash -c 'cd client ; while true ; do ./↙
    ↪ clive-client bela.mk ; sleep 1 ; done' &
xterm -geometry 80x8+0+588  -T htop   -e bash -c 'while true ; do htop ; sleep 1↙
    ↪  ; done' &
{ cd client ; while true ; do geany -mist dsp.h dsp/*.h go.h go.c ; done } &
wait
)
ssh-add -d
```

42 launch/remote-rpi3b.sh

```
#!/bin/bash
REMOTEHOST="${1:-pi@raspberrypi.home}"
REMOTEPATH="${2:-code/code.mathr.co.uk/clive-core}"
ssh-add
for c in $(cat /proc/cpuinfo | grep "^processor" | sed "s/processor\t: //")
do
  sudo cpufreq-set -c "${c}" -g performance
done
make -C client
mkdir -p build
rm -f build/go*.so
sshfs "${REMOTEHOST}:${REMOTEPATH}/build" ./build
SESSION="session-$(date -u +%F-%H%M%S)"
git checkout -b "${SESSION}"
(
ssh "${REMOTEHOST}" "ecasound -q -G:jack,record,send -f:f32,2,48000 -i:jack -o '↙
    ↪ ${REMOTEPATH}/${SESSION}.wav'" &
sleep 5
xterm -geometry 80x10+0+0   -T server -e ssh "${REMOTEHOST}" "${REMOTEPATH}/↙
    ↪ server/clive-server.sh" &
xterm -geometry 80x30+0+154 -T client -e bash -c 'cd client ; while true ; do ./↙
    ↪ clive-client rpi3b-64.mk ; sleep 1 ; done' &
xterm -geometry 80x8+0+588  -T htop   -e bash -c 'while true ; do htop ; sleep 1↙
    ↪  ; done' &
```

```
{ cd client ; while true ; do geany -mist dsp.h dsp/*.h go.c ; done } &
wait
)
ssh-add -d
```

43 LICENSE.md

```
# clive licensing

- clive core code is under dual BSD3 and GPL3 license (master branch)
- clive documentation is under CC-BY-SA license (master branch)
- Claude's clive performances are under GPL3 license (other branches)
```

44 README.md

```
# clive

C audio live-coding skeleton
<https://mathr.co.uk/clive>.

## audio api support

- jack (recommended, default, with scripts for recording)
- bela (for the single board computer)
- sdl2 (playback only, works with pulseaudio)

## deps

  - linux (POSIX for `-ldl`, Linux for `inotify`)

### server

  - make
  - gcc
  - git
  - jack-dev
  - libsdl2-dev
  - pkg-config
  - ecasound

### client

  - make
  - gcc
  - git
  - geany
  - htop
  - xterm

## usage

Editor save -> client recompiles code -> server reloads binary.

Automatic recording of code edits to git history.

Automatic recording of audio output to WAV file.
```

```
### local

Run editor, client, server all on one machine.  Assumes x86(_64) with SSE:

    git clone https://code.mathr.co.uk/clive-core.git
    cd clive-core
    ./launch/local-native-sse.sh

### remote

Run editor and client on local machine, server on a remote machine.

#### rpi

Assumes Raspberry Pi Model 3 B remote running Debian Buster aarch64 (arm64)
from unofficial unsupported image at <https://wiki.debian.org/RaspberryPi3>.

Client machine needs `gcc-8-aarch64-linux-gnu` installed.

Client user needs an ssh key.

Server machine needs `~/.ssh/authorized_keys` for the client user.

Server machine needs working JACK audio server; for some tips see:
<https://mathr.co.uk/blog/2018-10-23_jack_on_top_of_pulseaudio.html>
Alternatively, compiling the clive server with `make API=sdl2` might allow
audio via Pulseaudio to work more directly, but recording via `ecasound`
won't work out of the box.  To be tested and documented properly.

Run once on server and client:

    mkdir -p ~/code
    cd ~/code
    git clone https://code.mathr.co.uk/clive-core.git

Run each session on client:

    cd ~/code/clive-core
    ./launch/remote-rpi3b.sh user@server code/clive-core

Note: cleanup is not yet automatic; you need to kill left-over tasks on
the server host manually.  `ps aux | grep clive` may offer some hints.

#### bela

For Bela running its system (`armv7l`) connected via USB to a Debian
machine.  No ssh keys needed as passwordless root works by default.

Server machine (the Bela) needs directory layout like
`/root/Bela/projects/clive-core`.  You need the `bela` branch of the
repository.  Compile `clive-server` on the Bela with `make API=bela`.
Launch it by `cd server; ./clive-server`.
The script `launch/remote-bela.sh` does this automatically (and more),
but read it first in case it does not match your setup.

Client machine needs to be Debian Stretch (native or a chroot).  You
can install a chroot with `debootstrap`, read its fine documentation.
```

Debian Buster won't work because its `glibc/libm` is too new for the Bela, giving this error message when `clive-server` attempts to load the cross-compiled code:

```
/lib/arm-linux-gnueabihf/libm.so.6: version 'GLIBC_2.27' not found
```

Client machine needs `sshfs` installed. You might need to create the `/dev/fuse` device node inside the chroot if you are using one:

```
mknod /dev/fuse c 10 229
chmod g+w,o+w /dev/fuse
```

Client machine needs `gcc-6-arm-linux-gnueabihf` cross-compiler installed. If you get errors relating to `crti.o` or `-lm` or other basic functionality, you might need to install more packages, not sure which but I installed `g++-6-arm-linux-gnueabihf` and it worked for me.

Client machine chroot user numeric ids should match your regular user, if you have a single user then both should be 1000. This allows you to run a graphical-mode text editor outside the chroot, modifying files inside it with the correct permissions.

If using a chroot, you need to `sshfs` mount the Bela `clive-core/build` directory over the `clive-core/build` directory in the chroot. If using a chroot, `./clive-client bela.mk` should be run inside it. If using a native client host, you can use `launch/remote-bela.sh` which does both of these things (and more).

Note: cleanup is not yet automatic; you need to kill left-over tasks on the Bela manually. `ps aux | grep clive` may offer some hints.

Note: there is no audio recording support for Bela.

html export

HTML5 with audio player synced to syntax-highlighted source code diffs:

```
./extra/session2html.sh session-branch [start-datetime]
```

Expects the session audio recording encoded to `.ogg`.

Adjust `start-datetime` if the audio is out of sync (most commonly by an hour in summer due to daylight saving time).

webvtt would be better than Javascript searching a dictionary on each time change, if only it worked...

timebase idea

some kind of encoding of timestamps as audio, so cut/paste editing of sessions can still be linked to code changes (ie, at each cut boundary get a combined diff of all the changes that were skipped)

45 server/aarch64.mk

```
OFLAGS=-O3 -march=armv8-a+crc
```

46 server/armv7l.mk

```
OFLAGS=-O3 -march=native
```

47 server/bela.mk

```
AFLAGS=-DBELA -I/root/Bela/include -L/root/Bela/lib -lbela
```

48 server/clive-server.c

```
#include <stdbool.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>

#include <signal.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <unistd.h>

#ifdef __x86_64__
#include <fenv.h>
#endif

#include <linux/limits.h>
#include <sys/inotify.h>
#include <dlfcn.h>

#if defined(BELA) + defined(JACK) + defined(SDL2) != 1
#error exactly one audio API must be defined
#endif

#ifdef BELA
#include <stdarg.h>
#include <Bela.h>
#endif

#ifdef JACK
#include <jack/jack.h>
#endif

#ifdef SDL2
#include <SDL2/SDL.h>
#include <SDL2/SDL_audio.h>
#endif

// TODO: split into IN and OUT
// TODO: make it settable as a command line argument
#define CHANNELS 2

// per-sample callback implemented in go.so
typedef int callback(void *, int, const float *, float *);

// default silent callback
static int deffunc(void *data, int channels, const float *in, float *out) {
```

```
  (void) data;
  (void) in;
  for (int c = 0; c < channels; ++c) {
    out[c] = 0;
  }
  return 0;
}

static struct {
#ifdef BELA
#endif
#ifdef JACK
  jack_client_t *client;
  jack_port_t *in[CHANNELS], *out[CHANNELS];
#endif
  void *data;
  callback * volatile func;
  int volatile reload;
} state;

// race mitigation
volatile int inprocesscb = 0;

volatile int running = 1;
void interrupt_handler(int var)
{
  (void) var;
  running = 0;
}

#ifdef BELA
bool setup(BelaContext *context, void *userData)
{
  (void) context;
  (void) userData;
  return true;
}

void cleanup(BelaContext *context, void *userData)
{
  (void) context;
  (void) userData;
}

void render(BelaContext *context, void *userData)
{
  (void) userData;
  inprocesscb = 1; // race mitigation
  // get callback
  callback *f = state.func;
  // handle reloading
  if (state.reload) {
    int *reloaded = state.data;
    *reloaded = 1;
    state.reload = 0;
  }
  // loop over samples
```

```
  for (unsigned int n = 0; n < context->audioFrames; n++)
  {
    // to buffer
    float in[CHANNELS];
    float out[CHANNELS];
    for (unsigned int channel = 0; channel < CHANNELS; channel++)
    {
      if (channel < context->audioInChannels)
        in[channel] = audioRead(context, n, channel);
      else
        in[channel] = 0;
    }
    for (unsigned int channel = 0; channel < context->audioOutChannels; channel↙
        ↳ ++)
    {
      out[channel] = 0;
    }
    // callback
    f(state.data, CHANNELS, in, out);
    // from buffer
    for (unsigned int channel = 0; channel < CHANNELS; channel++)
    {
      if (channel < context->audioOutChannels)
        audioWrite(context, n, channel, out[channel]);
    }
  }
  // done
  inprocesscb = 0; // race mitigation
}
#endif

#ifdef JACK
static int processcb(jack_nframes_t nframes, void *arg) {
  inprocesscb = 1; // race mitigation
  // set floating point environment for denormal->0.0
#ifdef __x86_64__
  fenv_t fe;
  fegetenv(&fe);
  unsigned int old_mxcsr = fe.__mxcsr;
  fe.__mxcsr |= 0x8040; // set DAZ and FTZ
  fesetenv(&fe);
#endif
  // get jack buffers
  jack_default_audio_sample_t *in[CHANNELS];
  jack_default_audio_sample_t *out[CHANNELS];
  for (int c = 0; c < CHANNELS; ++c) {
    in[c] = (jack_default_audio_sample_t *) jack_port_get_buffer(state.in[c], ↙
        ↳ nframes);
    out[c] = (jack_default_audio_sample_t *) jack_port_get_buffer(state.out[c], ↙
        ↳ nframes);
  }
  // get callback
  callback *f = state.func;
  // handle reloading
  if (state.reload) {
    int *reloaded = state.data;
    *reloaded = 1;
```

```
      state.reload = 0;
    }
    // loop over samples
    for (jack_nframes_t i = 0; i < nframes; ++i) {
      // to buffer
      float ini[CHANNELS];
      float outi[CHANNELS];
      for (int c = 0; c < CHANNELS; ++c) {
        ini [c] = in[c][i];
        outi[c] = 0;
      }
      // callback
      /* int inhibit_reload = */ f(state.data, CHANNELS, ini, outi);
      // from buffer
      for (int c = 0; c < CHANNELS; ++c) {
        out[c][i] = outi[c];
      }
    }
    // restore floating point environment
#ifdef __x86_64__
    fe.__mxcsr = old_mxcsr;
    fesetenv(&fe);
#endif
    // done
    inprocesscb = 0; // race mitigation
    return 0;
}

static void errorcb(const char *desc) {
    fprintf(stderr, "JACK error: %s\n", desc);
}

static void shutdowncb(void *arg) {
    exit(1);
}

static void atexitcb(void) {
    jack_client_close(state.client);
}
#endif

#ifdef SDL2
static void audiocb(void *userdata, Uint8 *stream, int len) {
    inprocesscb = 1; // race mitigation
    // set floating point environment for denormal->0.0
#ifdef __x86_64__
    fenv_t fe;
    fegetenv(&fe);
    unsigned int old_mxcsr = fe.__mxcsr;
    fe.__mxcsr |= 0x8040; // set DAZ and FTZ
    fesetenv(&fe);
#endif
    // get buffers
    float *out = (float *) stream;
    int nframes = len / sizeof(float) / CHANNELS;
    // get callback
    callback *f = state.func;
```

```
215     // handle reloading
        if (state.reload) {
          int *reloaded = state.data;
          *reloaded = 1;
          state.reload = 0;
220     }
        // loop over samples
        int k = 0;
        for (int i = 0; i < nframes; ++i) {
          // to buffer
225       float ini[CHANNELS];
          float outi[CHANNELS];
          for (int c = 0; c < CHANNELS; ++c) {
            ini [c] = 0;
            outi[c] = 0;
230       }
          // callback
          /* int inhibit_reload = */ f(state.data, CHANNELS, ini, outi);
          // from buffer
          for (int c = 0; c < CHANNELS; ++c) {
235         out[k++] = outi[c];
          }
        }
        // restore floating point environment
      #ifdef __x86_64__
240     fe.__mxcsr = old_mxcsr;
        fesetenv(&fe);
      #endif
        // done
        inprocesscb = 0; // race mitigation
245   }
      #endif

      int main(int argc, char **argv) {
        srand(time(0));
250     state.func = deffunc;
        state.data = calloc(1, 64 * 1024 * 1024);
        state.reload = 0;
      #ifdef BELA
        BelaInitSettings *settings = Bela_InitSettings_alloc();
255     Bela_defaultSettings(settings);
        settings->setup = setup;
        settings->render = render;
        settings->cleanup = cleanup;
        if (Bela_initAudio(settings, 0))
260     {
          Bela_InitSettings_free(settings);
          fprintf(stderr, "could not init audio\n");
          return 1;
        }
265     Bela_InitSettings_free(settings);
        if (Bela_startAudio())
        {
          fprintf(stderr, "could not start audio\n");
          return 1;
270     }
      #endif
```

```
#ifdef JACK
  jack_set_error_function(errorcb);
  if (!(state.client = jack_client_open("clive", JackNoStartServer, 0))) {
    fprintf(stderr, "jack server not running?\n");
    return 1;
  }
  atexit(atexitcb);
  jack_set_process_callback(state.client, processcb, 0);
  jack_on_shutdown(state.client, shutdowncb, 0);
  // multi-channel processing
  for (int c = 0; c < CHANNELS; ++c) {
    char portname[100];
    snprintf(portname, 100, "input_%d", c + 1);
    state.in[c] = jack_port_register(state.client, portname, JACK_DEFAULT_AUDIO_TYPE, JackPortIsInput, 0);
    snprintf(portname, 100, "output_%d", c + 1);
    state.out[c] = jack_port_register(state.client, portname, JACK_DEFAULT_AUDIO_TYPE, JackPortIsOutput, 0);
  }
  if (jack_activate(state.client)) {
    fprintf (stderr, "cannot activate JACK client");
    return 1;
  }
  // stereo recording
  jack_connect(state.client, "clive:output_1", "record:in_1");
  jack_connect(state.client, "clive:output_2", "record:in_2");
  // stereo output to system
  jack_connect(state.client, "clive:output_1", "system:playback_1");
  jack_connect(state.client, "clive:output_2", "system:playback_2");
  // stereo output to pulse
  jack_connect(state.client, "clive:output_1", "PulseAudio JACK Source:front-left");
  jack_connect(state.client, "clive:output_2", "PulseAudio JACK Source:front-right");
  // stereo input from system
  jack_connect(state.client, "system:capture_1", "clive:input_1");
  jack_connect(state.client, "system:capture_2", "clive:input_2");
  // stereo input from pulse
  jack_connect(state.client, "PulseAudio JACK Sink:front-left", "clive:input_1");
  jack_connect(state.client, "PulseAudio JACK Sink:front-right", "clive:input_2");
#endif
#ifdef SDL2
  // initialize SDL2 audio
  SDL_Init(SDL_INIT_AUDIO);
  SDL_AudioSpec want, have;
  want.freq = 48000; // FIXME don't hardcode desired sample rate
  want.format = AUDIO_F32;
  want.channels = CHANNELS;
  want.samples = 4096; // FIXME don't hardcode buffer size
  want.callback = audiocb;
  SDL_AudioDeviceID dev = SDL_OpenAudioDevice(NULL, 0, &want, &have, SDL_AUDIO_ALLOW_ANY_CHANGE);
  if (have.format != AUDIO_F32 || have.channels != CHANNELS)
  {
    fprintf(stderr, "want: %d %d %d %d\n", want.freq, want.format, want.channels
```

```
        ↪ , want.samples);
      fprintf(stderr, "have: %d %d %d %d\n", have.freq, have.format, have.channels↙
        ↪ , have.samples);
      fprintf(stderr, "error: bad audio parameters\n");
      return 1;
325   }
    SDL_PauseAudioDevice(dev, 0);
  #endif
    signal(SIGINT, interrupt_handler);
    signal(SIGTERM, interrupt_handler);
330 // watch for filesystem changes
    int ino = inotify_init();
    if (ino == -1) {
      perror("inotify_init()");
      return 1;
335 }
    int wd = inotify_add_watch(ino, "../build/", IN_CLOSE_WRITE);
    if (wd == -1) {
      perror("inotify_add_watch()");
      return 1;
340 }
    ssize_t buf_bytes = sizeof(struct inotify_event) + NAME_MAX + 1;
    char *buf = malloc(buf_bytes);
    // double-buffering stuff
    void *old_dl = 0;
345 void *new_dl = 0;
    int which = 0;
    // main loop
    while (running) {
  /*
350 Double buffering to avoid glitches on reload. Can't dlopen the same file
  twice (or even symlinks thereof) due to aggressive caching seemingly based
  on file names by libdl, so copy the target to one of two names.
  To avoid crashing by unloading running code, open the other .so filename to the
  runnning code before swapping pointers when hopefully not in a JACK process
355 callback in the DSP thread. Finally dlclose the previous and swap the buffer
  index.
  */
      const char *copycmd[2] =
        { "cp -f ../build/go.so ../build/go.a.so"
360     , "cp -f ../build/go.so ../build/go.b.so"
        };
      const char *library[2] =
        { "../build/go.a.so"
        , "../build/go.b.so"
365     };
      if (system(copycmd[which])) {
        fprintf(stderr, "\x1b[31;1mCOPY COMMAND FAILED: '%s'\x1b[0m\n", copycmd[↙
          ↪ which]);
      }
      else if ((new_dl = dlopen(library[which], RTLD_NOW))) {
370     callback *new_cb;
        *(void **) (&new_cb) = dlsym(new_dl, "go");
        if (new_cb) {
          // race mitigation: dlclose with jack running in .so -> boom
          while (inprocesscb) ;
375       // JACK process callback not running
```

```
          state.func = new_cb;
          state.reload = 1;
          // JACK process callback might have started again
          // race mitigation
          while (inprocesscb) ;
          // JACK process callback not running, and not referencing old state.func
          if (old_dl) {
            dlclose(old_dl);
          }
          old_dl = new_dl;
          new_dl = 0;
          which = 1 - which;
          fprintf(stderr, "\x1b[32;1mRELOADED: '%p'\x1b[0m\n", *(void **) (&new_cb↙
              ↳ ));
        } else {
          fprintf(stderr, "\x1b[31;1mNO FUNCTION DEFINED: 'go()'\x1b[0m\n");
          dlclose(new_dl);
        }
      } else {
        // another race condition: the .so disappeared before load
        fprintf(stderr, "\x1b[31;1mFILE VANISHED: '%s'\x1b[0m\n", library[which]);
        const char *err = dlerror();
        if (err) {
          fprintf(stderr, "\x1b[31;1m%s\x1b[0m\n", err);
        }
      }
      // read events (blocking)
      int done = 0;
      do {
        memset(buf, 0, buf_bytes);
        ssize_t r = read(ino, buf, buf_bytes);
        if (r == -1) {
          perror("read()");
          sleep(1);
        } else {
          char *bufp = buf;
          while (bufp < buf + r)
          {
            struct inotify_event *ev = (struct inotify_event *) bufp;
            bufp += sizeof(struct inotify_event) + ev->len;
            if (ev->mask & IN_CLOSE_WRITE) {
              fprintf(stderr, "\x1b[32;1mFILE CHANGED: '%s'\x1b[0m\n", ev->name);
              if (0 == strcmp("go.so", ev->name)) {
                done = 1;
              }
            }
          }
        }
      } while (running && ! done);
    }
#ifdef BELA
    Bela_stopAudio();
    Bela_cleanupAudio();
#endif
#ifdef JACK
#endif
    // never reached
```

```
  close(ino);
  return 0;
}
```

49 server/clive-server.sh

```
#!/bin/bash
cd "$(readlink -e "$(dirname "$0")")"
make
while true
do
  ./clive-server
  sleep 1
done
```

50 server/i686.mk

```
OFLAGS=-O3 -march=native -mfpmath=sse
```

51 server/jack.mk

```
AFLAGS=-DJACK `pkg-config --cflags --libs jack`
```

52 server/Makefile

```
GCC ?= gcc

all: clive-server

ARCH ?= $(shell uname -m)
include $(ARCH).mk

API ?= jack
include $(API).mk

clive-server: clive-server.c
	$(GCC) -std=c99 -Wall -pedantic -Wextra -Wno-unused-parameter $(OFLAGS) ↙
	    ↳ $(AFLAGS) -I. -o clive-server clive-server.c -ldl -lm
```

53 server/sdl2.mk

```
AFLAGS=-DSDL2 `sdl2-config  --cflags --libs`
```

54 server/x86_64.mk

```
OFLAGS=-O3 -march=x86-64 -mfpmath=sse
```