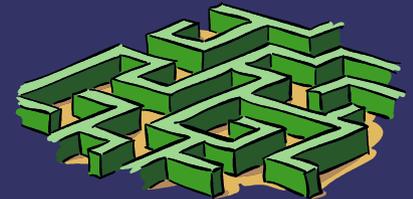


# *Status of my open source development*

Gregg Young



# Overview

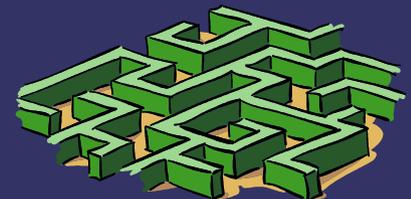
## ⇒ Open Source projects

- Lucide
- FM/2
- Lswitcher
- eFTE
- Multimedia Plugins
- Other



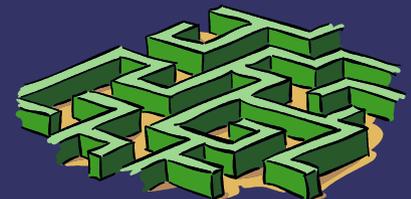
# *Lucide What's New*

- ➔ Added a help file that is context sensitive
- ➔ Updated to the latest Poppler and DjVU
- ➔ Added the GMB plugin to the distribution package
  - Thanks to Heiko Nitzsche
- Also update the JPEG plugin but recommend you use the GMB plugin instead.



# *Lucide What's New*

- Gave it its own INI file and remember many more settings
- Interface cleanup
- Improved Thumbnail handling
- Reverse order printing
- List font substitutions
- Exceptq support



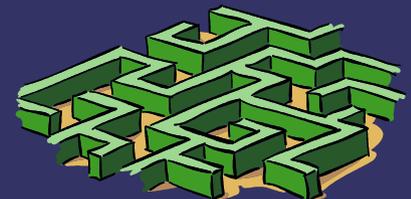
# *Lucide What's Needed*

- ⇒ Annotation support
- ⇒ Improved form fill-in support
- ⇒ Improved printer support
- ⇒ Better localization support
- ⇒ Other?



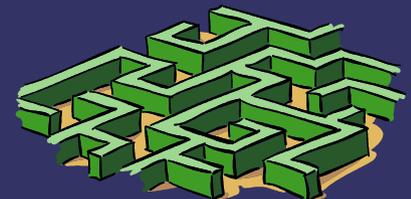
# *FM/2*

- ➔ Haven't fixed anything recently
- ➔ Is anyone but me actually using it?
- ➔ It needs to have some interface issues and random traps fixed
- ➔ Maybe I should do something about these.



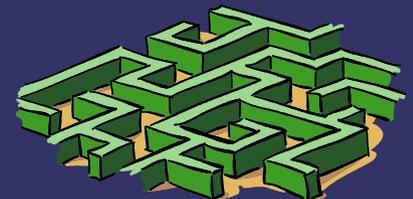
# *ISwitcher*

- ➔ Mostly fixed the disappearing window problem
- Fixed a conflict with Xworkplace's “Sticky windows”
- Added context sensitive help
- Needs to save settings to INI file individually not as struct to avoid version compatibility issues.
- Probably should fix the last and release it.



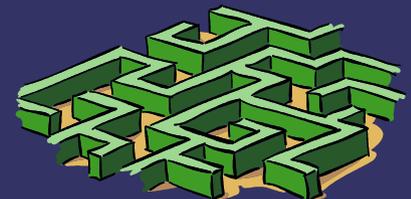
# eFTE

- ⇒ Nothing much fixed
- ⇒ Need to fix 4 rare traps/hangs
- ⇒ Need to fix some configuration file issues
- ⇒ Develop EPM and E personalities so it can be use as a replacement for them
  - Needs to use OS/2system colors



# *Multimedia Plugins*

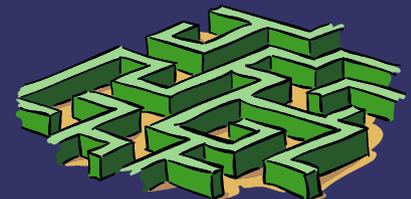
- ➔ Have Flac building with latest lib
- ➔ Have Vorbis and Ogg building with latest lib
- ➔ High sample rate Flacs fail
- ➔ Flac plugin crashes trying to convert Vorbis file to Flac.



# Other

## ⇒ Wipfc

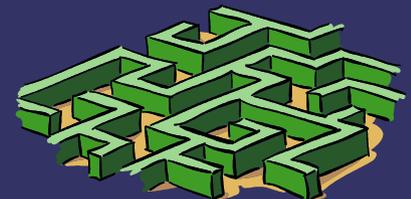
- Fixed some issues revealed by the Lucide help file
- This has never actually been released
- If you want it, it is in the tools directory of the Lucide SVN
- Should probably release a build of the latest unforked Open Watcom code



# *How can I help development?*

## ⇒ Writing code

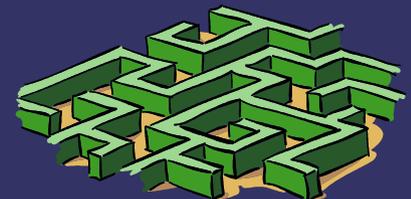
- Actual “programming”
- Scripts to automate repetitive tasks
- Updating documentation
- Installers
- Release updating
- Building
- Code cleaning



# *“Actual Programing”*

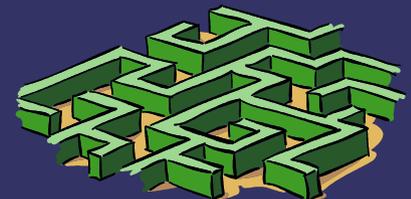
## ➔ Novice Programmer

- You really can't break it
  - SVN/CVS
  - Revert
  - SmartSVN/CNS
- Find a mentor
  - Be honest about your skill level
  - Take responsibility for what you break
  - Learn from your mistake; we all make them



# *“Actual Programming”*

- ➔ So what if it takes you 5 times longer
  - You are learning
  - If you don't do it someone else has to.
- ➔ Jobs exist for all skill levels
  - An example usually exists in the code already
  - Changes to menu or dialog wording only requires finding the present wording.
  - Creating a function from repetitive code is mostly cut & paste.



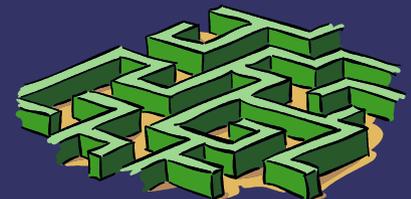
# *Scripts to automate repetitive tasks*

- ➔ Many people who don't consider themselves programmers write scripts
- ➔ These speed development
- ➔ Make doing releases easier
  - More frequent
  - Less time wasted



# *Updating documentation*

- ➔ Many programmers don't write good documentation
- ➔ The programmer rarely has extensively used every feature
- ➔ Documentation is best done by a user with extensive experience with the feature.
  - Speaks the same language as other users
  - Knows how the feature actually works.
- ➔ eFTE can actually add most of the IPF codes for you



# *Writing code - Other*

## ➔ Installers

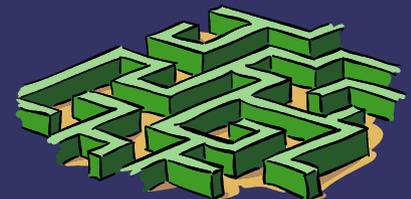
- Many programs suffer from a lack of a straight forward way to install.
- Warpin
- RPM

## ➔ Preparing the release or the new release goes how many places?

## ➔ Building

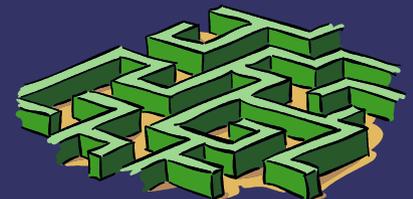
- Having extra people who can get the code and build offers additional opportunity for testing.

## ➔ Cleaning Code



# *What Other Jobs Need to Be Done?*

- ➔ Management of web site
  - Bug tickets
  - Milestone planning
  - Documentation for:
    - Testing
    - Building
    - Releases
    - FAQ
    - Usage tips.



# *What Other Jobs Need to Be Done?*

- ➔ Project management
  - Labor division (herding cats)
  - Prioritizing tickets
  - Setting goals
  - Time lines
  - Persuading unpaid people to do what needs doing
- ➔ Testing
  - Hey it compiled without an error



# *What Other Jobs Need to Be Done?*

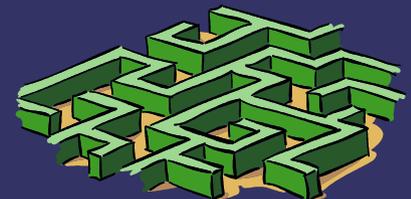
## ➔ Bug reports

- First, make sure you have the current release of the software.
- Read the documentation for the software you are using. There is a lot of good information and it just might give the fix for your problem.
- Is your problem reproducible?
- Is there something odd about your configuration?
- Are you using the latest drivers?
- Fix or test these issues before reporting



# *When opening a ticket*

- ➔ One problem per ticket. Don't usurp someone's ticket
- ➔ Choose a reasonable summary that is useful and briefly describes the problem.
- ➔ Provide a clear, easy to understand description of the problem. Don't assume that the developer magically knows what you are doing. Be specific about the hardware you are using (e.g., “XYZ USB 2.0 hub” and not “the box”).



# *When opening a ticket*

- ➔ Describe in detail how to reproduce the problem. This is very important. If the developer cannot reproduce your problem, the chances of it getting fixed are near zero.



# *Questions*

