&

wxHaskell

Jaak Ristioja
Tartu 2009

Gtk2Hs

- Arenduses vähemalt aastast 1999
- Ehitatud GTK+ (The Gimp Toolkit) baasil
  - Keeles C
  - LGPL
  - Arenduses alates 1997. aastast
  - X Window System, Microsoft Windows, MacOS X...

Gtk2Hs

- `initGUI`
- Blokeeruv `mainGUI` (event loop)
  - Callbacks
    - Kui töötleme kaua, GUI hangub
  - Taimerid
- Keeruline hoida olekut
  - IORef, Mvar
- Lõimed aitavad

Gtk2Hs

- Automaatne mäluhaldus
- Glade (via XML)
- Vidinad
  - Tüübid
    - Aknad
    - Konteinerid
    - Vidinad
  - Luuakse IO monaadis:

```
window <- windowNew
```

Gtk2Hs

- Vidinate hierarhia tüüpide ja tüübiklassidega:

```
class WidgetClass w => ButtonClass w
data Button
instance WidgetClass Button
instance ButtonClass Button
```

- Enamus operatsioone kasutab tüübiklasse:

```
widgetShow :: WidgetClass w => w -> IO ()
```

Gtk2Hs

- Atribuutid kontrollivad vidinate välimust, olekut ja käitumist.
  - Muutmine

```
set widget [ widgetAttr1 := foo,
             widgetAttr2 :~ \x -> f x ]
```

  - Lugemine

```
x <- get widget widgetAttr1
```

Gtk2Hs

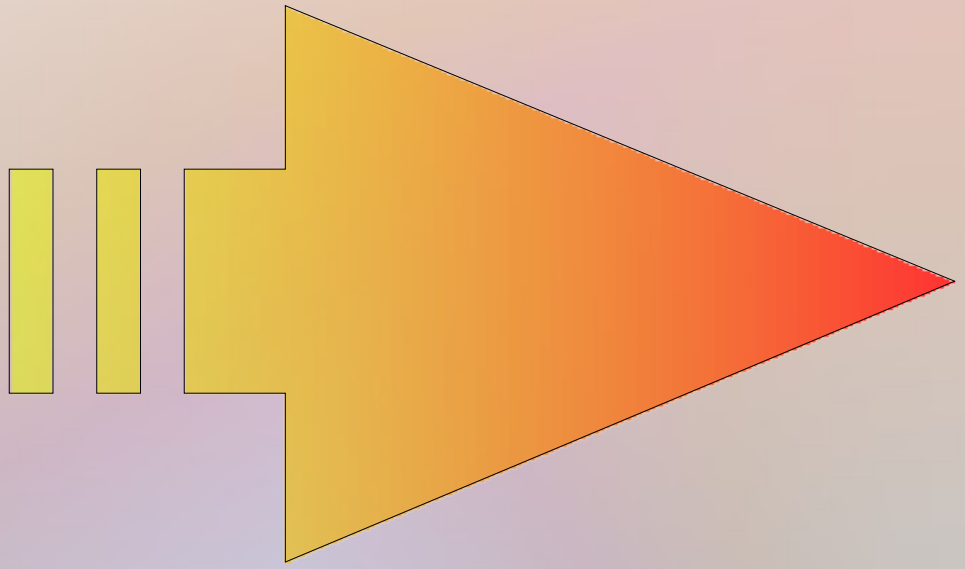
- Signaale saadetakse vidina olekute muutumisel või kasutaja sisendil
- Signaali vastuvõtvad sulundid on IO tüüpidega:

```
onClicked button $ do ...
```

- Mõned signaalid võtavad vastu parameetreid ja tagastavad väärtusi:

```
onExpose canvas $
  \Expose { eventRegion = region } -> do
  ...
  return True
```

Hello World!



DEMO(D)

Gtk2Hs: Paigutus

- Pakitakse horisontaalsetesse või vertikaalsetesse kastidesse (box)
  - hBoxNew võrdseltRuumi piksleidVahel
  - boxPack* box widget tüüp lisaruum
    - boxPackStart – ülevalt alla, vasakult paremale
    - boxPackEnd – alt üles, paremalt vasakule

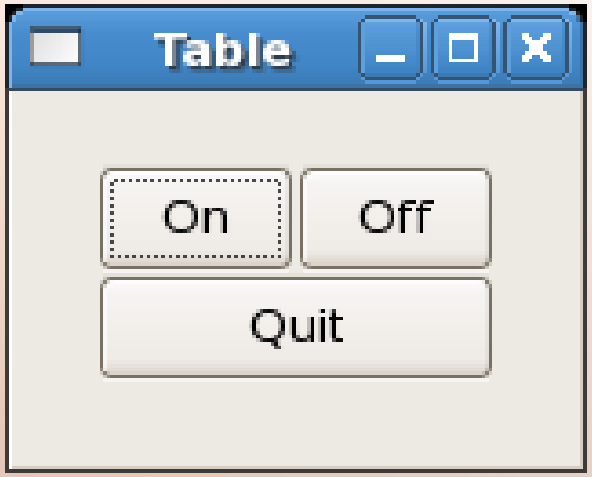
```
-- hBoxNew :: Bool -> Int -> IO Hbox
hbox <- hBoxNew True 10
set window [containerBorderWidth := 10, containerChild := hbox]

-- boxPack* :: (WidgetClass child, BoxClass self) => self -> child -> Packing -> Int -> IO ()
boxPackStart hbox button1 PackGrow 0
boxPackStart hbox button2 PackGrow 0
```

Gtk2Hs: Paigutus

- Pakitakse horisontaalsetesse või vertikaalsetesse kastidesse (box)
  - hBoxNew võrdseltRuumi piksleidVahel
  - boxPack* box widget tüüp lisaruum
    - boxPackStart – ülevalt alla, vasakult paremale
    - boxPackEnd – alt üles, paremalt vasakule

```
-- hBoxNew :: Bool -> Int -> IO Hbox
hbox <- hBoxNew True 10
set window [containerBorderWidth := 10, containerChild := hbox]

-- boxPack* :: (WidgetClass child, BoxClass self) => self -> child -> Packing -> Int -> IO ()
boxPackStart hbox button1 PackGrow 0
boxPackStart hbox button2 PackGrow 0
```

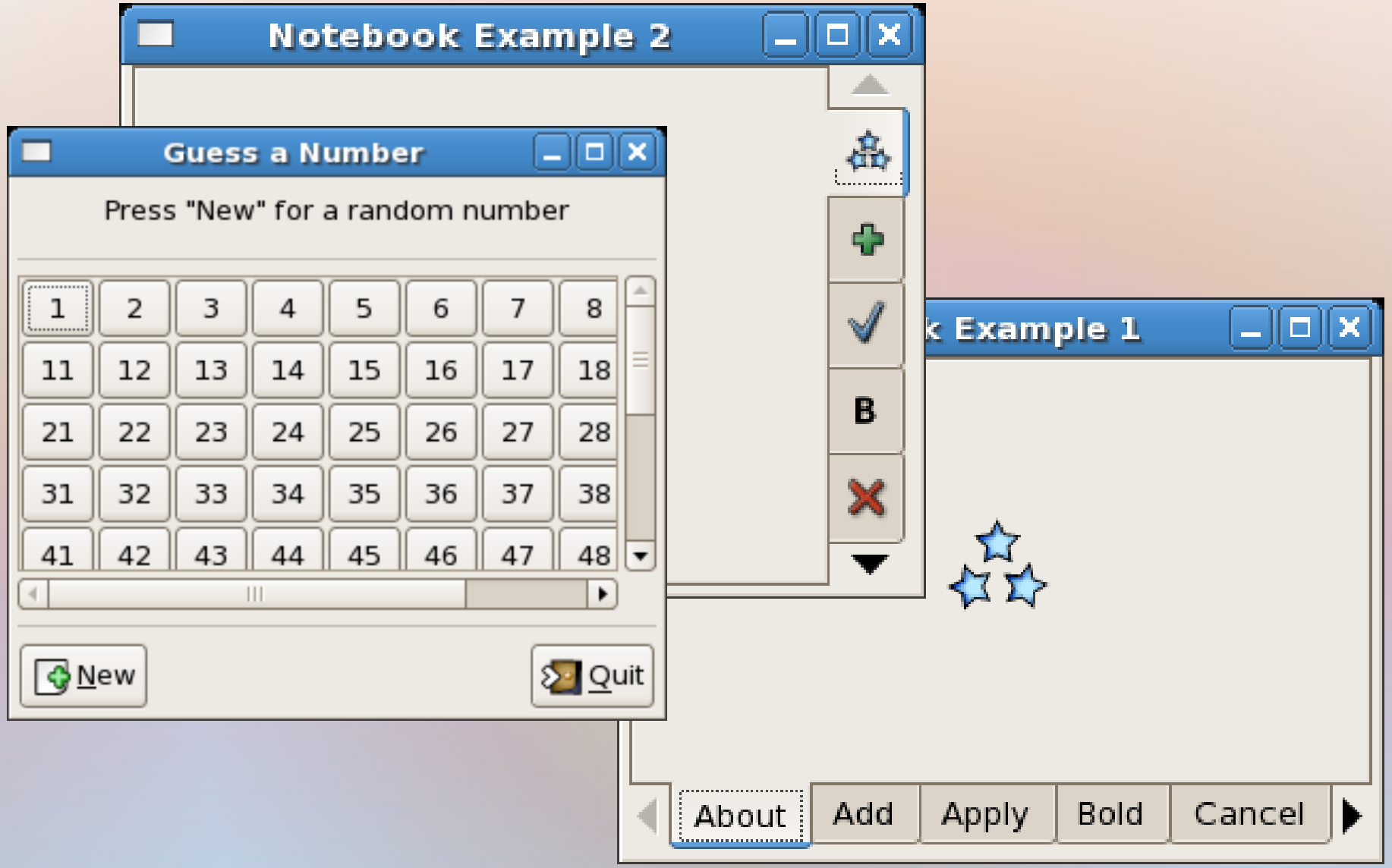
Gtk2Hs: Paigutus (2)



- Pakkimine tabelisse

```
-- tableNew :: Int -> Int -> Bool -> IO Table
--             ridu   veerge homogeenne

tabel <- tableNew 2 2 False

-- tableAttach :: (WidgetClass child, TableClass self) =>
                  self -> child -> Int -> Int -> Int -> Int
                  -> [AttachOptions] -> [AttachOptions]
                  -> Int -> Int -> IO ()
-- tabel, widget
-- vasak äär, parem äär, ülemine äär, alumine äär
-- (Expand, Shrink, Fill) hor. ja vert. Suurendamisel
-- hor. ja vert. ääris widgetile (0)


tableAttachDefaults tabel nupp 0 1 0 2
```

Gtk2Hs: Paigutus (3)



Gtk2Hs: Nupud

```
-- buttonNewWithLabel :: String -> IO Button
-- buttonNewWithMnemonic :: String -> IO Button
-- buttonNew :: IO Button
nupp <- buttonNew
containerAdd nupp sisu
```

Gtk2Hs: Nupud

```
-- buttonNewWithLabel :: String -> IO Button
-- buttonNewWithMnemonic :: String -> IO Button
-- buttonNew :: IO Button
nupp <- buttonNew
containerAdd nupp sisu
```

Gtk2Hs: Nupud (2)

- Signaalid
  - onPressed
  - onReleased
  - onClicked
  - onEnter
  - onLeave

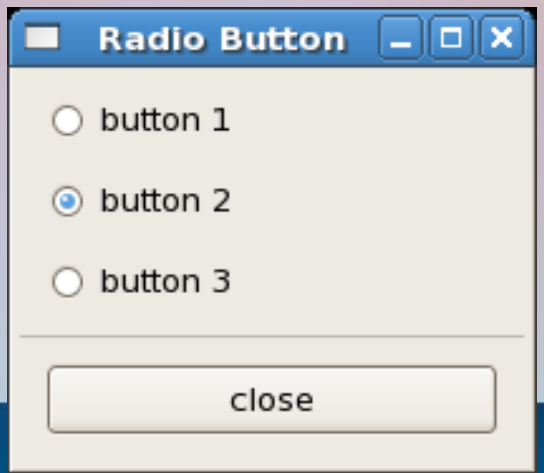
Gtk2Hs: Nupud (3)

```
toggleButtonNew :: IO ToggleButton
toggleButtonNewWithLabel :: String -> IO Togglebutton
toggleButtonNewWithMnemonic :: String -> IO ToggleButton
toggleButtonGetActive :: ToggleButtonClass self => self
                                                -> IO Bool
toggleButtonSetActive :: ToggleButtonClass self => self
                                                -> Bool
                                                -> IO ()
```

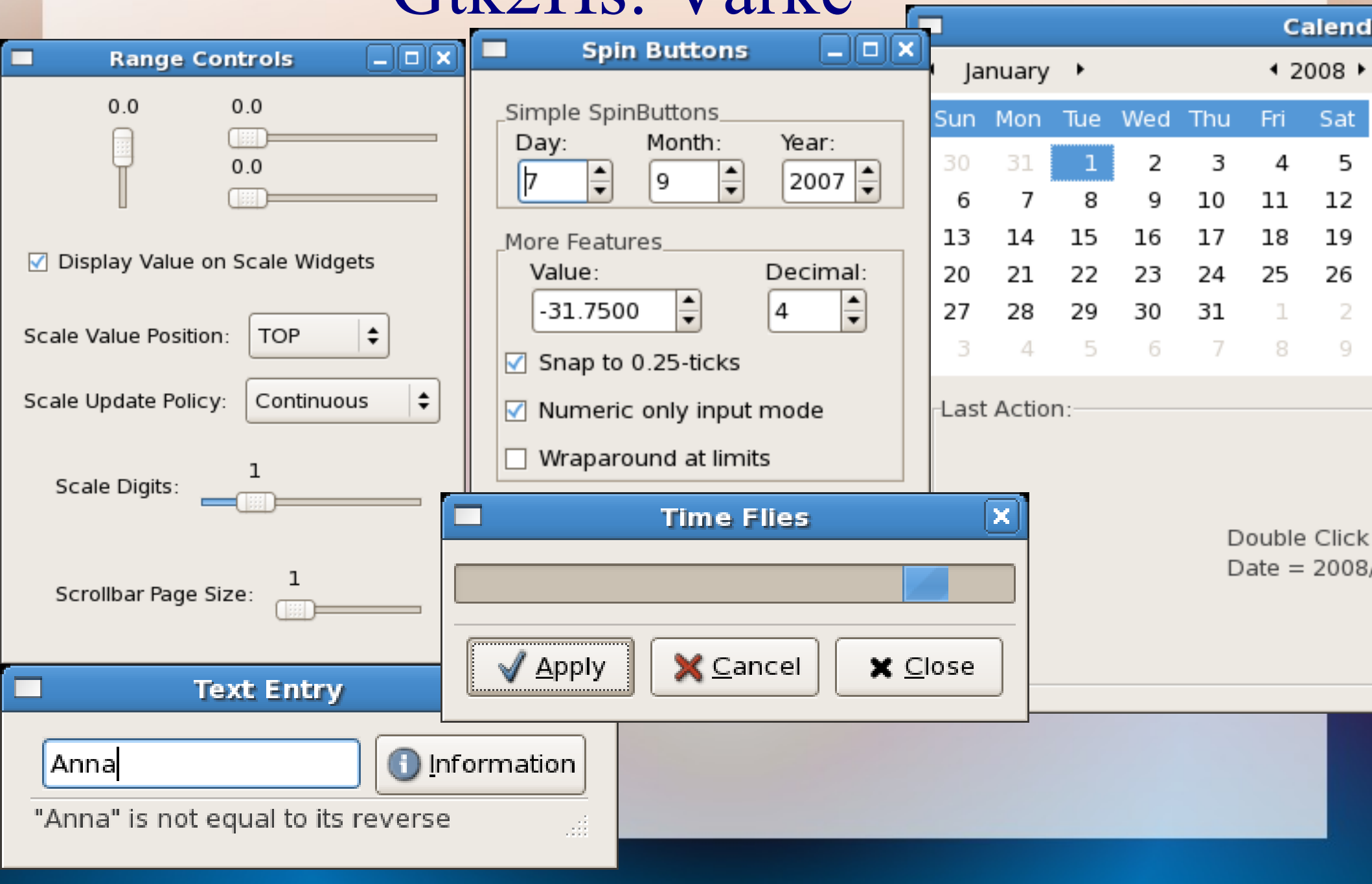
- Sama `checkBox` puhul
- onToggled

Gtk2Hs: Nupud (4)

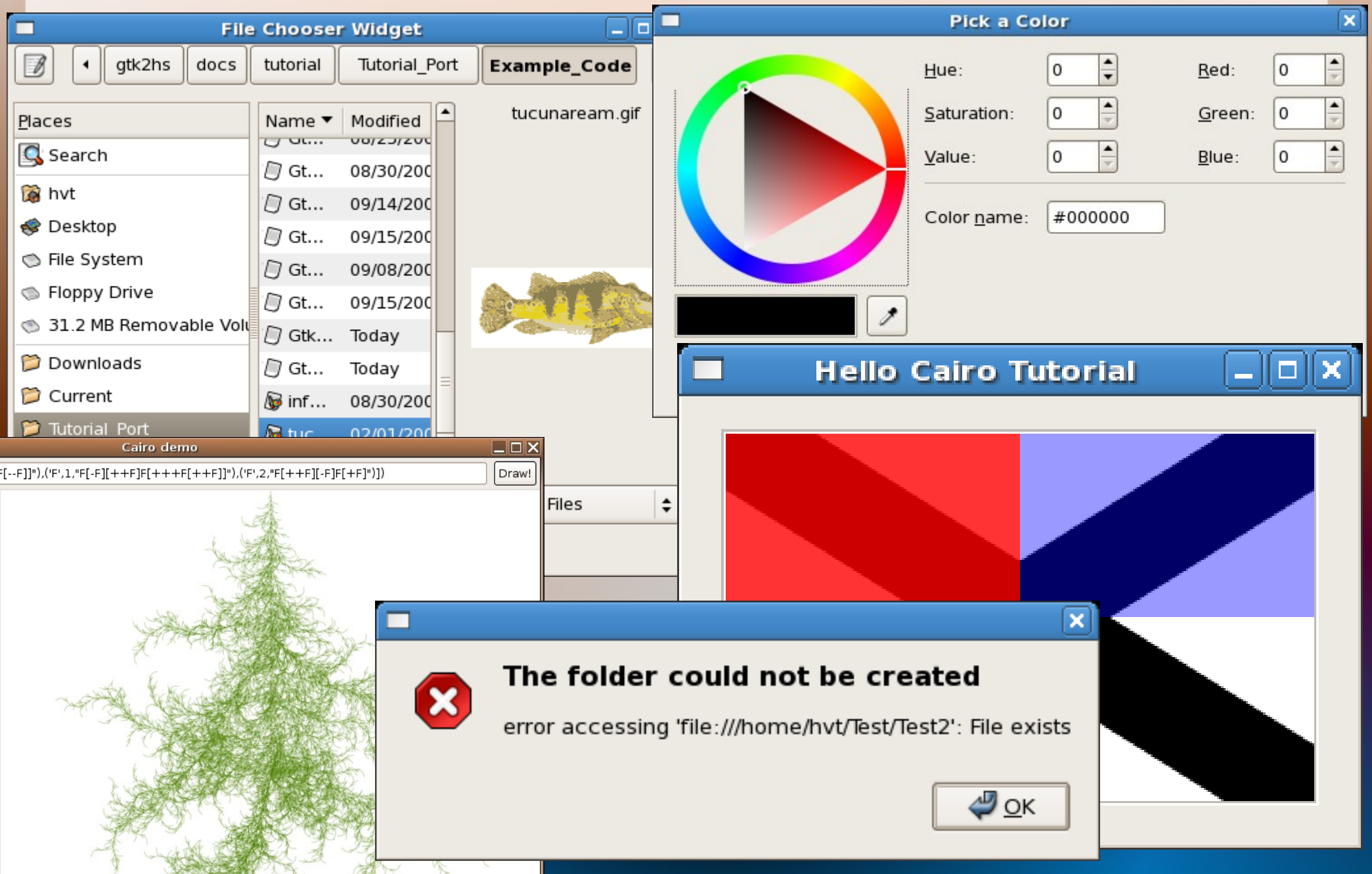
```
radioButtonNew :: IO RadioButton

radioButtonNewWithLabel :: String -> IO RadioButton

radioButtonNewWithMnemonic :: String -> IO RadioButton

radioButtonNewFromWidget :: RadioButton -> IO RadioButton

radioButtonNewWithLabelFromWidget :: RadioButton
                                  -> String
                                  -> IO RadioButton

radioButtonNewWithMnemonicFromWidget :: RadioButton
                                     -> String
                                     -> IO RadioButton
```



Gtk2Hs: Värke



Gtk2Hs: Veel värke



Gtk2Hs

- Cairo (2D vektorgraafika + PS/PDF/PNG jne väljundid)
- Pango (Unicode'i renderdamine)
- Gnome' integratsioon (GConf jms)
- GStreamer
- SVG (laiendus Cairo'le)
- Mozilla HTML renderdamine vidinas
- „SourceView“ süntaksit värviv redaktorvidin
- jpm

wxHaskell

- Arenduses vähemalt aastast 2003
- Ehitatud wxWidgets (algne wxWindows) baasil
  - Keeles C++
  - WxWindows Library Licence
    - Põhimõtteliselt LGPL mõne erandiga
    - Siiski vaba tarkvara (OSI approved)
  - Arenduses alates 1992. aastast
  - X Window System, Microsoft Windows, MacOS X...
- WXCore – madalatasemeline (üks-ühele C++'ga)
- WX – kõrgetasemelised abstraktsioonid

wxHaskell

- Automaatne mäluhaldus
- Hierarhia probleem
  - Kõik konstruktorid võtavad argumendiks vanem-elemendi
- Ei toeta hästi lõimi
- Käivitatakse IO () funktsioon

```
-- start :: IO a -> IO ()
main = start hello
hello :: IO ()
hello = do
       f <- frame [ text := "Hello!" ]
       ...
```

wxHaskell

- Vidinad (atribuudid sarnaselt Gtk2Hs)

```
f <- frame []
b <- button f [ attr1 := value ]
```

- Hierarhia
  - Iga C++ klassi kohta fantoom-andmetüüp

```
data CWindow a
```

  - Hierarhia

```
type Window a = Object (CWindow a)
```

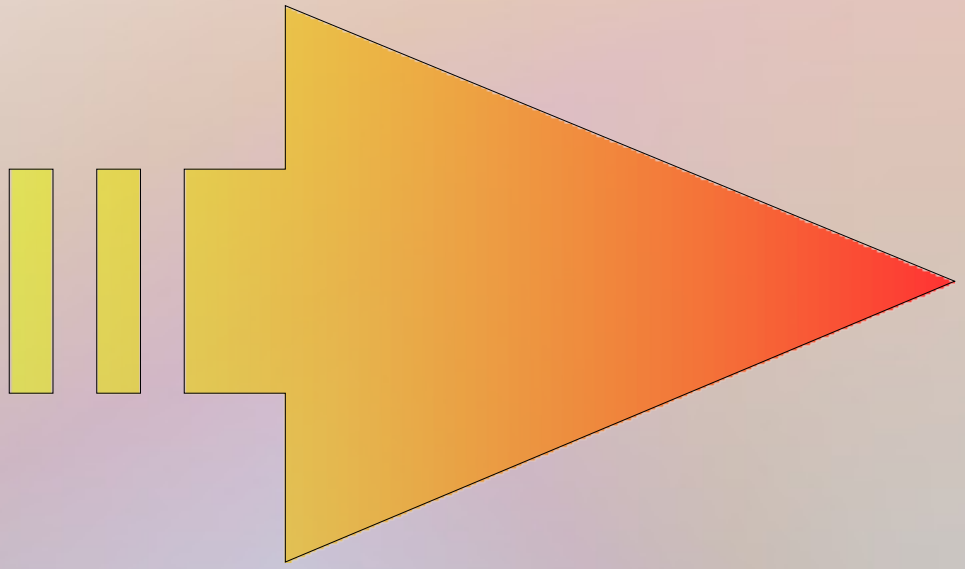
  - `type Control a = Window (CControl a)`
  - `type Button a = Control (CButton a)`

wxHaskell

- Signaale saadetakse vidina olekute muutumisel või kasutaja sisendil
- Signaali vastuvõtvad sulundid on IO () tüübiga:

```
set button [ on command := mingivärk ]
-- on :: Event w a -> Attr w a
-- command :: (Commanding w) => Event w (IO ())
-- paint :: (Paint w) => Event w (
--                 DC () -> Rect -> IO ()
--           )
```

Hello World!

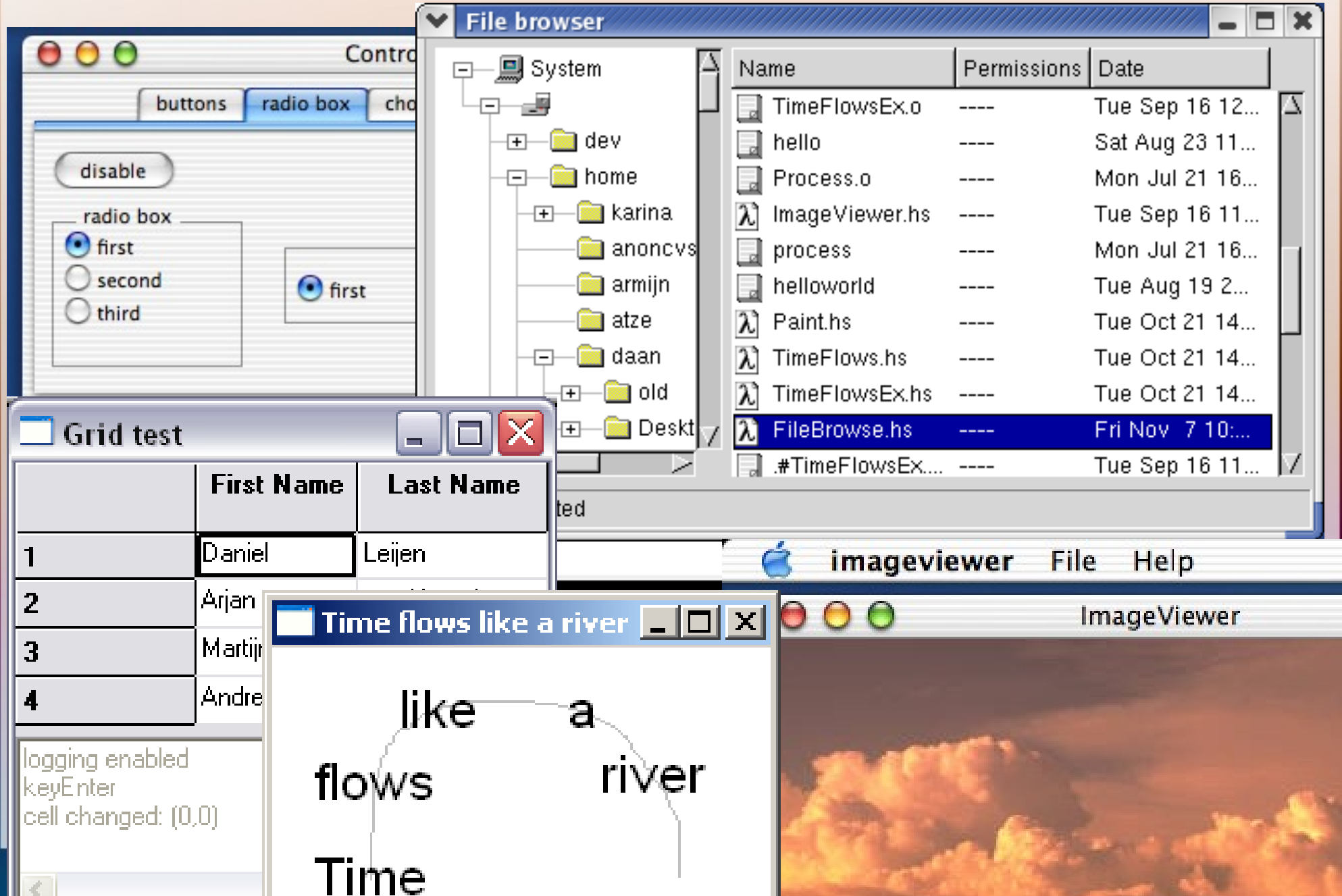


DEMO(D)

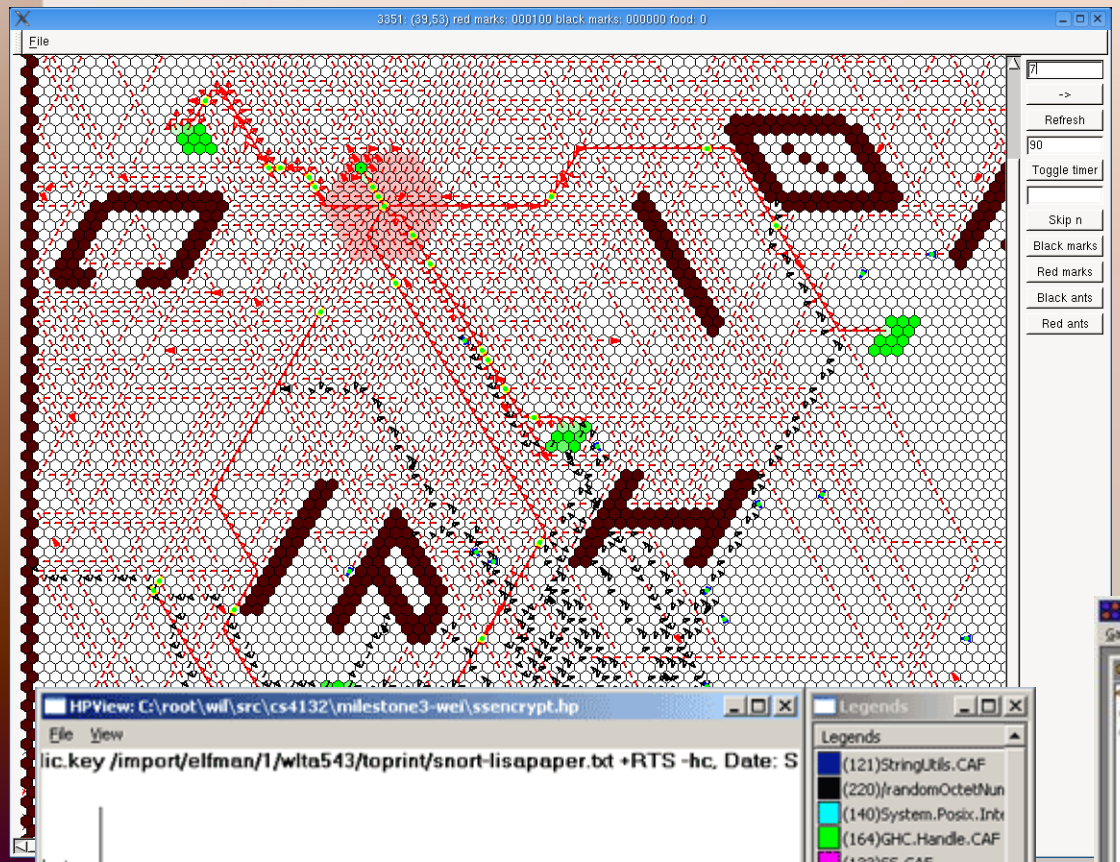
wxHaskell: Paigutus

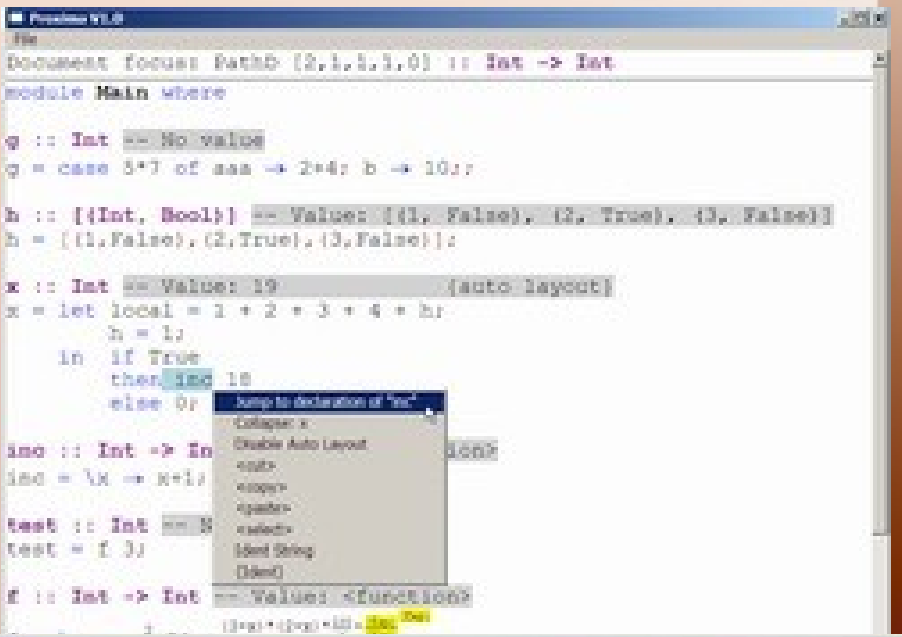
```
set f [ layout := margin 10 (column 5 [
        floatCentre (label "Hello"),
        floatCentre (widget quit)
]) ]
-- margin :: Int -> Layout -> Layout
-- floatCentre :: Layout -> Layout
-- row :: Int -> [Layout] -> Layout
-- widget :: (Widget w) => w -> Layout
-- label :: String -> Layout
-- hfill :: Layout -> Layout
-- hrule :: Int -> Layout
-- glue :: Layout
-- floatBottomRight :: Layout -> Layout
```

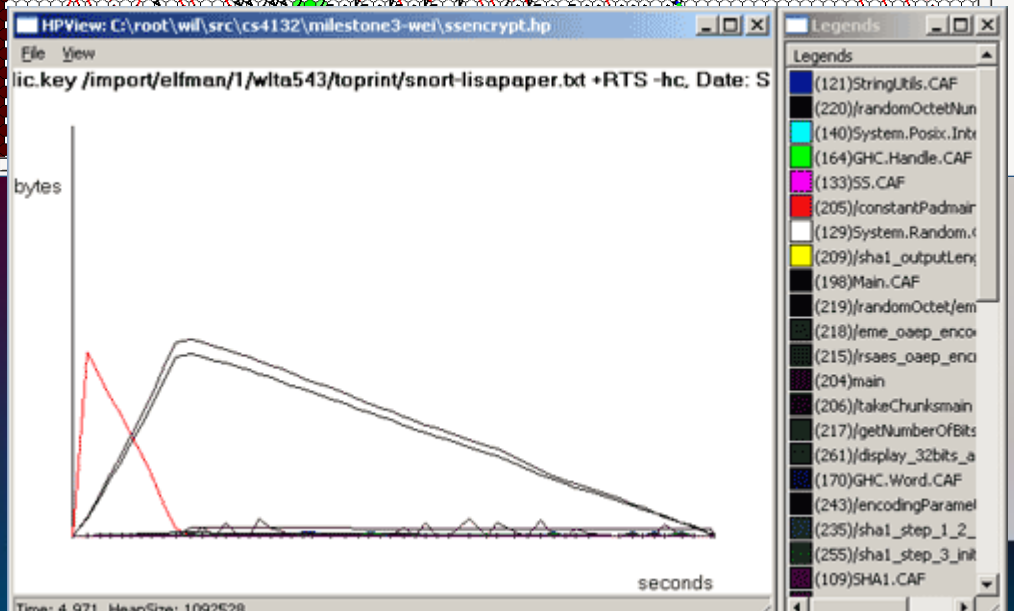
wxHaskell: Värke

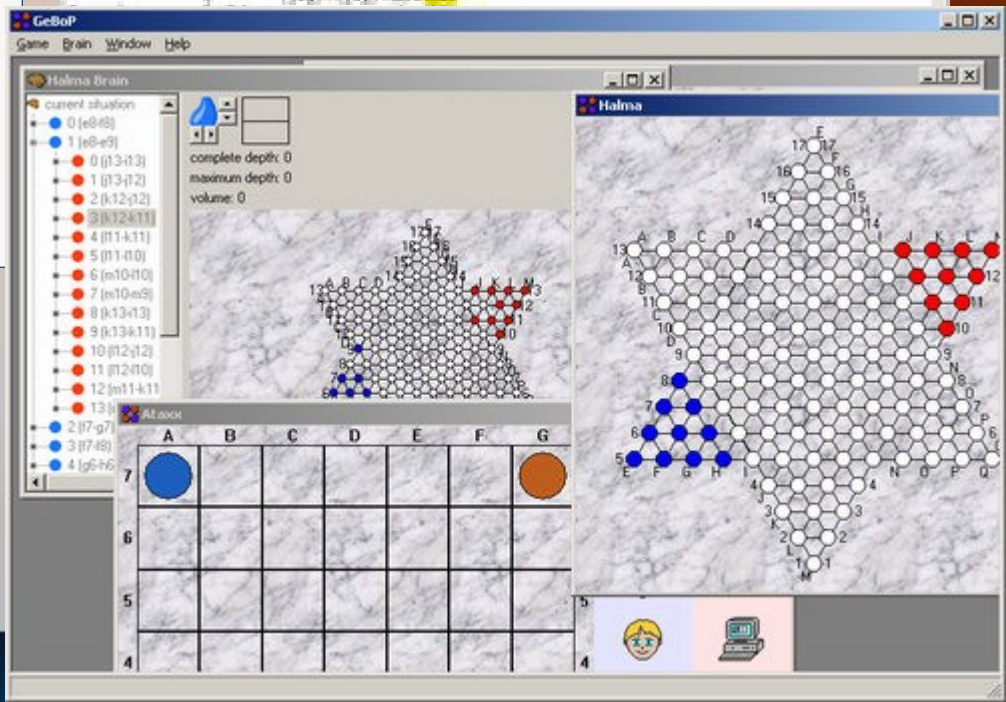


wxHaskell: Veel värke









Küsimusi?

LÕPP