

gearbox

Claude Heiland-Allen

2012–2018

Contents

1	gearbox.cabal	2
2	GearBox.hs	3
3	.gitignore	9
4	LICENSE	9
5	Setup.hs	21

1 gearbox.cabal

```
Name:                gearbox
Version:             1.0.0.6
Synopsis:             zooming rotating fractal gears graphics demo
Description:          OpenGL gearbox iterated function system. Usage:
5      @gearbox quality@ where quality is an integer between
                        6 and 14 (default 9). @f@ toggles full screen, any
                        other key to quit. Shaders generate a gear shape
                        and hue rotate successive copies via texture feedback.
10      .
                        Blog post:
                        .
                        * <https://mathr.co.uk/blog/2012-01-12-gearbox.html>

Homepage:            https://code.mathr.co.uk/gearbox
15  License:          GPL-3
License-file:        LICENSE
Author:              Claude Heiland-Allen
Maintainer:          claude@mathr.co.uk
Category:            Demo
20  Build-type:       Simple
Cabal-version:       >=1.6

Executable gearbox
  Main-is:            GearBox.hs
25  Build-depends:
      base < 5,
      GLUT == 2.7.*,
      OpenGLRaw == 3.3.*,
      Vec == 1.0.*
30  GHC-options:      -Wall

Source-repository head
  type:               git
  location:            https://code.mathr.co.uk/gearbox.git
35  Source-repository this
```

```

type:          git
location:      https://code.mathr.co.uk/gearbox.git
tag:           v1.0.0.6

```

2 GearBox.hs

```

-- GearBox -- zooming rotating fractal gearbox demo
-- (GPL3+) 2012,2015,2018 Claude Heiland-Allen <claude@mathr.co.uk>
-- tested with: ghc-8.2.2, ghc-8.4.4, ghc-8.6.1

5 {-# LANGUAGE TypeOperators #-}
import Prelude hiding ((<>))
import Control.Monad (forM, forM_, replicateM)
import Data.Fixed (mod')
import Data.IRef (IORef, newIORef, readIORef, modifyIORef)
10 import Data.Vector hiding (last, map, scale, take)
import Foreign (Ptr, Storable(..), alloca, allocaArray, castPtr, nullPtr, with, ↵
    ↵ withArray)
import Foreign.C (withCString, peekCStringLen)
import Graphics.UI.GLUT
    ( Size(Size)
15     , DisplayMode(OpenGLMode, DoubleBuffered)
    , Key(Char)
    , KeyState(Down)
    , KeyboardMouseCallback
    , ($=)
20     , addTimerCallback
    , createWindow
    , displayCallback
    , getArgsAndInitialize
    , initialDisplayMode
    , initialWindowSize
25     , keyboardMouseCallback
    , mainLoop
    , postRedisplay
    , reportErrors
30     , reshapeCallback
    , swapBuffers
    , fullScreenToggle
    )
import Graphics.GL
35 import System.Exit (exitSuccess)

-- 4x4 matrix type
type R = GLdouble
type V = R :: R :: R :: R :: ()
40 type M = V :: V :: V :: V :: ()

-- length of animation loop in frames
looplength :: Int
looplength = 60 * 25

45
-- number of passes to iterate function system
passes :: Int
passes = 6

50 -- state for callbacks

```

```

data IFS = IFS
  -- immutable state
  { fbo :: GLuint
  , t_source, t_ping, t_pong :: GLuint
55  , p_shift :: GLuint, p_shift_u_t, p_shift_u_k :: GLint
  , w_tsize, w_width, w_height :: Int
  -- mutable state
  , w_frame :: Int, w_phase :: GLdouble
  }

60
-- shader to generate a single blue gear wheel
gear_frag_src :: String
gear_frag_src = unlines
  [ "void main() {"
65  , "    vec2 p = (gl_TexCoord[0].xy - vec2(0.5)) * 2.0;"
  , "    float r = length(p);"
  , "    float a = atan(p.y, p.x);"
  , "    float ro = 0.94 + 0.03 * sin(30.0 * a);"
  , "    float ri = 0.82 - 0.01 * sin(90.0 * a);"
70  , "    if (r < ro) {"
  , "        gl_FragColor = ri < r ? vec4(0.5,0.5,1.0,1.0) : vec4(0.25,0.25,0.5,1.0) ↵
  , "        ↵;"
  , "    } else {"
  , "        discard;"
  , "    }"
75  , "}"
  ]

-- shader to rotate colours in YUV space
shift_frag_src :: String
80 shift_frag_src = unlines
  [ "uniform sampler2D t;"
  , "uniform float k;"
  , ""
  , "void main() {"
85  , "    float c = cos(6.283185307179586 * k);"
  , "    float s = sin(6.283185307179586 * k);"
  , "    vec4 p = texture2D(t, gl_TexCoord[0].xy).rgba;"
  , "    mat3 yuv2rgb = mat3(1.0, 0.0, 1.28033, 1.0, -0.21482, -0.38059, 1.0, ↵
  , "        ↵ 2.12798, 0.0);" -- rec.709
  , "    mat3 rgb2yuv = mat3(0.2126, 0.7152, 0.0722, -0.09991, -0.33609, 0.43600, ↵
  , "        ↵ 0.615, -0.5586, -0.05639);"
90  , "    p.rgb = clamp(p.rgb * rgb2yuv * mat3(1, 0, 0, 0, c, s, 0, -s, c) * ↵
  , "        ↵ yuv2rgb, 0.0, 1.0);"
  , "    gl_FragColor = p;"
  , "}"
  ]

95 -- draw a textured quad
unitQuad :: IO ()
unitQuad = do
  glBegin GL_QUADS
  glTexCoord2f 0 0
100 glVertex2f (-1) (-1)
  glTexCoord2f 1 0
  glVertex2f 1 (-1)
  glTexCoord2f 1 1
  glVertex2f 1 1

```

```

105     glVertex2f 1 1
        glTexCoord2f 0 1
        glVertex2f (-1) 1
        glEnd

-- load a matrix into GL
110 loadMatrix :: M -> IO ()
loadMatrix ((a::b::c::d::())::(e::f::g::h::())::(i::j::k::l::())::(m::n::o::p)
    ↳ ::())::()) =
    withArray [a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p] $ glLoadMatrixd

-- helper for ping-pong technique
115 swap :: (a, b) -> (b, a)
swap ~(a,b) = (b,a)

-- simple transformations
scale :: R -> M
120 scale s = diagonal (s :: s :: s :: 1 :: ())
mirror :: M
mirror = diagonal (1 :: (-1) :: 1 :: 1 :: ())
rotateXY :: R -> M
rotateXY t = (c::s::0::0::())::((-s)::c::0::0::())::(0::0::1::0::())
    ↳ ::(0::0::0::1::())::()
125     where c = cos t ; s = sin t
translateX :: R -> M
translateX d = (1::0::0::0::())::(0::1::0::0::())::(0::0::1::0::())::(d
    ↳ ::0::0::1::())::()
(<>) :: M -> M -> M
(<>) = multmm
130
-- display callback
display :: IORef IFS -> IO ()
display ifsR = do
    ifs <- readIORef ifsR
135     -- compute transforms for iterated function system fractal
    let p0 = w_phase ifs
        f0 = w_frame ifs
        q = f0 `mod` looplength
        k = q < (looplength `div` 2)
140     p = if k then p0 else -p0
        s = 0.28
        d = 0.54
        transforms =
            [ identity
145             , mirror < scale s < rotateXY ( 4 * p)
              ] ++
            [ mirror < scale s < rotateXY (-3 * p) < translateX d <
              rotateXY (2 * pi * fromIntegral (if k then i else -i) / 5 + p)
              | i <- [0 .. 4 :: Int]
150             ]

-- iterated function system from gear to gears
glViewport 0 0 (fromIntegral $ w_size ifs) (fromIntegral $ w_size ifs)
glMatrixMode GLPROJECTION
glLoadIdentity
155 glOrtho (-1) 1 (-1) 1 (-1) 1
glMatrixMode GLMODELVIEW
glLoadIdentity

```

```

glBindFramebuffer GL_FRAMEBUFFER (fbo ifs)
glUseProgram (fromIntegral $ p_shift ifs)
160 glUniform1i (p_shift_u_t ifs) 0
to <- fmap last $ forM (take passes . ((True, (t_source ifs, t_ping ifs)) :)) .
    zip (repeat False) $ (iterate swap (t_ping ifs, t_pong ifs))) $
    \ (pass1, (ping, pong)) -> do
        glBindTexture GL_TEXTURE_2D ping
165 glGenerateMipmap GL_TEXTURE_2D
glFramebufferTexture2D GL_FRAMEBUFFER GL_COLOR_ATTACHMENT0 GL_TEXTURE_2D ↵
    ↵ pong 0
glClear (fromIntegral GL_COLOR_BUFFER_BIT)
forM_ ([0 .. 6 :: Int] 'zip' transforms) $ \ (w, m) -> do
    let sk = case w of
170         0 -> 0
         1 -> 0
         _ -> fromIntegral (4 * w - 7) / 10
    glUniform1f (p_shift_u_k ifs) sk
    loadMatrix $ (if pass1 && not k then rotateXY (pi / 30) else identity) <> ↵
        ↵ m
175 unitQuad
glBindTexture GL_TEXTURE_2D 0
glFramebufferTexture2D GL_FRAMEBUFFER GL_COLOR_ATTACHMENT0 GL_TEXTURE_2D 0 0
return pong
glUseProgram 0
180 glBindFramebuffer GL_FRAMEBUFFER 0
-- display the output scaled to the window
let w = fromIntegral $ w_width ifs
    h = fromIntegral $ w_height ifs
glViewport 0 0 w h
185 glClear (fromIntegral GL_COLOR_BUFFER_BIT)
glMatrixMode GL_PROJECTION
glLoadIdentity
let (ax, ay)
    | w >= h = (1, fromIntegral h / fromIntegral w)
190   | otherwise = (fromIntegral w / fromIntegral h, 1)
glOrtho (-ax) ax (-ay) ay (-1) 1
glMatrixMode GL_MODELVIEW
glLoadIdentity
glBindTexture GL_TEXTURE_2D to
195 glGenerateMipmap GL_TEXTURE_2D
loadMatrix $ scale 1.5 <>
    scale (s ** negate ((p0 / (2 * pi)) 'mod' 1)) <> rotateXY (- 2 * p)
unitQuad
glBindTexture GL_TEXTURE_2D 0
200 -- animate
modifyIORef ifsR $ \ifs' ->
    let w_frame' = w_frame ifs' + 1
    in ifs'
        { w_frame = w_frame'
205          , w_phase = fromIntegral w_frame' * 4 * pi / fromIntegral looplevelength
        }
swapBuffers
reportErrors

210 -- timer callback (25fps)
timer :: IORef IFS -> IO ()
timer ifsR = do

```

```

    addTimerCallback 40 (timer ifsR)
    postRedisplay Nothing
215
-- window reshape callback
reshape :: IORef IFS -> Size -> IO ()
reshape ifsR (Size w h) = modifyIORef ifsR $ \ifs -> ifs
    { w_width = fromIntegral w, w_height = fromIntegral h }
220
-- compile a shader
shader :: String -> IO (Either String GLuint)
shader frag = do
    p <- glCreateProgram
225    f <- glCreateShader GL_FRAGMENT_SHADER
    withCString frag $ \s -> with s $ \src -> do
        glShaderSource (fromIntegral f) 1 (castPtr src) nullPtr
        glCompileShader (fromIntegral f)
        glAttachShader p f
230    glLinkProgram (fromIntegral p)
    alloca $ \sp -> do
        glGetProgramiv (fromIntegral p) GL_LINK_STATUS sp
        success <- peek sp
        if (success == 0)
235            then
                alloca $ \lp -> do
                    glGetProgramiv (fromIntegral p) GL_INFO_LOG_LENGTH lp
                    l <- peek lp
                    poke lp 0
240                    allocaArray (fromIntegral l) $ \lg -> do
                        glGetProgramInfoLog (fromIntegral p) (fromIntegral l) (castPtr lp)
                        ↪ lg
                        l' <- peek lp
                        ss <- peekCStringLen (castPtr lg, fromIntegral l')
                        return (Left ss)
245            else return (Right (fromIntegral p))

-- generate a texture
newTexture :: Int -> IO GLuint
newTexture size = alloca $ \pt -> do
250    glGenTextures 1 pt
    t <- peek pt
    glBindTexture GL_TEXTURE_2D t
    let s = fromIntegral size
    glTexImage2D GL_TEXTURE_2D 0 (fromIntegral GL_RGBA) s s 0
255    GL_RGBA GL_UNSIGNED_BYTE (nullPtr :: Ptr GLubyte)
    glTexParameterf GL_TEXTURE_2D GL_TEXTURE_MIN_FILTER (fromIntegral ↵
        ↪ GL_LINEAR_MIPMAP_LINEAR)
    glTexParameterf GL_TEXTURE_2D GL_TEXTURE_MAG_FILTER (fromIntegral GL_LINEAR)
    glTexParameterf GL_TEXTURE_2D GL_TEXTURE_WRAP_S (fromIntegral GL_CLAMP_TO_EDGE ↵
        ↪ )
    glTexParameterf GL_TEXTURE_2D GL_TEXTURE_WRAP_T (fromIntegral GL_CLAMP_TO_EDGE ↵
        ↪ )
260    glBindTexture GL_TEXTURE_2D 0
    return t

-- keyboard callback
keyboard :: IORef IFS -> KeyboardMouseCallback
265 keyboard - (Char 'f') Down - = fullScreenToggle

```

```

keyboard - (Char _) Down _ _ = exitSuccess
keyboard _ _ _ _ = return ()

-- main program
270 main :: IO ()
main = do
    let w = 512
        h = 288
    initialWindowSize $= Size (fromIntegral w) (fromIntegral h)
275 initialDisplayMode $= [ RGBMode, DoubleBuffered ]
    (_, args) <- getArgsAndInitialize
    quality <- case map reads args of
        [[(q,"")] | 5 < q && q < 15-> return q
        _ -> return (9 :: Int)
280 let tsize' = 2 ^ quality
    _ <- createWindow "GearBox"
    -- compile shaders (FIXME: print log to stderr on failure and exit)
    Right p_shift' <- shader shift_frag_src
    p_shift_u_t' <- withCString "t" $ glGetUniformLocation p_shift' . castPtr
285 p_shift_u_k' <- withCString "k" $ glGetUniformLocation p_shift' . castPtr
    Right p_gear' <- shader gear_frag_src
    -- misc setup
    glHint GL_GENERATE_MIPMAP_HINT GL_NICEST
    glEnable GL_TEXTURE_2D
290 glEnable GL_BLEND
    glBlendFunc GL_SRC_ALPHA GL_ONE_MINUS_SRC_ALPHA
    glClearColor 0.25 0.25 0.5 0
    -- allocate textures
    [t_source', t_ping', t_pong'] <- replicateM 3 (newTexture tsize')
295 -- allocate frame buffer
    fbo' <- alloca $ \p -> do
        glGenFramebuffers 1 p
        peek p
    -- initialize state
300 ifsR <- newIORef IFS
    { fbo = fbo', t_source = t_source', t_ping = t_ping', t_pong = t_pong'
      , p_shift = p_shift', p_shift_u_t = p_shift_u_t', p_shift_u_k = p_shift_u_k'
      , w_width = w, w_height = h, w_tsize = tsize'
      , w_frame = 0, w_phase = 0
305 }
    -- render the gear texture
    glViewport 0 0 (fromIntegral tsize') (fromIntegral tsize')
    glMatrixMode GL_PROJECTION
    glLoadIdentity
310 glOrtho (-1) 1 (-1) 1 (-1) 1
    glMatrixMode GL_MODELVIEW
    glLoadIdentity
    glBindFramebuffer GL_FRAMEBUFFER fbo'
    glFramebufferTexture2D GL_FRAMEBUFFER GL_COLOR_ATTACHMENT0 GL_TEXTURE_2D ↵
        ↵ t_source' 0
315 glClear (fromIntegral GL_COLOR_BUFFER_BIT)
    glUseProgram (fromIntegral p_gear')
    unitQuad
    glUseProgram 0
    glFramebufferTexture2D GL_FRAMEBUFFER GL_COLOR_ATTACHMENT0 GL_TEXTURE_2D 0 0
320 glBindFramebuffer GL_FRAMEBUFFER 0
    glBindTexture GL_TEXTURE_2D t_source'

```



```

    glGenerateMipmap GL_TEXTURE_2D
    glBindTexture GL_TEXTURE_2D 0
    -- set callbacks
325  reshapeCallback $= Just (reshape ifsR)
    displayCallback $= display ifsR
    addTimerCallback 40 (timer ifsR)
    keyboardMouseCallback $= Just (keyboard ifsR)
    reportErrors
330  mainLoop

```

3 .gitignore

```

dist*/
.cabal-sandbox
cabal.sandbox.config
.ghc.env*

```

4 LICENSE

GNU GENERAL PUBLIC LICENSE Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>
 5 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

Preamble

10 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,
 15 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
 20 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
 25 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
 30 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
 35 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.

40 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.

45 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.

50 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
55 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.

60 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
65 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.

70 TERMS AND CONDITIONS

0. Definitions.

75 "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.

80 "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.

85 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.

90 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
95 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.

100 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.

105 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.

115 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.

120 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.

125 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.

135 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.

150 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.

210 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:

215 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
220 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
225 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.

230 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.

235 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
240 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.

245 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,
250 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
255 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
260 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
265 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.

270 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.

275 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
280 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
285 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
290 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
295 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
300 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
305 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.

310 "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
315 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
320 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

495 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.

500 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
505 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
510 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
515 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.

520 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
525 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
530 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.

535 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.

540 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
545 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
550 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/>>.

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

5 Setup.hs

```
import Distribution.Simple
main = defaultMain
```