John Romero Programming Proverbs

- 10. “Try to code transparently. Tell your lead and peers exactly how you are going to solve your current task and get feedback and advice. Do not treat game programming like each coder is a black box. The project could go off the rails and cause delays.”

- John Romero, “The Early Days of Id Software - John Romero @ WeAreDevelopers Conference 2017”

Creating a Debian Game Package

- these notes do not include actually writing the game!
  - the game is assumed to be written and full debugged
- all we are considering is how best to package the source, data
  - and how to install the game sensibly

Why use packages at all?

- in the past source code was downloaded and built from `foo.tar.gz` (often called tarballs)
  - some distributions still use this mechanism
  - normally an activity reserved for enthusiasts or developers
- although occasionally experienced GNU/Linux users will do this to work around a bug or dependency problem
- Debian package management software is superb

Advantage of Debian packages

- can reference multiple repositories in `/etc/apt/sources.list`

-
```
deb http://ftp.uk.debian.org/debian/ stretch main contrib non-free
deb-src http://ftp.uk.debian.org/debian/ stretch main contrib non-free

deb http://security.debian.org/ stretch/updates main contrib non-free
deb-src http://security.debian.org/ stretch/updates main contrib non-free

# deb http://www.debian-multimedia.org stretch main non-free

#
#  GNU Modula-2 repo
#
deb http://floppsie.comp.glam.ac.uk/debian/ stretch main
deb-src http://floppsie.comp.glam.ac.uk/debian/ stretch main
```

Advantage of Debian packages

- possible to request a machine to update its complete list of packages and install updates
  - each time it boots up
  - this is done in the labs, and nobody notices the updates
  - contrast this with shutting down a Windows client :-)

Advantage of Debian packages

- a well formed Debian package allow you to
  - rebuild it with minimal effort
  - obtain the source

- rev the source and rebuild
  - possibly support multiple architectures
  - includes all correct package dependencies

Example

- ```
  $ sudo apt-get update
  ```
- ```
  $ sudo apt-get install halma
  ```
- installs the package `halma`
  - it will also ensure the dependencies Python/PyGame are installed
- ```
  $ sudo apt-get remove halma
  ```
  ```

Example

- ```
  $ apt-get source halma
  ```

- obtains the current source for the package `halma`
  ```

Halma package contents

```
$ apt-get source halma
$ ls -ld halma*
drwxr-xr-x 6 fred  fred   4096 Mar 17  2011 halma-0.0.1
-rw-r--r-- 1 fred  fred    524 Nov 22 18:12 halma_0.0.1-1.10.dsc
-rw-r--r-- 1 fred  fred  94023 Nov 22 18:13 halma_0.0.1-1.10.tar.gz
```

- `halma_0.0.1-1.10.dsc` a description of the package, including version, maintainers email address, dependencies, tarball filename

Halma package contents

-
```
$ ls halma-0.0.1
config.guess  configure      data            desktop
Makefile      version.c      config.sub      configure.in
debian        gameEngine.py  gui.py          halma.in
halma.mod     Makefile.in    pixmaps
```

- examine these files during lab times

Package related files

- held in the directory `debian`

```
$ ls -ltr halma-0.0.1/debian/
-rw-r--r-- 1 fred  fred    18 Mar 15  2011 dirs
-rwxr-xr-x 1 fred  fred   441 Mar 16  2011 rules
-rw-r--r-- 1 fred  fred    91 Mar 16  2011 menu
-rw-r--r-- 1 fred  fred   764 Mar 17  2011 control
-rw-r--r-- 1 fred  fred  1886 Mar 17  2011 changelog
```

dirs

- ```
  usr/bin
  usr/share
  ```

- details the subdirectories where the contents will reside once installed
  ```

menu

```
$ cat halma-0.0.1/debian/menu
?package(halma):needs="X11" section="Games/Toys" \
  title="halma" command="/usr/bin/halma"
```

ChangeLog

```
$ cat halma-0.0.1/debian/changelog
 -- Fred <fred@somewhere.org>  Wed, 16 Mar 2011 20:47:24 +0000

halma (0.0.1-1.4) unstable; urgency=low

  * Non-maintainer upload.
  * corrected menu entry.
```

Control

```
$ cat halma-0.0.1/debian/control
Source: halma
Section: games
Priority: extra
Maintainer: Fred <fred@somewhere.org>
Build-Depends: debhelper (>= 5), autotools-dev, dbs (>=0.22), texinfo, \
gcc, python, python-pygame, gm2
Standards-Version: 3.7.2

Package: halma
Architecture: any
Depends: ${shlibs:Depends}, ${misc:Depends}, python-pygame, python-pexpect
Suggests:
Conflicts:
Description: The Victorian board game Halma.
 A two player game where the object is to move your pieces into the
 base of the opposing player.  In this variant of the game (Kangaroo Halma)
 you may jump multiple squares so long as both sides are symmetrical.
 A tactical game which typically lasts between 10 and 15 minutes.  You
 play red, the computer plays blue and the computer will make a move
 every 10 seconds.
```

Building the package

- further reading ⟨http://www.nongnu.org/gm2/creating_packages.html#section9⟩ and ⟨http://people.connexer.com/~roberto/howtos/debcustomize⟩

- install pbuilder and re-create an up to date copy

- ```
  $ sudo pbuilder create
  $ wget http://floppsie.comp.glam.ac.uk/download/scripts/myrevdeb
  $ wget http://floppsie.comp.glam.ac.uk/download/scripts/mypdebbuild
  $ cd halma-0.0.1
  $ bash ../myrevdeb
  $ bash ../mypdebbuild
  ```

- will build a new version of halma which is placed into `/usr/local/src/results`
  ```