puzzle

Claude Heiland-Allen

2008–2019

Contents

1 COPYING

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

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.

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

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

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

    <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.

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".

  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
<http://www.gnu.org/licenses/>.

  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, please read
<http://www.gnu.org/philosophy/why-not-lgpl.html>.
```

2 COPYING-drums

```
drum synth patches extracted from:
Cheesebox/AnalogDrumEngine
Copyright (c) 2008 Andy Farnell
License: GPL
<https://web.archive.org/web/20150316000621/http://obiwannabe.co.uk/html/toys/
```

```
    ↳ cheesebox/cheesebox.html>
<https://www.gnu.org/copyleft/gpl.html>
```

3 edit.sh

```
#!/bin/bash
"${HOME}/opt/pd/0.49-0/bin/pd" -noprefs -stderr -nrt \
    -lib Gem:pdlua \
    -r 48000 -jack -channels 2 \
    -path ./puzzle/drums/ -open puzzle/PUZZLE3X3.pd \
    -send "; PUZZLE 256 256 1110 24 ${DISPLAY} 0"
```

4 .gitignore

```
Gem
Gem.bc
pd.lua
pdlua.bc
warp
```

5 glu.c

```
/*
Based on:
<https://gitlab.freedesktop.org/mesa/glu/blob/master/src/libutil/error.c>
<https://gitlab.freedesktop.org/mesa/glu/blob/master/src/libutil/project.c>
with minor modifications:
- use double throughout lookAt
- no nurbs/tess error strings
*/
/*
 * SGI FREE SOFTWARE LICENSE B (Version 2.0, Sept. 18, 2008)
 * Copyright (C) 1991-2000 Silicon Graphics, Inc. All Rights Reserved.
 *
 * Permission is hereby granted, free of charge, to any person obtaining a
 * copy of this software and associated documentation files (the "Software"),
 * to deal in the Software without restriction, including without limitation
 * the rights to use, copy, modify, merge, publish, distribute, sublicense,
 * and/or sell copies of the Software, and to permit persons to whom the
 * Software is furnished to do so, subject to the following conditions:
 *
 * The above copyright notice including the dates of first publication and
 * either this permission notice or a reference to
 * http://oss.sgi.com/projects/FreeB/
 * shall be included in all copies or substantial portions of the Software.
 *
 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
 * OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
 * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
 * SILICON GRAPHICS, INC. BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
 * WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF
 * OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
 * SOFTWARE.
 *
 * Except as contained in this notice, the name of Silicon Graphics, Inc.
 * shall not be used in advertising or otherwise to promote the sale, use or
```

```
 * other dealings in this Software without prior written authorization from
 * Silicon Graphics, Inc.
 */

#include <GL/Regal.h>
#include <GL/RegalGLU.h>

#include <math.h>

static void makeIdentity(GLdouble m[16])
{
    m[0+4*0] = 1; m[0+4*1] = 0; m[0+4*2] = 0; m[0+4*3] = 0;
    m[1+4*0] = 0; m[1+4*1] = 1; m[1+4*2] = 0; m[1+4*3] = 0;
    m[2+4*0] = 0; m[2+4*1] = 0; m[2+4*2] = 1; m[2+4*3] = 0;
    m[3+4*0] = 0; m[3+4*1] = 0; m[3+4*2] = 0; m[3+4*3] = 1;
}

static void normalize(double v[3])
{
    double r;

    r = sqrt( v[0]*v[0] + v[1]*v[1] + v[2]*v[2] );
    if (r == 0.0) return;

    v[0] /= r;
    v[1] /= r;
    v[2] /= r;
}

static void cross(double v1[3], double v2[3], double result[3])
{
    result[0] = v1[1]*v2[2] - v1[2]*v2[1];
    result[1] = v1[2]*v2[0] - v1[0]*v2[2];
    result[2] = v1[0]*v2[1] - v1[1]*v2[0];
}

extern void gluLookAt
  ( GLdouble eyex, GLdouble eyey, GLdouble eyez
  , GLdouble centerx, GLdouble centery, GLdouble centerz
  , GLdouble upx, GLdouble upy, GLdouble upz
  )
{
    double forward[3], side[3], up[3];
    GLdouble m[4][4];

    forward[0] = centerx - eyex;
    forward[1] = centery - eyey;
    forward[2] = centerz - eyez;

    up[0] = upx;
    up[1] = upy;
    up[2] = upz;

    normalize(forward);

    /* Side = forward x up */
    cross(forward, up, side);
```

```
    normalize(side);

    /* Recompute up as: up = side x forward */
    cross(side, forward, up);

    makeIdentity(&m[0][0]);
    m[0][0] = side[0];
    m[1][0] = side[1];
    m[2][0] = side[2];

    m[0][1] = up[0];
    m[1][1] = up[1];
    m[2][1] = up[2];

    m[0][2] = -forward[0];
    m[1][2] = -forward[1];
    m[2][2] = -forward[2];

    glMultMatrixd(&m[0][0]);
    glTranslated(-eyex, -eyey, -eyez);
}

extern const GLubyte *gluErrorString(GLenum err)
{
  switch (err)
  {
  case GL_NO_ERROR: return (const GLubyte *)"no error";
  case GL_INVALID_ENUM: return (const GLubyte *)"invalid enumerant";
  case GL_INVALID_VALUE: return (const GLubyte *)"invalid value";
  case GL_INVALID_OPERATION: return (const GLubyte *)"invalid operation";
  case GL_STACK_OVERFLOW: return (const GLubyte *)"stack overflow";
  case GL_STACK_UNDERFLOW: return (const GLubyte *)"stack underflow";
  case GL_OUT_OF_MEMORY: return (const GLubyte *)"out of memory";
  case GL_TABLE_TOO_LARGE: return (const GLubyte *)"table too large";
#ifdef GL_EXT_framebuffer_object
  case GL_INVALID_FRAMEBUFFER_OPERATION_EXT: return (const GLubyte *)"invalid ↙
      ↪ framebuffer operation";
#endif
  default: return 0;
  }
}
```

6 Makefile

```
LIBPD_DIR = ../../../
GLU_DIR = $(HOME)/opt

SRC_FILES = puzzle.c glu.c
TARGET = puzzle-em.html

CFLAGS = -I$(LIBPD_DIR)/pure-data/src -I$(LIBPD_DIR)/libpd_wrapper -O3
LDFLAGS = -L$(LIBPD_DIR)/libs -lpd -L$(GLU_DIR)/lib -lGLU Gem.bc pdlua.bc

$(TARGET): $(SRC_FILES) Makefile
	emcc $(CFLAGS) -o $(TARGET) $(SRC_FILES) \
	-DUSE_MGL_NAMESPACE=1 \
	-s USE_SDL=2 \
```

```
        -s USE_REGAL=1 \
        -s LEGACY_GL_EMULATION=0 \
        -s ERROR_ON_UNDEFINED_SYMBOLS=0 \
        --preload-file puzzle \
        --preload-file Gem \
        --preload-file pd.lua \
        $(LDFLAGS)
```

7 puzzle.c

```
#include <SDL2/SDL.h>
#include <SDL2/SDL_audio.h>

#include <emscripten.h>

#include <stdio.h>
#include "z_libpd.h"

extern void Gem_setup(void);
extern void pdlua_setup(void);

void audio(void *userdata, Uint8 *stream, int len)
{
  float inbuf[64][2], outbuf[64][2];
  float *b = (float *) stream;
  int m = len / sizeof(float) / 2;
  int k = 0;
  while (m > 0)
  {
    for (int i = 0; i < 64; ++i)
      for (int j = 0; j < 2; ++j)
        inbuf[i][j] = 0;
    libpd_process_float(1, &inbuf[0][0], &outbuf[0][0]);
    for (int i = 0; i < 64; ++i)
      for (int j = 0; j < 2; ++j)
        b[k++] = outbuf[i][j];
    m -= 64;
  }
  if (m < 0)
  {
    fprintf(stderr, "buffer overflow, m went negative: %d\n", m);
  }
}

void pdprint(const char *s) {
  printf("%s", s);
}

void main1(void)
{
  // nop
}

int main(int argc, char **argv)
{
  // initialize SDL2 audio
  SDL_Init(SDL_INIT_AUDIO);
```

```
    SDL_AudioSpec want, have;
    want.freq = 48000;
    want.format = AUDIO_F32;
    want.channels = 2;
    want.samples = 4096;
    want.callback = audio;
    SDL_AudioDeviceID dev = SDL_OpenAudioDevice(NULL, 0, &want, &have, ↙
        ↳ SDL_AUDIO_ALLOW_ANY_CHANGE);
    printf("AUDIO: want: %d %d %d %d\n", want.freq, want.format, want.channels, ↙
        ↳ want.samples);
    printf("AUDIO: have: %d %d %d %d\n", have.freq, have.format, have.channels, ↙
        ↳ have.samples);

    // initialize libpd
    libpd_set_printhook(pdprint);
    libpd_init();
    libpd_init_audio(2, 2, have.freq);

    // initialize Gem
    Gem_setup();
    pdlua_setup();

    // open patch
    libpd_openfile("PUZZLE3X3.pd", "puzzle");

    // startup message
    libpd_start_message(6);
    libpd_add_float(512); // size, must be power of 2
    libpd_add_float(512);
    libpd_add_float(0); // position, irrelevant for emscripten
    libpd_add_float(0);
    libpd_add_symbol(":0.0"); // display, irrelevant for emscripten
    libpd_add_float(0); // recording toggle
    libpd_finish_message("PUZZLE", "list");

    // start audio processing
    SDL_PauseAudioDevice(dev, 0);
    emscripten_set_main_loop(main1, 0, 1);
    return 0;
}

    // emscripten dl aborts; replace with graceful failure
    void *dlopen(const char *fn, int flags) { return 0; }
    int dlclose(void *handle) { return 0; }
    void *dlsym(void *h, const char *sym) { return 0; }
    char *dlerror(void) { return "not supported"; }
```

8 puzzle/drums/cm-ead~.pd

```
#N canvas 0 0 450 300 10;
#X obj 21 17 inlet;
#X obj 81 17 inlet;
#X obj 141 17 inlet;
#X obj 313 56 samplerate~;
#X obj 182 5 loadbang;
#X obj 182 32 t b b;
#X obj 141 68 f \$2;
```

```
#X obj 81 84 f \$1;
#X obj 97 201 vline~;
#X obj 66 157 f;
#X obj 126 159 f;
#X obj 81 137 clip 0 1;
#X obj 141 138 clip 0 1;
#X obj 44 237 rpole~;
#X obj 21 107 delay;
#X obj 21 202 vline~;
#X obj 21 155 f 0;
#X obj 44 262 outlet~;
#X obj 42 157 f 1;
#X obj 141 88 expr exp((1000*log(0.001))/($f1*$f2));
#X obj 81 118 expr exp((1000*log(0.001))/($f1*$f2));
#X obj 47 180 -;
#X obj 21 45 t b b b b b b;
#X connect 0 0 22 0;
#X connect 1 0 14 1;
#X connect 1 0 7 0;
#X connect 2 0 6 0;
#X connect 3 0 19 1;
#X connect 3 0 20 1;
#X connect 4 0 5 0;
#X connect 5 0 6 0;
#X connect 5 0 7 0;
#X connect 5 1 3 0;
#X connect 6 0 19 0;
#X connect 7 0 14 1;
#X connect 7 0 20 0;
#X connect 8 0 13 1;
#X connect 9 0 8 0;
#X connect 9 0 21 1;
#X connect 10 0 8 0;
#X connect 11 0 9 1;
#X connect 12 0 10 1;
#X connect 13 0 17 0;
#X connect 14 0 10 0;
#X connect 14 0 16 0;
#X connect 15 0 13 0;
#X connect 16 0 15 0;
#X connect 18 0 21 0;
#X connect 19 0 12 0;
#X connect 20 0 11 0;
#X connect 21 0 15 0;
#X connect 22 0 14 0;
#X connect 22 1 18 0;
#X connect 22 2 9 0;
#X connect 22 3 7 0;
#X connect 22 4 6 0;
#X connect 22 5 3 0;
```

9 puzzle/drums/cm-obi-ade-bell~.pd

```
#N canvas 0 0 380 504 10;
#X obj 155 103 wrap~;
#X obj 138 76 -~ 0.5;
#X obj 154 127 -~ 0.5;
```

```
#X obj 138 153 -~;
#X obj 172 278 wrap~;
#X obj 155 255 -~ 0.5;
#X obj 171 303 -~ 0.5;
#X obj 155 327 -~;
#X obj 138 350 *~;
#X obj 96 442 outlet~;
#X obj 154 232 phasor~;
#X obj 228 32 t b;
#X obj 218 7 inlet;
#X obj 96 386 *~;
#X obj 56 103 wrap~;
#X obj 39 76 -~ 0.5;
#X obj 55 127 -~ 0.5;
#X obj 39 153 -~;
#X obj 73 278 wrap~;
#X obj 56 255 -~ 0.5;
#X obj 72 303 -~ 0.5;
#X obj 56 327 -~;
#X obj 39 350 *~;
#X obj 55 232 phasor~;
#X obj 39 53 phasor~ 2139;
#X obj 138 53 phasor~ 441;
#X obj 60 29 *~ 1.33;
#X obj 154 179 *~ 1000;
#X obj 55 179 *~ 1000;
#X obj 95 407 *~ 0.25;
#X obj 101 10 sig~ 881;
#X obj 54 207 +~ 1763;
#X obj 153 207 +~ 884;
#X msg 268 15 bang;
#X obj 228 57 cm-ead~ 1 300;
#X connect 0 0 2 0;
#X connect 1 0 0 0;
#X connect 1 0 3 0;
#X connect 2 0 3 1;
#X connect 3 0 8 0;
#X connect 3 0 27 0;
#X connect 4 0 6 0;
#X connect 5 0 4 0;
#X connect 5 0 7 0;
#X connect 6 0 7 1;
#X connect 7 0 8 1;
#X connect 8 0 13 0;
#X connect 10 0 5 0;
#X connect 11 0 34 0;
#X connect 12 0 11 0;
#X connect 13 0 29 0;
#X connect 14 0 16 0;
#X connect 15 0 14 0;
#X connect 15 0 17 0;
#X connect 16 0 17 1;
#X connect 17 0 22 0;
#X connect 17 0 28 0;
#X connect 18 0 20 0;
#X connect 19 0 18 0;
#X connect 19 0 21 0;
```

```
#X connect 20 0 21 1;
#X connect 21 0 22 1;
#X connect 22 0 13 0;
#X connect 23 0 19 0;
#X connect 24 0 15 0;
#X connect 25 0 1 0;
#X connect 26 0 24 0;
#X connect 27 0 32 0;
#X connect 28 0 31 0;
#X connect 29 0 9 0;
#X connect 30 0 25 0;
#X connect 30 0 26 0;
#X connect 31 0 23 0;
#X connect 32 0 10 0;
#X connect 33 0 11 0;
#X connect 34 0 13 1;
```

10 puzzle/drums/cm-obi-ade-bongoh~.pd

```
#N canvas 0 0 450 300 10;
#X obj 143 93 noise~;
#X obj 194 120 clip~ -0.9 0.9;
#X obj 102 68 inlet;
#X obj 102 97 t b;
#X obj 187 185 *~;
#X obj 195 93 *~ 20;
#X obj 187 227 outlet~;
#X obj 194 144 bp~ 600 25;
#X obj 195 66 osc~ 660;
#X obj 186 205 *~ 0.7;
#X msg 51 66 bang;
#X obj 103 142 cm-ead~ 3 120;
#X connect 0 0 1 0;
#X connect 1 0 7 0;
#X connect 2 0 3 0;
#X connect 3 0 11 0;
#X connect 4 0 9 0;
#X connect 5 0 1 0;
#X connect 7 0 4 1;
#X connect 8 0 5 0;
#X connect 9 0 6 0;
#X connect 10 0 3 0;
#X connect 11 0 4 0;
```

11 puzzle/drums/cm-obi-ade-bongol~.pd

```
#N canvas 0 0 450 300 10;
#X obj 143 93 noise~;
#X obj 195 66 osc~ 420;
#X obj 194 120 clip~ -0.9 0.9;
#X obj 102 68 inlet;
#X obj 102 97 t b;
#X obj 187 185 *~;
#X obj 195 93 *~ 20;
#X obj 194 144 bp~ 400 25;
#X obj 187 227 outlet~;
```

```
#X obj 187 206 *~ 0.6;
#X obj 103 142 cm-ead~ 3 150;
#X connect 0 0 2 0;
#X connect 1 0 6 0;
#X connect 2 0 7 0;
#X connect 3 0 4 0;
#X connect 4 0 10 0;
#X connect 5 0 9 0;
#X connect 6 0 2 0;
#X connect 7 0 5 1;
#X connect 9 0 8 0;
#X connect 10 0 5 0;
```

12 puzzle/drums/cm-obi-ade-clap~.pd

```
#N canvas 0 0 545 467 10;
#X obj 189 66 noise~;
#X obj 336 325 *~;
#X obj 151 310 *~;
#X obj 167 250 *~ 0.8;
#X obj 148 343 *~ 0.2;
#X obj 376 145 del 30;
#X obj 377 165 del 20;
#X obj 378 185 del 15;
#X obj 376 206 del 10;
#X obj 63 217 *~ 0.1;
#X obj 85 147 bp~ 1300 10;
#X obj 253 147 bp~ 700 10;
#X obj 175 147 bp~ 2500 4;
#X obj 336 369 bp~ 2700 2;
#X obj 3 147 lop~ 8000;
#X obj 188 106 hip~ 300;
#X obj 322 11 inlet;
#X obj 322 40 t b;
#X obj 339 415 outlet~;
#X obj 339 390 *~ 0.7;
#X msg 278 11 bang;
#X obj 336 292 cm-ead~ 3 50;
#X obj 179 284 cm-ead~ 1 800;
#X connect 0 0 14 0;
#X connect 0 0 15 0;
#X connect 1 0 13 0;
#X connect 2 0 1 1;
#X connect 2 0 4 0;
#X connect 3 0 2 1;
#X connect 4 0 13 0;
#X connect 5 0 6 0;
#X connect 5 0 21 0;
#X connect 6 0 7 0;
#X connect 6 0 21 0;
#X connect 7 0 8 0;
#X connect 7 0 21 0;
#X connect 8 0 21 0;
#X connect 9 0 3 0;
#X connect 10 0 3 0;
#X connect 11 0 3 0;
#X connect 12 0 3 0;
```

```
#X connect 13 0 19 0;
#X connect 14 0 9 0;
#X connect 15 0 10 0;
#X connect 15 0 11 0;
#X connect 15 0 12 0;
#X connect 16 0 17 0;
#X connect 17 0 5 0;
#X connect 17 0 21 0;
#X connect 17 0 22 0;
#X connect 19 0 18 0;
#X connect 20 0 17 0;
#X connect 21 0 1 0;
#X connect 22 0 2 0;
```

13 puzzle/drums/cm-obi-ade-cowbell~.pd

```
#N canvas 0 0 588 492 10;
#X obj 121 192 osc~ 440;
#X obj 189 190 osc~ 440;
#X obj 254 189 osc~ 440;
#X obj 322 187 osc~ 440;
#X obj 122 219 *~;
#X obj 122 241 *~;
#X obj 123 263 *~;
#X obj 124 290 cos~;
#X obj 125 315 *~;
#X obj 172 287 *~ 0.3;
#X obj 39 290 *~;
#X obj 121 133 * 2.666;
#X obj 189 132 * 2.333;
#X obj 254 132 * 0.875;
#X obj 322 132 * 0.666;
#X msg 205 100 0;
#X obj 190 55 t b f;
#X obj 124 340 *~ 0.2;
#X msg 190 28 680;
#X obj 121 162 +;
#X obj 322 159 +;
#X obj 254 159 +;
#X obj 189 160 +;
#X obj 264 79 random 100;
#X obj 264 100 / 50;
#X obj 339 79 random 100;
#X obj 414 78 random 100;
#X obj 339 100 / 40;
#X obj 414 99 / 30;
#X obj 124 365 outlet~;
#X obj 149 28 inlet;
#X obj 173 264 cm-ead~ 1 400;
#X obj 38 264 cm-ead~ 1 900;
#X connect 0 0 4 0;
#X connect 1 0 4 1;
#X connect 2 0 5 1;
#X connect 3 0 6 1;
#X connect 4 0 5 0;
#X connect 5 0 6 0;
#X connect 6 0 7 0;
```

```
#X connect 6 0 10 1;
#X connect 7 0 8 0;
#X connect 8 0 17 0;
#X connect 9 0 8 1;
#X connect 10 0 17 0;
#X connect 11 0 19 0;
#X connect 12 0 22 0;
#X connect 13 0 21 0;
#X connect 14 0 20 0;
#X connect 15 0 0 1;
#X connect 15 0 1 1;
#X connect 15 0 2 1;
#X connect 15 0 3 1;
#X connect 16 0 15 0;
#X connect 16 0 23 0;
#X connect 16 0 25 0;
#X connect 16 0 26 0;
#X connect 16 0 31 0;
#X connect 16 0 32 0;
#X connect 16 1 11 0;
#X connect 16 1 12 0;
#X connect 16 1 13 0;
#X connect 16 1 14 0;
#X connect 17 0 29 0;
#X connect 18 0 16 0;
#X connect 19 0 0 0;
#X connect 20 0 3 0;
#X connect 21 0 2 0;
#X connect 22 0 1 0;
#X connect 23 0 24 0;
#X connect 24 0 22 1;
#X connect 25 0 27 0;
#X connect 26 0 28 0;
#X connect 27 0 21 1;
#X connect 28 0 20 1;
#X connect 30 0 16 0;
#X connect 31 0 9 0;
#X connect 32 0 10 0;
```

14 puzzle/drums/cm-obi-ade-crash~.pd

```
#N canvas 0 0 615 516 10;
#X obj 415 47 inlet;
#X obj 415 72 t b;
#X obj 205 454 outlet~;
#X obj 83 181 *~;
#X obj 230 183 *~;
#X obj 150 213 *~;
#X obj 139 265 +~;
#X obj 138 288 osc~ 5435;
#X obj 78 238 sig~ 3435;
#X obj 150 236 *~ 12000;
#X obj 248 142 osc~ 2441;
#X obj 228 235 noise~;
#X obj 180 354 *~;
#X obj 235 352 *~;
#X obj 23 144 osc~ 12221;
```

```
#X obj 100 144 osc~ 8309;
#X obj 172 143 osc~ 2383;
#X obj 228 278 clip~ -0.9 0.9;
#X obj 347 229 noise~;
#X obj 347 272 clip~ -0.9 0.9;
#X obj 228 256 bp~ 8000 4;
#X obj 142 319 *~;
#X obj 245 373 *~ 0.2;
#X obj 320 349 *~;
#X obj 177 374 *~ 0.2;
#X obj 348 377 *~ 0.4;
#X obj 346 250 bp~ 4000 7;
#X obj 204 433 *~ 0.8;
#X msg 360 39 bang;
#X obj 307 99 cm-ead~ 4000 3000;
#X obj 414 100 cm-ead~ 2 5000;
#X obj 432 128 cm-ead~ 300 7000;
#X obj 529 97 cm-ead~ 1 200;
#X connect 0 0 1 0;
#X connect 1 0 29 0;
#X connect 1 0 30 0;
#X connect 1 0 31 0;
#X connect 1 0 32 0;
#X connect 3 0 5 0;
#X connect 4 0 5 1;
#X connect 5 0 9 0;
#X connect 6 0 7 0;
#X connect 7 0 21 0;
#X connect 8 0 6 0;
#X connect 9 0 6 1;
#X connect 10 0 4 1;
#X connect 11 0 20 0;
#X connect 12 0 24 0;
#X connect 13 0 22 0;
#X connect 14 0 3 0;
#X connect 15 0 3 1;
#X connect 16 0 4 0;
#X connect 17 0 12 0;
#X connect 18 0 26 0;
#X connect 19 0 13 0;
#X connect 19 0 23 0;
#X connect 20 0 17 0;
#X connect 21 0 12 0;
#X connect 21 0 23 0;
#X connect 22 0 27 0;
#X connect 23 0 25 0;
#X connect 24 0 27 0;
#X connect 25 0 27 0;
#X connect 26 0 19 0;
#X connect 27 0 2 0;
#X connect 28 0 1 0;
#X connect 29 0 21 1;
#X connect 30 0 12 1;
#X connect 31 0 13 1;
#X connect 32 0 23 1;
```

15 puzzle/drums/cm-obi-ade-cymbal~.pd

```
#N canvas 0 0 386 454 10;
#X obj 140 26 inlet;
#X obj 140 51 t b;
#X obj 202 400 outlet~;
#X obj 83 181 *~;
#X obj 230 183 *~;
#X obj 150 213 *~;
#X obj 139 265 +~;
#X obj 150 300 osc~ 5435;
#X obj 78 238 sig~ 3435;
#X obj 150 236 *~ 12000;
#X obj 23 144 osc~ 6221;
#X obj 98 144 osc~ 4409;
#X obj 172 143 osc~ 3383;
#X obj 248 142 osc~ 2441;
#X obj 228 235 noise~;
#X obj 180 328 +~;
#X obj 228 256 bp~ 9000 12;
#X obj 230 299 clip~ -0.7 0.7;
#X obj 180 354 *~;
#X obj 235 352 *~;
#X obj 176 374 *~ 0.05;
#X obj 245 373 *~ 0.05;
#X msg 102 25 bang;
#X obj 139 79 cm-ead~ 2 8000;
#X obj 157 107 cm-ead~ 2 100;
#X connect 0 0 1 0;
#X connect 1 0 23 0;
#X connect 1 0 24 0;
#X connect 3 0 5 0;
#X connect 4 0 5 1;
#X connect 5 0 9 0;
#X connect 6 0 7 0;
#X connect 7 0 15 0;
#X connect 8 0 6 0;
#X connect 9 0 6 1;
#X connect 10 0 3 0;
#X connect 11 0 3 1;
#X connect 12 0 4 0;
#X connect 13 0 4 1;
#X connect 14 0 16 0;
#X connect 15 0 18 0;
#X connect 15 0 19 0;
#X connect 16 0 17 0;
#X connect 17 0 15 1;
#X connect 18 0 20 0;
#X connect 19 0 21 0;
#X connect 20 0 2 0;
#X connect 21 0 2 0;
#X connect 22 0 1 0;
#X connect 23 0 18 1;
#X connect 24 0 19 1;
```

16 puzzle/drums/cm-obi-ade-drumkit~.pd

```
#N canvas 489 116 450 385 10;
#X obj 17 13 inlet;
#X obj 17 38 sel 0 1 2 3 4 5 6 7 8 9 10 11 12 13;
#X obj 17 69 cm-obi-ade-kick~;
#X obj 27 89 cm-obi-ade-snare~;
#X obj 37 109 cm-obi-ade-clap~;
#X obj 47 129 cm-obi-ade-bell~;
#X obj 57 149 cm-obi-ade-bongoh~;
#X obj 67 169 cm-obi-ade-bongol~;
#X obj 77 209 cm-obi-ade-handdrum~;
#X obj 87 249 cm-obi-ade-cowbell~;
#X obj 97 269 cm-obi-ade-hihat~;
#X obj 107 289 cm-obi-ade-cymbal~;
#X obj 117 309 cm-obi-ade-crash~;
#X obj 18 355 outlet~;
#X msg 77 189 82.47;
#X msg 194 189 82.47;
#X msg 87 228 660;
#X msg 117 229 440;
#X connect 0 0 1 0;
#X connect 1 0 2 0;
#X connect 1 1 3 0;
#X connect 1 2 4 0;
#X connect 1 3 5 0;
#X connect 1 4 6 0;
#X connect 1 5 7 0;
#X connect 1 6 15 0;
#X connect 1 7 14 0;
#X connect 1 8 16 0;
#X connect 1 9 17 0;
#X connect 1 10 10 1;
#X connect 1 11 10 0;
#X connect 1 12 11 0;
#X connect 1 13 12 0;
#X connect 2 0 13 0;
#X connect 3 0 13 0;
#X connect 4 0 13 0;
#X connect 5 0 13 0;
#X connect 6 0 13 0;
#X connect 7 0 13 0;
#X connect 8 0 13 0;
#X connect 9 0 13 0;
#X connect 10 0 13 0;
#X connect 11 0 13 0;
#X connect 12 0 13 0;
#X connect 14 0 8 0;
#X connect 15 0 8 1;
#X connect 16 0 9 0;
#X connect 17 0 9 0;
```

17 puzzle/drums/cm-obi-ade-handdrum~.pd

```
#N canvas 0 0 610 531 10;
#X obj 121 192 osc~ 440;
#X obj 189 190 osc~ 440;
#X obj 254 189 osc~ 440;
#X obj 322 187 osc~ 440;
```

```
#X obj 122 219 *~;
#X obj 122 241 *~;
#X obj 123 263 *~;
#X obj 124 311 cos~;
#X obj 125 344 *~;
#X obj 39 290 *~;
#X msg 205 100 0;
#X obj 121 49 t b f;
#X obj 126 435 outlet~;
#X obj 120 4 inlet;
#X obj 121 133 * 0.8;
#X obj 189 132 * 1.6666;
#X obj 254 132 * 0.833;
#X obj 123 287 clip~ -1 1;
#X obj 121 165 + 7.88;
#X obj 189 163 + 8.66;
#X obj 322 161 + 7.88;
#X obj 254 162 + 4.725;
#X obj 126 410 *~ 0.4;
#X obj 205 286 *~ 0.15;
#X obj 81 370 clip~ -0.7 0.5;
#X msg 449 87 2.66666;
#X msg 388 87 1.14141;
#X obj 322 54 t b f;
#X obj 322 4 inlet;
#X obj 322 133 *;
#X obj 206 263 cm-ead~ 1 400;
#X obj 37 264 cm-ead~ 1 500;
#X connect 0 0 4 0;
#X connect 1 0 4 1;
#X connect 2 0 5 1;
#X connect 3 0 6 1;
#X connect 4 0 5 0;
#X connect 5 0 6 0;
#X connect 6 0 9 1;
#X connect 6 0 17 0;
#X connect 7 0 8 0;
#X connect 8 0 22 0;
#X connect 9 0 24 0;
#X connect 10 0 0 1;
#X connect 10 0 1 1;
#X connect 10 0 2 1;
#X connect 10 0 3 1;
#X connect 11 0 10 0;
#X connect 11 0 30 0;
#X connect 11 0 31 0;
#X connect 11 1 26 0;
#X connect 11 1 29 0;
#X connect 11 1 16 0;
#X connect 11 1 15 0;
#X connect 11 1 14 0;
#X connect 13 0 11 0;
#X connect 14 0 18 0;
#X connect 15 0 19 0;
#X connect 16 0 21 0;
#X connect 17 0 7 0;
#X connect 18 0 0 0;
```

```
#X connect 19 0 1 0;
#X connect 20 0 3 0;
#X connect 21 0 2 0;
#X connect 22 0 12 0;
#X connect 23 0 8 1;
#X connect 24 0 22 0;
#X connect 25 0 29 1;
#X connect 26 0 29 1;
#X connect 27 0 10 0;
#X connect 27 0 30 0;
#X connect 27 0 31 0;
#X connect 27 1 25 0;
#X connect 27 1 29 0;
#X connect 27 1 16 0;
#X connect 27 1 15 0;
#X connect 27 1 14 0;
#X connect 28 0 27 0;
#X connect 29 0 20 0;
#X connect 30 0 23 0;
#X connect 31 0 9 0;
```

18 puzzle/drums/cm-obi-ade-hihat~.pd

```
#N canvas 0 0 459 480 10;
#X obj 140 26 inlet;
#X obj 140 51 t b;
#X obj 202 400 outlet~;
#X obj 83 181 *~;
#X obj 230 183 *~;
#X obj 150 213 *~;
#X obj 139 265 +~;
#X obj 150 300 osc~ 5435;
#X obj 150 236 *~ 12000;
#X obj 23 144 osc~ 6221;
#X obj 248 142 osc~ 2441;
#X obj 228 235 noise~;
#X obj 180 328 +~;
#X obj 230 299 clip~ -0.7 0.7;
#X obj 180 354 *~;
#X obj 235 352 *~;
#X obj 245 373 *~ 0.06;
#X obj 177 374 *~ 0.06;
#X obj 78 238 sig~ 7435;
#X obj 228 256 bp~ 12000 12;
#X obj 172 143 osc~ 3283;
#X obj 98 144 osc~ 4109;
#X obj 245 101 sqrt~;
#X obj 226 25 inlet;
#X obj 227 51 t b;
#X obj 246 77 cm-ead~ 2 800;
#X obj 139 79 cm-ead~ 2 200;
#X obj 157 107 cm-ead~ 2 100;
#X connect 0 0 1 0;
#X connect 1 0 26 0;
#X connect 1 0 27 0;
#X connect 3 0 5 0;
#X connect 4 0 5 1;
```

```
#X connect 5 0 8 0;
#X connect 6 0 7 0;
#X connect 7 0 12 0;
#X connect 8 0 6 1;
#X connect 9 0 3 0;
#X connect 10 0 4 1;
#X connect 11 0 19 0;
#X connect 12 0 14 0;
#X connect 12 0 15 0;
#X connect 13 0 12 1;
#X connect 14 0 17 0;
#X connect 15 0 16 0;
#X connect 16 0 2 0;
#X connect 17 0 2 0;
#X connect 18 0 6 0;
#X connect 19 0 13 0;
#X connect 20 0 4 0;
#X connect 21 0 3 1;
#X connect 22 0 14 1;
#X connect 23 0 24 0;
#X connect 24 0 25 0;
#X connect 24 0 27 0;
#X connect 25 0 22 0;
#X connect 26 0 14 1;
#X connect 27 0 15 1;
```

19 puzzle/drums/cm-obi-ade-kick~.pd

```
#N canvas 0 0 325 345 10;
#X obj 36 197 osc~ 200;
#X obj 36 228 *~;
#X obj 102 198 sqrt~;
#X obj 35 140 *~ 15;
#X obj 35 164 +~ 70;
#X obj 35 249 *~ 0.5;
#X msg 100 150 0;
#X obj 36 113 *~;
#X obj 36 19 inlet;
#X obj 36 44 t b;
#X obj 34 271 outlet~;
#X obj 36 72 cm-ead~ 2 800;
#X connect 0 0 1 0;
#X connect 1 0 5 0;
#X connect 2 0 1 1;
#X connect 3 0 4 0;
#X connect 4 0 0 0;
#X connect 5 0 10 0;
#X connect 6 0 0 1;
#X connect 7 0 3 0;
#X connect 8 0 9 0;
#X connect 9 0 6 0;
#X connect 9 0 11 0;
#X connect 11 0 2 0;
#X connect 11 0 7 0;
#X connect 11 0 7 1;
```

20 puzzle/drums/cm-obi-ade-snare~.pd

```
#N canvas 0 0 345 478 10;
#X obj 112 437 outlet~;
#X obj 39 8 inlet;
#X obj 146 74 noise~;
#X obj 38 29 t b;
#X obj 34 158 osc~;
#X obj 100 280 *~;
#X obj 128 147 *~;
#X obj 113 304 *~;
#X obj 170 272 *~;
#X obj 14 80 *~;
#X obj 68 302 *~;
#X obj 14 105 *~ 12;
#X msg 97 7 bang;
#X obj 107 334 clip~ 0 1;
#X obj 34 181 clip~ -0.8 0.8;
#X obj 166 296 *~ 0.1;
#X obj 108 355 *~ 0.4;
#X obj 54 339 *~ 0.9;
#X obj 146 98 bp~ 3300 0.5;
#X obj 146 122 lop~ 7000;
#X obj 112 408 bp~ 8000 1;
#X obj 73 243 clip~ -0.9 0.9;
#X obj 71 221 *~ 10000;
#X obj 15 134 +~ 150;
#X obj 113 379 *~ 0.35;
#X obj 40 50 cm-ead~ 1 300;
#X connect 1 0 3 0;
#X connect 2 0 18 0;
#X connect 3 0 25 0;
#X connect 4 0 14 0;
#X connect 5 0 7 0;
#X connect 6 0 7 1;
#X connect 7 0 13 0;
#X connect 8 0 15 0;
#X connect 9 0 10 1;
#X connect 9 0 11 0;
#X connect 10 0 17 0;
#X connect 11 0 23 0;
#X connect 12 0 3 0;
#X connect 13 0 16 0;
#X connect 14 0 10 0;
#X connect 14 0 22 0;
#X connect 15 0 24 0;
#X connect 16 0 24 0;
#X connect 17 0 24 0;
#X connect 18 0 8 0;
#X connect 18 0 19 0;
#X connect 19 0 6 1;
#X connect 20 0 0 0;
#X connect 21 0 5 0;
#X connect 22 0 21 0;
#X connect 23 0 4 0;
#X connect 24 0 20 0;
#X connect 25 0 5 1;
```

```
#X connect 25 0 6 0;
#X connect 25 0 9 0;
#X connect 25 0 9 1;
#X connect 25 0 8 1;
```

21 puzzle/puzzle3x3-comb~.pd

```
#N canvas 3 61 450 300 10;
#X obj 187 70 *~ -1;
#X obj 63 187 +~;
#X obj 143 174 +~;
#X obj 39 29 inlet~;
#X obj 242 28 inlet~;
#X obj 80 261 outlet~;
#X obj 141 260 outlet~;
#X obj 95 218 delwrite~ \$0-pp-R 30;
#X obj 33 239 delwrite~ \$0-pp-L 30;
#X obj 179 28 osc~ 8;
#X obj 81 115 vd~ \$0-pp-R;
#X obj 244 116 vd~ \$0-pp-L;
#X obj 91 7 phasor~ 0.03125;
#X obj 121 29 +~ 0.25;
#X obj 121 51 cos~;
#X obj 87 49 cos~;
#X obj 99 70 sig~ 8;
#X obj 179 50 *~ 0.1;
#X obj 242 154 expr~ -tanh($v1)*0.95;
#X obj 80 153 expr~ tanh($v1)*0.95;
#X obj 81 134 lop~ 4000;
#X obj 243 136 lop~ 4000;
#X obj 96 197 hip~ 1;
#X obj 33 212 hip~ 1;
#X connect 0 0 11 0;
#X connect 1 0 5 0;
#X connect 1 0 23 0;
#X connect 2 0 6 0;
#X connect 2 0 22 0;
#X connect 3 0 1 0;
#X connect 4 0 2 0;
#X connect 9 0 17 0;
#X connect 10 0 20 0;
#X connect 11 0 21 0;
#X connect 12 0 13 0;
#X connect 12 0 15 0;
#X connect 13 0 14 0;
#X connect 14 0 11 0;
#X connect 15 0 10 0;
#X connect 16 0 10 0;
#X connect 16 0 11 0;
#X connect 17 0 0 0;
#X connect 17 0 10 0;
#X connect 18 0 2 1;
#X connect 19 0 1 1;
#X connect 20 0 19 0;
#X connect 21 0 18 0;
#X connect 22 0 7 0;
#X connect 23 0 8 0;
```

22 puzzle/puzzle3x3-compress~.pd

```
#N canvas 95 278 452 587 10;
#X obj 28 28 inlet~;
#X obj 176 20 inlet~;
#X obj 27 543 outlet~;
#X obj 177 544 outlet~;
#X obj 58 71 hip~ 5;
#X obj 121 71 hip~ 5;
#X obj 58 97 *~;
#X obj 120 98 *~;
#X obj 87 158 +~;
#X obj 281 25 inlet;
#X obj 87 281 rmstodb~;
#X obj 86 353 dbtorms~;
#X obj 27 474 *~;
#X obj 177 474 *~;
#X obj 120 123 lop~ 10;
#X obj 57 124 lop~ 10;
#X obj 87 221 lop~ 25;
#X obj 25 272 /~;
#X obj 178 270 /~;
#X obj 201 390 dbtorms;
#X obj 86 417 /~ 0;
#X obj 87 179 sqrt~;
#X obj 86 442 *~ 0.25;
#X obj 310 55 loadbang;
#X obj 310 79 f \$1;
#X obj 27 514 expr~ tanh($v1);
#X obj 177 514 expr~ tanh($v1);
#X obj 86 314 expr~ if($v1>$f2 \, $f2+($v1-$f2)/8 \, $f2);
#X obj 201 366 expr (100-$f1)/8+$f1;
#X obj 88 202 +~ 0.01;
#X connect 0 0 4 0;
#X connect 0 0 17 0;
#X connect 1 0 5 0;
#X connect 1 0 18 0;
#X connect 4 0 6 0;
#X connect 4 0 6 1;
#X connect 5 0 7 0;
#X connect 5 0 7 1;
#X connect 6 0 15 0;
#X connect 7 0 14 0;
#X connect 8 0 21 0;
#X connect 9 0 27 1;
#X connect 9 0 28 0;
#X connect 10 0 27 0;
#X connect 11 0 20 0;
#X connect 12 0 25 0;
#X connect 13 0 26 0;
#X connect 14 0 8 1;
#X connect 15 0 8 0;
#X connect 16 0 17 1;
#X connect 16 0 18 1;
#X connect 16 0 10 0;
#X connect 17 0 12 0;
#X connect 18 0 13 0;
```

```
#X connect 19 0 20 1;
#X connect 20 0 22 0;
#X connect 21 0 29 0;
#X connect 22 0 12 1;
#X connect 22 0 13 1;
#X connect 23 0 24 0;
#X connect 24 0 27 1;
#X connect 24 0 28 0;
#X connect 25 0 2 0;
#X connect 26 0 3 0;
#X connect 27 0 11 0;
#X connect 28 0 19 0;
#X connect 29 0 16 0;
```

23 puzzle/puzzle3x3-delay~.pd

```
#N canvas 3 61 495 476 10;
#X obj 116 172 mtof~;
#X obj 104 371 delwrite~ \$0-dub-L 3000;
#X obj 262 375 delwrite~ \$0-dub-R 3000;
#X obj 225 151 vd~ \$0-dub-L;
#X obj 26 149 vd~ \$0-dub-R;
#X obj 166 172 mtof~;
#X obj 27 80 cos~;
#X obj 67 80 cos~;
#X obj 62 54 +~ 0.25;
#X obj 26 104 *~ 12;
#X obj 77 104 *~ 12;
#X obj 202 256 expr~ tanh($v1*3);
#X obj 201 209 expr~ tanh($v1*3);
#X obj 53 210 expr~ tanh($v1*3);
#X obj 54 257 expr~ tanh($v1*3);
#X obj 27 29 phasor~ 0.125;
#X obj 137 80 cos~;
#X obj 177 80 cos~;
#X obj 172 54 +~ 0.25;
#X obj 136 104 *~ 250;
#X obj 187 104 *~ -250;
#X obj 137 29 phasor~ 0.03125;
#X obj 57 409 outlet~;
#X obj 233 412 outlet~;
#X obj 338 251 inlet~;
#X obj 399 251 inlet~;
#X obj 116 149 +~ 79;
#X obj 166 149 +~ 79;
#X obj 262 107 sig~ 333;
#X obj 202 231 vcf~ 5;
#X obj 54 232 vcf~ 5;
#X obj 54 282 vcf~ 5;
#X obj 202 284 vcf~ 5;
#X obj 109 326 puzzle3x3-compress~ 80;
#X obj 225 173 *~ 1;
#X obj 23 171 *~ 1;
#X obj 336 274 *~ 2;
#X obj 398 276 *~ 2;
#X connect 0 0 30 1;
#X connect 0 0 32 1;
```

```
#X connect 3 0 34 0;
#X connect 4 0 35 0;
#X connect 5 0 29 1;
#X connect 5 0 31 1;
#X connect 6 0 9 0;
#X connect 7 0 10 0;
#X connect 8 0 7 0;
#X connect 9 0 26 0;
#X connect 10 0 27 0;
#X connect 11 0 32 0;
#X connect 12 0 29 0;
#X connect 13 0 30 0;
#X connect 14 0 31 0;
#X connect 15 0 6 0;
#X connect 15 0 8 0;
#X connect 16 0 19 0;
#X connect 17 0 20 0;
#X connect 18 0 17 0;
#X connect 19 0 4 0;
#X connect 20 0 3 0;
#X connect 21 0 16 0;
#X connect 21 0 18 0;
#X connect 24 0 36 0;
#X connect 25 0 37 0;
#X connect 26 0 0 0;
#X connect 27 0 5 0;
#X connect 28 0 3 0;
#X connect 28 0 4 0;
#X connect 29 0 11 0;
#X connect 30 0 14 0;
#X connect 31 0 33 0;
#X connect 32 0 33 1;
#X connect 33 0 1 0;
#X connect 33 0 22 0;
#X connect 33 1 2 0;
#X connect 33 1 23 0;
#X connect 34 0 12 0;
#X connect 35 0 13 0;
#X connect 36 0 33 0;
#X connect 37 0 33 1;
```

24 puzzle/puzzle3x3-drumkit~.pd

```
#N canvas 3 293 1056 784 10;
#X obj 8 403 drums/cm-obi-ade-kick~;
#X obj 8 425 throw~ \$1-0-src;
#X obj 124 148 f 0;
#X obj 196 133 + 1;
#X obj 195 159 mod 16;
#X obj 125 8 r pd;
#X obj 125 30 route dsp;
#X obj 250 403 drums/cm-obi-ade-snare~;
#X obj 250 425 throw~ \$1-1-src;
#X obj 508 408 drums/cm-obi-ade-hihat~;
#X obj 508 430 throw~ \$1-2-src;
#X obj 747 408 drums/cm-obi-ade-handdrum~;
#X msg 748 385 84;
```

```
15  #X msg 899 384 84;
    #X obj 747 430 throw~ \$1-3-src;
    #X obj 7 607 throw~ \$1-4-src;
    #X obj 27 587 drums/cm-obi-ade-bongoh~;
    #X obj 7 567 drums/cm-obi-ade-bongol~;
20  #X obj 258 591 drums/cm-obi-ade-cowbell~;
    #X msg 256 568 660;
    #X obj 258 613 throw~ \$1-5-src;
    #X msg 286 566 880;
    #X obj 513 579 drums/cm-obi-ade-clap~;
25  #X obj 513 607 throw~ \$1-6-src;
    #X obj 753 572 drums/cm-obi-ade-bell~;
    #X obj 753 594 throw~ \$1-7-src;
    #X obj 124 56 metro 250;
    #X obj 150 83 delay 144;
30  #X msg 10 449 0 1 0 0 0 1 0 0 0 1 0 0 1 0 0 0 0;
    #X obj 8 379 sel 1;
    #X msg 744 452 0 1 0 0 2 0 1 0 0 0 2 0 0 0 0 0 0;
    #X obj 749 361 sel 1 2;
    #X msg 250 449 0 0 0 1 0 0 0 1 0 0 0 0 0 1 0 0 1;
35  #X obj 250 380 sel 1;
    #X msg 515 628 0 0 0 0 0 0 1 0 0 0 0 1 0 0 1 0 0 0;
    #X obj 513 554 sel 1;
    #X obj 508 381 sel 1 2;
    #X msg 15 634 0 0 0 0 0 0 0 1 0 0 1 0 0 1 2 2 2;
40  #X obj 9 545 sel 1 2;
    #X msg 254 637 0 1 0 0 0 1 2 0 2 0 0 0 1 0 0 0 0;
    #X obj 256 543 sel 1 2;
    #X msg 508 450 0 0 1 0 0 0 0 0 1 0 0 2 0 0 1 0 0;
    #X msg 754 620 0 0 1 0 1 0 0 1 0 1 0 0 1 0 0 0 0;
45  #X obj 752 549 sel 1;
    #X obj 398 167 loadbang;
    #X obj 280 15 inlet;
    #X obj 9 471 s \$0-0-tab;
    #X obj 9 356 tabread \$0-0-tab;
50  #X obj 10 491 table \$0-0-tab 16;
    #X obj 14 655 s \$0-4-tab;
    #X obj 15 680 table \$0-4-tab 16;
    #X obj 9 522 tabread \$0-4-tab;
    #X obj 250 357 tabread \$0-1-tab;
55  #X obj 250 472 s \$0-1-tab;
    #X obj 250 495 table \$0-1-tab 16;
    #X obj 256 519 tabread \$0-5-tab;
    #X obj 254 660 s \$0-5-tab;
    #X obj 254 683 table \$0-5-tab 16;
60  #X obj 508 358 tabread \$0-2-tab;
    #X obj 508 496 table \$0-2-tab 16;
    #X obj 508 473 s \$0-2-tab;
    #X obj 749 339 tabread \$0-3-tab;
    #X obj 743 473 s \$0-3-tab;
65  #X obj 743 496 table \$0-3-tab 16;
    #X obj 752 526 tabread \$0-7-tab;
    #X obj 754 643 s \$0-7-tab;
    #X obj 754 666 table \$0-7-tab 16;
    #X obj 515 674 table \$0-6-tab 16;
70  #X obj 513 531 tabread \$0-6-tab;
    #X obj 515 651 s \$0-6-tab;
```

```
#X obj 279 63 f \$0;
#X obj 279 86 pack f f;
#X obj 279 134 puzzle3x3-tswap;
#X obj 279 37 t b f;
#X msg 279 111 symbol \$1-\$2-tab;
#X obj 410 36 loadbang;
#X msg 410 59 0 \, 1 \, 2 \, 3 \, 4 \, 5 \, 6 \, 7;
#X msg 410 82 \$1 \, \$1 \, \$1 \, \$1 \, \$1 \, \$1 \, \$1 \, \$1
;
#X msg 410 102 \$1 \, \$1 \, \$1 \, \$1 \, \$1 \, \$1 \, \$1 \, \$1
;
#X connect 0 0 1 0;
#X connect 2 0 3 0;
#X connect 2 0 47 0;
#X connect 2 0 51 0;
#X connect 2 0 52 0;
#X connect 2 0 55 0;
#X connect 2 0 58 0;
#X connect 2 0 61 0;
#X connect 2 0 64 0;
#X connect 2 0 68 0;
#X connect 3 0 4 0;
#X connect 4 0 2 1;
#X connect 5 0 6 0;
#X connect 6 0 26 0;
#X connect 7 0 8 0;
#X connect 9 0 10 0;
#X connect 11 0 14 0;
#X connect 12 0 11 0;
#X connect 13 0 11 1;
#X connect 16 0 15 0;
#X connect 17 0 15 0;
#X connect 18 0 20 0;
#X connect 19 0 18 0;
#X connect 21 0 18 0;
#X connect 22 0 23 0;
#X connect 24 0 25 0;
#X connect 26 0 2 0;
#X connect 26 0 27 0;
#X connect 27 0 2 0;
#X connect 28 0 46 0;
#X connect 29 0 0 0;
#X connect 30 0 62 0;
#X connect 31 0 12 0;
#X connect 31 1 13 0;
#X connect 32 0 53 0;
#X connect 33 0 7 0;
#X connect 34 0 69 0;
#X connect 35 0 22 0;
#X connect 36 0 9 0;
#X connect 36 1 9 1;
#X connect 37 0 49 0;
#X connect 38 0 17 0;
#X connect 38 1 16 0;
#X connect 39 0 56 0;
#X connect 40 0 19 0;
#X connect 40 1 21 0;
```

```
#X connect 41 0 60 0;
#X connect 42 0 65 0;
#X connect 43 0 24 0;
#X connect 44 0 28 0;
#X connect 44 0 32 0;
#X connect 44 0 37 0;
#X connect 44 0 39 0;
#X connect 44 0 41 0;
#X connect 44 0 34 0;
#X connect 44 0 30 0;
#X connect 44 0 42 0;
#X connect 45 0 73 0;
#X connect 47 0 29 0;
#X connect 51 0 38 0;
#X connect 52 0 33 0;
#X connect 55 0 40 0;
#X connect 58 0 36 0;
#X connect 61 0 31 0;
#X connect 64 0 43 0;
#X connect 68 0 35 0;
#X connect 70 0 71 0;
#X connect 71 0 74 0;
#X connect 73 0 70 0;
#X connect 73 1 71 1;
#X connect 74 0 72 0;
#X connect 75 0 76 0;
#X connect 76 0 77 0;
#X connect 77 0 78 0;
#X connect 78 0 73 0;
```

25 puzzle/puzzle3x3-fx.pd

```
#N canvas 47 647 1279 239 10;
#X obj 14 16 catch~ \$1-0-L;
#X obj 154 16 catch~ \$1-1-L;
#X obj 294 16 catch~ \$1-2-L;
#X obj 434 16 catch~ \$1-3-L;
#X obj 574 16 catch~ \$1-4-L;
#X obj 714 16 catch~ \$1-5-L;
#X obj 854 16 catch~ \$1-6-L;
#X obj 994 16 catch~ \$1-7-L;
#X obj 1124 16 catch~ \$1-8-L;
#X obj 542 218 outlet~;
#X obj 670 217 outlet~;
#X obj 133 119 puzzle3x3-pitchshift~;
#X obj 433 71 *~ 1.5;
#X obj 486 69 *~ 0.5;
#X obj 711 62 *~ 0.5;
#X obj 768 63 *~ 1.5;
#X obj 16 64 *~ 1.5;
#X obj 74 64 *~ 0.5;
#X obj 291 73 *~ 0.5;
#X obj 371 72 *~ 1.5;
#X obj 854 68 *~ 1.5;
#X obj 911 64 *~ 0.5;
#X obj 1123 63 *~ 0.5;
#X obj 1180 64 *~ 1.5;
```

```
#X obj 546 116 puzzle3x3-delay~;
#X obj 13 40 puzzle3x3-ring~;
#X obj 856 43 puzzle3x3-ring~;
#X obj 433 42 puzzle3x3-ring~;
#X obj 293 42 puzzle3x3-samphold~;
#X obj 715 39 puzzle3x3-samphold~;
#X obj 1123 41 puzzle3x3-samphold~;
#X obj 978 110 puzzle3x3-comb~;
#X obj 154 39 *~ 2;
#X obj 574 39 *~ 2;
#X obj 994 39 *~ 2;
#X obj 542 194 puzzle3x3-compress~ 80;
#X obj 545 169 *~ 1;
#X obj 652 169 *~ 1;
#X connect 0 0 25 0;
#X connect 1 0 32 0;
#X connect 2 0 28 0;
#X connect 3 0 27 0;
#X connect 4 0 33 0;
#X connect 5 0 29 0;
#X connect 6 0 26 0;
#X connect 7 0 34 0;
#X connect 8 0 30 0;
#X connect 11 0 36 0;
#X connect 11 1 37 0;
#X connect 12 0 24 0;
#X connect 13 0 24 1;
#X connect 14 0 24 0;
#X connect 15 0 24 1;
#X connect 16 0 11 0;
#X connect 17 0 11 1;
#X connect 18 0 11 0;
#X connect 19 0 11 1;
#X connect 20 0 31 0;
#X connect 21 0 31 1;
#X connect 22 0 31 0;
#X connect 23 0 31 1;
#X connect 24 0 36 0;
#X connect 24 1 37 0;
#X connect 25 0 16 0;
#X connect 25 0 17 0;
#X connect 26 0 20 0;
#X connect 26 0 21 0;
#X connect 27 0 12 0;
#X connect 27 0 13 0;
#X connect 28 0 18 0;
#X connect 28 0 19 0;
#X connect 29 0 14 0;
#X connect 29 0 15 0;
#X connect 30 0 22 0;
#X connect 30 0 23 0;
#X connect 31 0 36 0;
#X connect 31 1 37 0;
#X connect 32 0 11 0;
#X connect 32 0 11 1;
#X connect 33 0 24 0;
#X connect 33 0 24 1;
```

```
#X connect 34 0 31 0;
#X connect 34 0 31 1;
#X connect 35 0 9 0;
#X connect 35 1 10 0;
#X connect 36 0 35 0;
#X connect 37 0 35 1;
```

26 puzzle/puzzle3x3-markov.pd_lua

```
local P = pd.Class:new():register("puzzle3x3-markov")

math.randomseed( tonumber(tostring(os.time()):reverse():sub(1,6)) )

function P:initialize(atoms)
  self.inlets = 1
  self.outlets = 1
  self.x = 2
  self.y = 2
  self.back = "_"
  self.moves = {
    { { "S", "E" }, { "W", "S", "E" }, { "W", "S" } },
    { { "N", "S", "E" }, { "N", "W", "S", "E" }, { "N", "W", "S" } },
    { { "N", "E" }, { "N", "W", "E" }, { "N", "W" } }
  }
  self.spin = { "L", "R" }
  self.flip = { "NS", "SN", "EW", "WE", "NWSE", "SENW", "NESW", "SWNE" }
  self.dx = { W = -1, E = 1, N = 0, S = 0 }
  self.dy = { W = 0, E = 0, N = -1, S = 1 }
  self.opposite = { W = "E", E = "W", N = "S", S = "N" }
  return true
end

function P:in_1_bang()
  local d, r, m
  r = math.random(1,16)
  if r == 1 then
    t = math.random(0,7)
    r = math.random(1,2)
    m = self.spin[r]
    self:outlet(1, m, { t })
  else if r == 2 then
    t = math.random(0,7)
    r = math.random(1,8)
    m = self.flip[r]
    self:outlet(1, m, { t })
  else while true do
    d = 2
    if self.x == 2 then d = d + 1 end
    if self.y == 2 then d = d + 1 end
    r = math.random(1, d)
    m = self.moves[self.y][self.x][r]
    if m ~= self.back then
      self.back = self.opposite[m]
      self.x = self.x + self.dx[m]
      self.y = self.y + self.dy[m]
      self:outlet(1, m, { })
      return
```

```
    end
  end end end
end
```

27 puzzle/PUZZLE3X3.pd

```
#N canvas 3 91 306 107 10;
#N canvas 101 61 328 598 \$0-puzzle 0;
#X obj 14 63 gemhead;
#X obj 61 437 dac~;
#X obj 116 184 puzzle3x3-markov;
#X obj 48 184 puzzle3x3;
#X obj 8 162 t a b;
#X obj 31 408 puzzle3x3-fx \$0;
#X obj 11 11 tgl 32 0 \$0-go-s \$0-go-r GO 7 7 0 16 -204786 -1 -1 1
1;
#X obj 18 461 puzzle3x3-record;
#X floatatom 18 481 5 0 0 0 - - -;
#X obj 144 463 tgl 15 0 empty empty empty 17 7 0 10 -262144 -1 -1 0
1;
#X obj 10 108 scale 2.66667;
#X obj 9 134 scale 0.92;
#N canvas 295 61 401 523 \$0-base-tile 0;
#X obj 16 16 inlet;
#X msg 24 317 snap;
#X obj 60 153 loadbang;
#X obj 258 222 separator;
#X obj 23 434 pix_texture;
#X msg 71 406 quality 2;
#X obj 23 375 pix_snap2tex 0 0 512 512;
#X obj 55 352 GEMglClear;
#X obj 110 331 GLdefine GL_DEPTH_BUFFER_BIT;
#X obj 20 183 alpha;
#X obj 258 480 square 8;
#X obj 256 295 colorRGB 1 1 1 1;
#X obj 253 273 scale 0.48;
#X obj 22 205 t b a a a a;
#X obj 22 480 outlet;
#X msg 30 164 1;
#X obj 274 5 inlet;
#X obj 23 398 GEMglGenerateMipmap GL_TEXTURE_2D;
#X connect 0 0 9 0;
#X connect 1 0 6 0;
#X connect 2 0 5 0;
#X connect 2 0 8 0;
#X connect 2 0 15 0;
#X connect 3 0 12 0;
#X connect 4 0 14 0;
#X connect 5 0 4 0;
#X connect 6 0 17 0;
#X connect 6 1 4 1;
#X connect 8 0 7 1;
#X connect 9 0 13 0;
#X connect 11 0 10 0;
#X connect 12 0 11 0;
#X connect 13 0 1 0;
#X connect 13 1 6 0;
```

```
#X connect 13 2 7 0;
#X connect 13 4 3 0;
#X connect 15 0 9 0;
#X connect 16 0 6 2;
#X connect 17 0 4 0;
#X restore 12 86 pd \$0-base-tile;
#N canvas 3 61 635 425 \$0-context 0;
#X msg 94 262 0 \, destroy;
#X msg 26 299 \; pd dsp \$1;
#X msg 43 270 1;
#X obj 32 52 t b b;
#X msg 9 267 0;
#X obj 182 17 r PUZZLE;
#X obj 147 136 list append;
#X obj 31 31 sel 0 1;
#X obj 31 8 r \$0-go-s;
#X obj 182 39 t b a;
#X msg 183 86 1;
#X obj 182 112 s \$0-go-r;
#X obj 182 63 delay 1000;
#X msg 178 264 title Puzzle \, border 0 \, dimen \$1 \$2 \, offset
\$3 \$4 \, color 0 0 0 \, create \$5 \, 1;
#X obj 236 235 print PUZZLE;
#X obj 220 314 outlet;
#X obj 148 157 t a a a a;
#X msg 109 157 \$6;
#X obj 109 179 pipe f 0;
#X obj 109 201 t a b;
#X obj 109 223 pipe f 0;
#X obj 283 316 outlet;
#X msg 187 231 \$1 \$2;
#X obj 287 13 loadbang;
#X msg 287 35 512 512 512 32 :0.0 0;
#X obj 138 311 gemwin 20;
#X connect 0 0 25 0;
#X connect 2 0 1 0;
#X connect 3 0 4 0;
#X connect 3 1 0 0;
#X connect 4 0 1 0;
#X connect 5 0 9 0;
#X connect 6 0 16 0;
#X connect 7 0 3 0;
#X connect 7 1 6 0;
#X connect 8 0 7 0;
#X connect 9 0 12 0;
#X connect 9 1 6 1;
#X connect 10 0 11 0;
#X connect 12 0 10 0;
#X connect 13 0 25 0;
#X connect 16 0 17 0;
#X connect 16 1 13 0;
#X connect 16 2 22 0;
#X connect 16 3 14 0;
#X connect 17 0 18 0;
#X connect 18 0 19 0;
#X connect 19 0 20 0;
#X connect 19 1 2 0;
```

```
#X connect 20 0 21 0;
#X connect 22 0 15 0;
#X connect 23 0 24 0;
#X connect 24 0 9 0;
#X restore 114 57 pd \$0-context;
#N canvas 3 58 450 300 \$0-tween 0;
#X obj 31 48 f 0;
#X obj 73 46 + 1;
#X obj 104 147 sel 0;
#X obj 30 98 t f f;
#X obj 28 205 clip 0 1;
#X obj 30 170 expr 0.5 - 0.5 * tanh(cos(3.141592653 * tanh($f1 * 2)))/tanh(-cos↙
    ↳ (3.141592653
* tanh(2)));
#X obj 31 14 inlet;
#X obj 28 226 outlet;
#X obj 101 227 outlet;
#X obj 31 70 / 40;
#X obj 72 68 mod 40;
#X connect 0 0 1 0;
#X connect 0 0 9 0;
#X connect 1 0 10 0;
#X connect 2 0 8 0;
#X connect 3 0 5 0;
#X connect 3 1 2 0;
#X connect 4 0 7 0;
#X connect 5 0 4 0;
#X connect 6 0 0 0;
#X connect 9 0 3 0;
#X connect 10 0 0 1;
#X restore 50 162 pd \$0-tween;
#N canvas 0 50 450 300 \$0-debug 0;
#X obj 40 40 loadbang;
#X obj 40 61 samplerate~;
#X obj 40 82 print SR;
#X connect 0 0 1 0;
#X connect 1 0 2 0;
#X restore 101 21 pd \$0-debug;
#X obj 18 360 puzzle3x3-tile 0 \$0;
#X obj 29 384 puzzle3x3-drumkit~ \$0;
#X obj 18 220 puzzle3x3-tile 7 \$0;
#X obj 18 280 puzzle3x3-tile 4 \$0;
#X obj 18 240 puzzle3x3-tile 6 \$0;
#X obj 18 260 puzzle3x3-tile 5 \$0;
#X obj 18 300 puzzle3x3-tile 3 \$0;
#X obj 18 320 puzzle3x3-tile 2 \$0;
#X obj 18 340 puzzle3x3-tile 1 \$0;
#X connect 0 0 12 0;
#X connect 2 0 3 1;
#X connect 3 0 18 1;
#X connect 4 0 18 0;
#X connect 4 1 14 0;
#X connect 5 0 1 0;
#X connect 5 0 7 1;
#X connect 5 1 1 1;
#X connect 5 1 7 2;
#X connect 7 0 8 0;
```

```
#X connect 9 0 7 3;
#X connect 10 0 11 0;
#X connect 11 0 4 0;
#X connect 12 0 10 0;
#X connect 13 0 12 1;
#X connect 13 1 9 0;
#X connect 14 0 3 0;
#X connect 14 1 2 0;
#X connect 16 0 7 0;
#X connect 18 0 20 0;
#X connect 18 1 20 1;
#X connect 19 0 22 0;
#X connect 19 1 22 1;
#X connect 20 0 21 0;
#X connect 20 1 21 1;
#X connect 21 0 19 0;
#X connect 21 1 19 1;
#X connect 22 0 23 0;
#X connect 22 1 23 1;
#X connect 23 0 24 0;
#X connect 23 1 24 1;
#X connect 24 0 16 0;
#X connect 24 1 16 1;
#X coords 0 -1 1 1 34 34 2 10 10;
#X restore 11 37 pd \$0-puzzle;
#X text 58 37 Puzzle (GPL) Claude Heiland-Allen;
#X text 58 51 mathr.co.uk;
```

28 puzzle/puzzle3x3.pd_lua

```
local P = pd.Class:new():register("puzzle3x3")

local function O()
  return {
    0,0,0,0,
    0,0,0,0,
    0,0,0,0,
    0,0,0,0
  }
end

local function I()
  return {
    1,0,0,0,
    0,1,0,0,
    0,0,1,0,
    0,0,0,1
  }
end

local function M(a,b)
  local c = O(), i, j, k
  for i = 0,3 do
    for j = 0,3 do
      for k = 0,3 do
        c[4 * i + j + 1] = c[4 * i + j + 1] + a[4 * i + k + 1] * b[4 * k + j + ↙
            ↳ 1]
```

```
      end
    end
  end
  return c
end

local function translate(x, y) return function(t)
  return {
    1,0,0,x*t,
    0,1,0,y*t,
    0,0,1,0,
    0,0,0,1
  }
end end

local function rotate(x,y,z)
  local a, h, i, j, k, l
  if x ~= 0 then
    a = x
    h = 4 * 0 + 0 + 1
    i = 4 * 1 + 1 + 1
    j = 4 * 1 + 2 + 1
    k = 4 * 2 + 1 + 1
    l = 4 * 2 + 2 + 1
  else if y ~= 0 then
    a = y
    h = 4 * 1 + 1 + 1
    i = 4 * 0 + 0 + 1
    j = 4 * 0 + 2 + 1
    k = 4 * 2 + 0 + 1
    l = 4 * 2 + 2 + 1
  else if z ~= 0 then
    a = z
    h = 4 * 2 + 2 + 1
    i = 4 * 0 + 0 + 1
    j = 4 * 0 + 1 + 1
    k = 4 * 1 + 0 + 1
    l = 4 * 1 + 1 + 1
  end end end
  return function(t)
    local m, c, s, ll
    ll = 1
    if t == 1 then
      if a ==   0 then c =  1 ; s =  0 else
      if a ==  90 then c =  0 ; s =  1 else
      if a == -90 then c =  0 ; s = -1 else
      if a == 180 then c = -1 ; s =  0 else
      if a == -180 then c = -1 ; s =  0 else
      if a ==  45 then c = math.sqrt(0.5) ; s =  math.sqrt(0.5) else
      if a == -45 then c = math.sqrt(0.5) ; s = -math.sqrt(0.5) else
      c = math.cos(a * math.pi / 180)
      s = math.sin(a * math.pi / 180)
      end end end end end end end
    else
      if a == 90 then
        ll = math.sqrt(0.5) / math.cos((t * 90 - 45) * math.pi / 180)
      else if a == -90 then
```

```
        ll = math.sqrt(0.5) / math.cos((t * 90 - 45) * math.pi / 180)
      end end
      c = math.cos(t * a * math.pi / 180)
      s = math.sin(t * a * math.pi / 180)
    end
    m = I()
    m[h] = ll
    m[i] = ll *  c
    m[j] = ll *  s
    m[k] = ll * -s
    m[l] = ll *  c
    return m
  end
end

local function deepen(t)
  local m
  m = I()
  m[12] = -0.5 * math.cos((t * 180 - 90) * math.pi / 180)
  return m
end

function P:initialize(atoms)
  self.inlets = 2
  self.outlets = 2
  self.x = 2
  self.y = 2
  self.tiles = {
    {  0,  1,  2 },
    {  7, "_", 3 },
    {  6,  5,  4 }
  }
  self.matrix = { }
  self.tweenL = { }
  self.tweenR = { }
  local x, y, t
  for y = 1,3 do
    for x = 1,3 do
      t = self.tiles[y][x]
      if t ~= "_" then
        self.matrix[t] = translate(x - 2, y - 2)(1)
        self.tweenL[t] = I
        self.tweenR[t] = I
      end
    end
  end
  return true
end

function P:in_2_N()
  if self.y > 1 then
    self:update()
    self.tweenL[self.tiles[self.y - 1][self.x]] = translate(0, 1)
    self.tiles[self.y][self.x] = self.tiles[self.y - 1][self.x]
    self.y = self.y - 1
    self.tiles[self.y][self.x] = "_"
  else
```

```
      self:outlet(2, "bang", { })
    end
  end
  function P:in_2_E()
    if self.x < 3 then
      self:update()
      self.tweenL[self.tiles[self.y][self.x + 1]] = translate(-1, 0)
      self.tiles[self.y][self.x] = self.tiles[self.y][self.x + 1]
      self.x = self.x + 1
      self.tiles[self.y][self.x] = "_"
    else
      self:outlet(2, "bang", { })
    end
  end
  function P:in_2_S()
    if self.y < 3 then
      self:update()
      self.tweenL[self.tiles[self.y + 1][self.x]] = translate(0, -1)
      self.tiles[self.y][self.x] = self.tiles[self.y + 1][self.x]
      self.y = self.y + 1
      self.tiles[self.y][self.x] = "_"
    else
      self:outlet(2, "bang", { })
    end
  end
  function P:in_2_W()
    if self.x > 1 then
      self:update()
      self.tweenL[self.tiles[self.y][self.x - 1]] = translate(1, 0)
      self.tiles[self.y][self.x] = self.tiles[self.y][self.x - 1]
      self.x = self.x - 1
      self.tiles[self.y][self.x] = "_"
    else
      self:outlet(2, "bang", { })
    end
  end

  function P:in_2_L(atoms)
    self:update()
    self.tweenR[atoms[1]] = rotate(0,0,90)
  end
  function P:in_2_R(atoms)
    self:update()
    self.tweenR[atoms[1]] = rotate(0,0,-90)
  end

  function P:in_2_NS(atoms)
    self:update()
    self.tweenR[atoms[1]] = rotate(180,0,0)
    self.tweenL[atoms[1]] = deepen
  end
  function P:in_2_SN(atoms)
    self:update()
    self.tweenR[atoms[1]] = rotate(-180,0,0)
    self.tweenL[atoms[1]] = deepen
  end
```

```
function P:in_2_EW(atoms)
  self:update()
  self.tweenR[atoms[1]] = rotate(0,180,0)
  self.tweenL[atoms[1]] = deepen
end
function P:in_2_WE(atoms)
  self:update()
  self.tweenR[atoms[1]] = rotate(0,-180,0)
  self.tweenL[atoms[1]] = deepen
end
function P:in_2_NWSE(atoms)
  self:update()
  self.tweenR[atoms[1]] = function(t) return M(rotate(0,0,45)(1),M(rotate(0,180,0)(t),rotate(0,0,-45)(1))) end
  self.tweenL[atoms[1]] = deepen
end
function P:in_2_SENW(atoms)
  self:update()
  self.tweenR[atoms[1]] = function(t) return M(rotate(0,0,45)(1),M(rotate(0,-180,0)(t),rotate(0,0,-45)(1))) end
  self.tweenL[atoms[1]] = deepen
end
function P:in_2_NESW(atoms)
  self:update()
  self.tweenR[atoms[1]] = function(t) return M(rotate(0,0,-45)(1),M(rotate(0,180,0)(t),rotate(0,0,45)(1))) end
  self.tweenL[atoms[1]] = deepen
end
function P:in_2_SWNE(atoms)
  self:update()
  self.tweenR[atoms[1]] = function(t) return M(rotate(0,0,-45)(1),M(rotate(0,-180,0)(t),rotate(0,0,45)(1))) end
  self.tweenL[atoms[1]] = deepen
end

function P:in_1_float(z)
  local x, y, t, m
  for y = 1,3 do
    for x = 1,3 do
      t = self.tiles[y][x]
      if t ~= "_" then
        m = M(M(self.tweenL[t](z), self.matrix[t]), self.tweenR[t](z))
        self:outlet(1, "list", { t, "matrix", m[1], m[2], m[3], m[4], m[5], m[6], m[7], m[8], m[9], m[10], m[11], m[12], m[13], m[14], m[15], m[16] })
      end
    end
  end
end

function P:update()
  for t = 0,7 do
    self.matrix[t] = M(M(self.tweenL[t](1), self.matrix[t]), self.tweenR[t](1))
    self.tweenL[t] = I
    self.tweenR[t] = I
  end
end
```

29 puzzle/puzzle3x3-pitchshift~.pd

```
#N canvas 3 61 432 553 10;
#X obj 17 19 inlet~;
#X obj 51 397 cos~;
#X obj 52 419 *~;
#X obj 52 448 +~;
#X obj 109 205 wrap~;
#X obj 110 402 cos~;
#X obj 110 429 *~;
#X obj 109 178 +~ 0.5;
#X obj 51 345 -~ 0.5;
#X obj 51 371 *~ 0.5;
#X obj 110 341 -~ 0.5;
#X obj 110 374 *~ 0.5;
#X obj 52 155 phasor~;
#X obj 51 104 *~ 0.05776;
#X obj 51 130 expr~ -10*(exp($v1)-1);
#X obj 51 475 outlet~;
#X obj 291 45 *~ -0.1;
#X obj 310 66 *~ 0.1;
#X obj 126 292 vd~ \$0-del-L;
#X obj 65 319 vd~ \$0-del-L;
#X obj 241 397 cos~;
#X obj 242 419 *~;
#X obj 242 448 +~;
#X obj 299 205 wrap~;
#X obj 300 402 cos~;
#X obj 300 429 *~;
#X obj 299 178 +~ 0.5;
#X obj 241 345 -~ 0.5;
#X obj 241 371 *~ 0.5;
#X obj 300 341 -~ 0.5;
#X obj 300 374 *~ 0.5;
#X obj 242 155 phasor~;
#X obj 241 104 *~ 0.05776;
#X obj 241 130 expr~ -10*(exp($v1)-1);
#X obj 241 475 outlet~;
#X obj 186 20 inlet~;
#X obj 316 292 vd~ \$0-del-R;
#X obj 255 319 vd~ \$0-del-R;
#X obj 19 516 delwrite~ \$0-del-L 200;
#X obj 188 517 delwrite~ \$0-del-R 200;
#X obj 63 292 +~ 2;
#X obj 126 263 +~ 2;
#X obj 253 292 +~ 2;
#X obj 316 263 +~ 2;
#X obj 316 230 *~ 60;
#X obj 253 266 *~ 60;
#X obj 126 230 *~ 60;
#X obj 63 266 *~ 60;
#X obj 307 23 osc~ 0.25;
#X obj 297 452 hip~ 1;
#X obj 115 450 hip~ 1;
#X obj 113 66 sig~ -1;
#X obj 298 476 *~ 0.75;
#X obj 115 473 *~ 0.75;
```

```
#X connect 0 0 3 0;
#X connect 1 0 2 0;
#X connect 2 0 3 0;
#X connect 3 0 15 0;
#X connect 3 0 50 0;
#X connect 4 0 10 0;
#X connect 4 0 46 0;
#X connect 5 0 6 0;
#X connect 6 0 3 1;
#X connect 7 0 4 0;
#X connect 8 0 9 0;
#X connect 9 0 1 0;
#X connect 10 0 11 0;
#X connect 11 0 5 0;
#X connect 12 0 8 0;
#X connect 12 0 7 0;
#X connect 12 0 47 0;
#X connect 13 0 14 0;
#X connect 14 0 12 0;
#X connect 16 0 13 0;
#X connect 17 0 32 0;
#X connect 18 0 6 1;
#X connect 19 0 2 1;
#X connect 20 0 21 0;
#X connect 21 0 22 0;
#X connect 22 0 34 0;
#X connect 22 0 49 0;
#X connect 23 0 29 0;
#X connect 23 0 44 0;
#X connect 24 0 25 0;
#X connect 25 0 22 1;
#X connect 26 0 23 0;
#X connect 27 0 28 0;
#X connect 28 0 20 0;
#X connect 29 0 30 0;
#X connect 30 0 24 0;
#X connect 31 0 27 0;
#X connect 31 0 26 0;
#X connect 31 0 45 0;
#X connect 32 0 33 0;
#X connect 33 0 31 0;
#X connect 35 0 22 0;
#X connect 36 0 25 1;
#X connect 37 0 21 1;
#X connect 40 0 19 0;
#X connect 41 0 18 0;
#X connect 42 0 37 0;
#X connect 43 0 36 0;
#X connect 44 0 43 0;
#X connect 45 0 42 0;
#X connect 46 0 41 0;
#X connect 47 0 40 0;
#X connect 48 0 16 0;
#X connect 48 0 17 0;
#X connect 49 0 52 0;
#X connect 50 0 53 0;
#X connect 51 0 13 0;
```

```
#X connect 51 0 32 0;
#X connect 52 0 39 0;
#X connect 53 0 38 0;
```

30 puzzle/puzzle3x3-record.pd

```
#N canvas 3 44 388 424 10;
#X obj 21 19 inlet;
#X obj 114 20 inlet~;
#X obj 219 22 inlet~;
#X obj 342 21 inlet;
#X obj 125 366 writesf~ 2;
#X msg 292 318 stop;
#X obj 22 82 spigot;
#X obj 22 248 pix_write;
#X obj 34 131 spigot;
#X obj 34 154 t b b;
#X msg 104 145 0;
#X obj 8 195 loadbang;
#X obj 22 269 b;
#X obj 21 292 f 0;
#X obj 61 294 + 1;
#X obj 292 293 sel 0;
#X obj 342 51 f;
#X obj 20 397 outlet;
#X msg 8 217 file out/img 0 \, auto 1;
#X msg 34 317 open -bytes 4 out/puzzle.wav \, start;
#X connect 0 0 6 0;
#X connect 1 0 4 0;
#X connect 2 0 4 1;
#X connect 3 0 16 0;
#X connect 5 0 4 0;
#X connect 6 0 7 0;
#X connect 6 0 8 0;
#X connect 7 0 12 0;
#X connect 8 0 9 0;
#X connect 9 0 19 0;
#X connect 9 1 10 0;
#X connect 10 0 8 1;
#X connect 11 0 18 0;
#X connect 12 0 13 0;
#X connect 13 0 14 0;
#X connect 13 0 17 0;
#X connect 14 0 13 1;
#X connect 15 0 5 0;
#X connect 16 0 6 1;
#X connect 16 0 8 1;
#X connect 16 0 15 0;
#X connect 18 0 7 0;
#X connect 19 0 4 0;
```

31 puzzle/puzzle3x3-ring~.pd

```
#N canvas 3 61 687 501 10;
#X obj 69 3 inlet~;
#X obj 71 460 outlet~;
```

```
#X obj 95 78 rpole~;
#X obj 95 53 rzero_rev~;
#X obj 95 127 rpole~;
#X obj 95 102 rzero_rev~;
#X obj 95 175 rpole~;
#X obj 95 150 rzero_rev~;
#X obj 95 224 rpole~;
#X obj 95 199 rzero_rev~;
#X obj 185 79 rpole~;
#X obj 185 54 rzero_rev~;
#X obj 185 128 rpole~;
#X obj 185 103 rzero_rev~;
#X obj 185 176 rpole~;
#X obj 185 151 rzero_rev~;
#X obj 185 225 rpole~;
#X obj 185 200 rzero_rev~;
#X obj 275 79 rpole~;
#X obj 275 54 rzero_rev~;
#X obj 275 128 rpole~;
#X obj 275 103 rzero_rev~;
#X obj 275 176 rpole~;
#X obj 275 151 rzero_rev~;
#X obj 275 225 rpole~;
#X obj 275 200 rzero_rev~;
#X obj 72 436 *~ 0.5;
#X obj 132 31 expr~ 1 - 0.01 - 0.3*abs($v1-0.5)*abs($v1-0.5);
#X obj 135 6 phasor~ 4;
#X connect 0 0 26 0;
#X connect 0 0 3 0;
#X connect 2 0 5 0;
#X connect 3 0 2 0;
#X connect 4 0 7 0;
#X connect 5 0 4 0;
#X connect 6 0 9 0;
#X connect 7 0 6 0;
#X connect 8 0 11 0;
#X connect 8 0 26 0;
#X connect 9 0 8 0;
#X connect 10 0 13 0;
#X connect 11 0 10 0;
#X connect 12 0 15 0;
#X connect 13 0 12 0;
#X connect 14 0 17 0;
#X connect 15 0 14 0;
#X connect 16 0 19 0;
#X connect 16 0 26 0;
#X connect 17 0 16 0;
#X connect 18 0 21 0;
#X connect 19 0 18 0;
#X connect 20 0 23 0;
#X connect 21 0 20 0;
#X connect 22 0 25 0;
#X connect 23 0 22 0;
#X connect 24 0 26 0;
#X connect 25 0 24 0;
#X connect 26 0 1 0;
#X connect 27 0 8 1;
```

```
#X connect 27 0 9 1;
#X connect 27 0 6 1;
#X connect 27 0 7 1;
#X connect 27 0 4 1;
#X connect 27 0 5 1;
#X connect 27 0 2 1;
#X connect 27 0 3 1;
#X connect 27 0 11 1;
#X connect 27 0 10 1;
#X connect 27 0 13 1;
#X connect 27 0 12 1;
#X connect 27 0 15 1;
#X connect 27 0 14 1;
#X connect 27 0 17 1;
#X connect 27 0 16 1;
#X connect 27 0 19 1;
#X connect 27 0 18 1;
#X connect 27 0 21 1;
#X connect 27 0 20 1;
#X connect 27 0 23 1;
#X connect 27 0 22 1;
#X connect 27 0 25 1;
#X connect 27 0 24 1;
#X connect 28 0 27 0;
```

32 puzzle/puzzle3x3-samphold~.pd

```
#N canvas 3 61 341 300 10;
#X obj 15 193 samphold~;
#X obj 76 114 phasor~;
#X obj 77 45 osc~ 0.05;
#X obj 17 13 inlet~;
#X obj 14 248 outlet~;
#X obj 77 67 *~ 1000;
#X obj 77 89 +~ 3000;
#X obj 16 167 expr~ sin(6 * $v1);
#X connect 0 0 4 0;
#X connect 1 0 0 1;
#X connect 2 0 5 0;
#X connect 3 0 7 0;
#X connect 5 0 6 0;
#X connect 6 0 1 0;
#X connect 7 0 0 0;
```

33 puzzle/puzzle3x3-throw~.pd

```
#N canvas 3 61 450 300 10;
#X obj 26 161 *~;
#X obj 60 103 clip 0 1;
#X obj 16 19 inlet~;
#X obj 60 45 r \$1-x;
#X obj 178 47 r \$1-y;
#X obj 335 10 loadbang;
#X obj 300 44 f \$3;
#X obj 418 44 f \$4;
#X obj 22 215 throw~ \$2-L;
```

```
#X obj 16 258 outlet~;
#X obj 59 126 lop~ 25;
#X obj 60 69 expr 1-sqrt(($f3-$f1)*($f3-$f1)+($f4-$f2)*($f4-$f2))/sqrt(2.0)
;
#X connect 0 0 8 0;
#X connect 1 0 10 0;
#X connect 2 0 0 0;
#X connect 2 0 9 0;
#X connect 3 0 11 0;
#X connect 4 0 11 1;
#X connect 5 0 6 0;
#X connect 5 0 7 0;
#X connect 6 0 11 2;
#X connect 7 0 11 3;
#X connect 10 0 0 1;
#X connect 11 0 1 0;
```

34 puzzle/puzzle3x3-tile.pd

```
#N canvas 3 44 704 623 10;
#X obj 16 14 inlet;
#X obj 127 14 inlet;
#X obj 36 60 separator;
#X obj 127 40 route \$1;
#X obj 225 245 hsv2rgb;
#X obj 36 306 colorRGB;
#X obj 225 119 loadbang;
#X obj 225 142 f \$1;
#X obj 225 165 mod 8;
#X obj 225 193 / 8;
#X obj 47 276 unpack f f f;
#X obj 117 302 loadbang;
#X obj 202 117 bng 15 250 50 0 empty empty empty 17 7 0 10 -262144
-1 -1;
#X obj 16 514 outlet;
#X obj 36 336 square 1;
#X obj 207 305 puzzle3x3-throw~ \$0 \$2-0 -1 -1;
#X obj 16 36 t a a;
#X obj 182 515 outlet;
#X obj 36 249 scale 0.45 1 1 1;
#X obj 56 156 s \$0-x;
#X obj 113 155 s \$0-y;
#X obj 36 221 GEMglMultTransposeMatrixf;
#X msg 56 107 \$4 \$8;
#X obj 56 128 unpack f f;
#X msg 118 322 1;
#X msg 225 220 \$1 0.333 1;
#X obj 127 64 route matrix;
#X obj 207 266 catch~ \$2-\$1-src;
#X obj 207 325 puzzle3x3-throw~ \$0 \$2-1 0 -1;
#X obj 207 345 puzzle3x3-throw~ \$0 \$2-2 1 -1;
#X obj 207 365 puzzle3x3-throw~ \$0 \$2-3 -1 0;
#X obj 207 385 puzzle3x3-throw~ \$0 \$2-4 0 0;
#X obj 207 405 puzzle3x3-throw~ \$0 \$2-5 1 0;
#X obj 207 425 puzzle3x3-throw~ \$0 \$2-6 -1 1;
#X obj 207 445 puzzle3x3-throw~ \$0 \$2-7 0 1;
#X obj 207 465 puzzle3x3-throw~ \$0 \$2-8 1 1;
```

```
#X obj 207 287 hip~ 1;
#X connect 0 0 16 0;
#X connect 1 0 3 0;
#X connect 2 0 21 0;
#X connect 3 0 26 0;
#X connect 3 1 17 0;
#X connect 4 0 10 0;
#X connect 5 0 14 0;
#X connect 6 0 7 0;
#X connect 7 0 8 0;
#X connect 8 0 9 0;
#X connect 9 0 25 0;
#X connect 10 0 5 1;
#X connect 10 1 5 2;
#X connect 10 2 5 3;
#X connect 11 0 24 0;
#X connect 12 0 7 0;
#X connect 15 0 28 0;
#X connect 16 0 13 0;
#X connect 16 1 2 0;
#X connect 18 0 5 0;
#X connect 21 0 18 0;
#X connect 22 0 23 0;
#X connect 23 0 19 0;
#X connect 23 1 20 0;
#X connect 24 0 5 4;
#X connect 25 0 4 0;
#X connect 26 0 22 0;
#X connect 26 0 21 1;
#X connect 27 0 36 0;
#X connect 28 0 29 0;
#X connect 29 0 30 0;
#X connect 30 0 31 0;
#X connect 31 0 32 0;
#X connect 32 0 33 0;
#X connect 33 0 34 0;
#X connect 34 0 35 0;
#X connect 36 0 15 0;
```

35 puzzle/puzzle3x3-tswap.pd_lua

```
local tswap = pd.Class:new():register("puzzle3x3-tswap")

function tswap:initialize(sel, atoms)
  self.inlets = 1
  self.outlets = 0
  return true
end

function tswap:in_1_symbol(s)
  local t = pd.Table:new():sync(s)
  if t ~= nil then
    local i
    local l = t:length()
    local n = 0
    for i = 1,l do
      local x = t:get(i-1)
```

```
      if x ~= 0 then
        n = n + 1
      end
    end
    local m = math.random(1, n)
    n = 0
    local a = math.random(1, l)
    local b = math.random(1, l)
    for i = 1,l do
      local x = t:get(i-1)
      if x ~= 0 then
        n = n + 1
      end
      if n == m then
        a = i
      end
    end
    local x = t:get(a-1)
    local y = t:get(b-1)
    t:set(a-1,y)
    t:set(b-1,x)
    t:redraw()
  end
end
```

36 README

```
Claude Heiland-Allen :: Puzzle
------------------------------------------------------------------------

oOoO   oOoO   oOoO      Sliding tile puzzles have existed for over a
oOoOoO oOoOoO oOoOoO    century. The 15-puzzle craze in 1880 offered
oOoOoO oOoOoO oOoOoO    a cash prize for a problem with no solution.

oOoO   oOoO   oOoO      In the Puzzle presented here the computer is
oOoOoO oOoOoO oOoOoO    manipulating the tiles. No malicious design,
oOoOoO oOoOoO oOoOoO    but insufficient specification means that no

oOoO   oOoO             solution can be found; the automaton forever
oOoOoO oOoOoO           explores the state space but finds every way
oOoOoO oOoOoO           to position the tiles as good as the last...


Each tile makes a sound, and each possible position has a processing
effect associated with it. Part of the Puzzle is to watch and listen
carefully, to see and hear and try to pick apart what it is that the

computer is doing, to reverse-engineer the machinery inside from its
outward appearance. This Puzzle is implemented in Pd + GEM for video
and pdlua for the tile-control logic. The video is built using eight

squares, each coloured tile is textured with the whole Puzzle, which
gives an infinite fractal cascade. The control algorithm is a Markov
Chain that avoids repetition. Puzzle is Free Software under GNU/GPL.


                        Technical requirements for Puzzle are fairly
```

```
                        minimal: a computer, speakers or headphones,
                        video monitor or projector. Operating system

                        must have 3D-accelerated graphics drivers to
35                      result in adequate performance. For example,
                        Puzzle was developed on a Pentium-M 1.6G CPU

                        with ATI Mobility Radeon 9700 GPU, and works
                        also on a Raspberry Pi 3 B with experimental
40                      OpenGL, provided the resolution is kept low.
```

---

37 start.sh

```
#!/bin/bash
set -x
cd "$(dirname "$(readlink -f "${0}")")"
while true
5 do
# detect screens
SCREENS="$(xrandr | grep ' connected ' | sed 's/ .*$//g')"
INTERNAL="$(echo "${SCREENS}" | tr ' ' '\n' | grep    '^LVDS' | head -n 1 | tr -↙
    ↳ d '[:space:]' )"
EXTERNAL="$(echo "${SCREENS}" | tr ' ' '\n' | grep -v '^LVDS' | head -n 1 | tr -↙
    ↳ d '[:space:]' )"
10 SCREEN_OFF=""
if [ "x${EXTERNAL}" = "x" ]
then
  if [ "x${INTERNAL}" = "x" ]
  then
15   echo "puzzle: no screens recognized"
     exit 1
  else
     echo "puzzle: using internal screen"
     SCREEN="${INTERNAL}"
20 fi
else
  SCREEN="${EXTERNAL}"
  if [ "x${INTERNAL}" = "x" ]
  then
25   echo "puzzle: using external screen"
  else
     echo "puzzle: using external screen (internal detected too)"
     SCREEN_OFF="${INTERNAL}"
  fi
30 fi
# set up screens
if [ "x${SCREEN_OFF}" = 'x' ]
then
  xrandr --output "${SCREEN}" --auto
35 else
  xrandr --output "${SCREEN}" --auto --output "${SCREEN_OFF}" --off
fi
xset -dpms s off
xsetroot -solid black -cursor_name none
40 # get screen resolution
```

```
RESOLUTION="$(xrandr | grep "^${SCREEN} connected " | sed "s/primary //g" | sed ↙
    ↳ "s/^${SCREEN} connected \([^ ]*\) .*$/\1/")"
if [ "x${RESOLUTION}" = "x" ]
then
  echo "puzzle: failed to detect screen resolution"
  exit 1
fi
echo "${RESOLUTION}" | tr 'x+' '  ' | (
  read SCREENW SCREENH SCREENX SCREENY
  echo "resolution '$SCREENW' '$SCREENH' '$SCREENX' '$SCREENY'"
  if (( SCREENH >= 2048 ))
  then
    echo "puzzle: screen height ${SCREENH} too large"
  elif (( SCREENH >= 1024 ))
  then
    SCREENY=$(( SCREENY + (SCREENH - 1024)/2 ))
    SCREENH=1024
  elif (( SCREENH >= 512 ))
  then
    SCREENY=$(( SCREENY + (SCREENH - 512)/2 ))
    SCREENH=512
  elif (( SCREENH >= 256 ))
  then
    SCREENY=$(( SCREENY + (SCREENH - 256)/2 ))
    SCREENH=256
  else
    echo "puzzle: screen height ${SCREENH} too small"
    exit 1
  fi
  if (( SCREENH > SCREENW ))
  then
    echo "puzzle: screen width ${SCREENW} smaller than height ${SCREENH}"
    exit 1
  else
    SCREENX=$(( SCREENX + (SCREENW - SCREENH)/2 ))
    SCREENW=$(( SCREENH ))
  fi
  # move mouse cursor to bottom right hand corner
  ./warp 10000 10000
  # geometry of window centered on screen will be WxH+X+Y
  "${HOME}/opt/pd/0.49-0/bin/pd" -noprefs -stderr -rt -nogui -noadc -r 48000 \
    -alsa -audiobuf 240 -outchannels 2 -lib Gem:pdlua -open puzzle/PUZZLE3X3.pd ↙
        ↳ \
    -send "; PUZZLE ${SCREENW} ${SCREENH} ${SCREENX} ${SCREENY} ${DISPLAY} 0"
)
echo "puzzle: done"
sleep 5
done
```

38 start-simple.sh

```
#!/bin/sh
pd -noprefs -path "${HOME}/opt/pd/0.49-0/lib/pd/extra/Gem" -lib Gem:pdlua -jack ↙
    ↳ -r 48000 -channels 2 -send "PUZZLE 512 512 512 32 :0.0 0" puzzle/PUZZLE3X3↙
    ↳ .pd
```

39 TODO

```
* turn off all displays apart from the target, even if they are disconnected
* initial transition should scale tiles from nothing to normal size
```

40 warp.c

```
// https://stackoverflow.com/a/20581721

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

#include <X11/Xlib.h>
#include <X11/Xutil.h>

void mouseMove(int dx, int dy)
{
    Display *displayMain = XOpenDisplay(NULL);

    if(displayMain == NULL)
    {
        fprintf(stderr, "XOpenDisplay() failed\n");
        exit(EXIT_FAILURE);
    }

    XWarpPointer(displayMain, None, None, 0, 0, 0, 0, dx, dy);

    XCloseDisplay(displayMain);
}

int main(int argc, char **argv) {
  if (! (argc > 2)) return 1;
  mouseMove(atoi(argv[1]), atoi(argv[2]));
  return 0;
}
```