Script to automatically build and run touchmap

- here is a script you can run from the command line to automatically rebuild and run your touchmap ⟨http://floppsie.comp.glam.ac.uk/download/targz/run⟩
- you can install it via:
- ```
  $ wget http://floppsie.comp.glam.ac.uk/download/targz/run
  $ chmod 755 run
  ```
- you can run it via:
- ```
  $ ./run
  ```


Script to automatically build and run touchmap

- the contents of `run` is shown on the next slide
- it removes the `touchgui` cache rebuilds `touchmap`
- it also reconfigures touchmap
  - necessary if you make significant changes to `Makefile.am`
- `run` hides all these and will lastly run your version of `touchmap`

Script to automatically build and run touchmap

- **run**

```
#!/bin/bash

VERSION=0.2

cd $HOME
if [ -d .cache ] ; then
    cd .cache
    rm -rf touchgui
    mkdir touchgui
fi
```

Script to automatically build and run touchmap

- **run**

```
cd $HOME/Sandpit/touchmap-${VERSION}
autoreconf

cd $HOME/Sandpit
rm -rf build-touchmap
mkdir build-touchmap
cd build-touchmap

../touchmap-${VERSION}/configure
make
./localrun.sh touchmap.py
```


Adding image assets to touchmap

- **touchmap-0.2/Makefile.am**

```
all: doorh.png doorv.png doorh-bw.png doorv-bw.png hingeh
  wallh.png wallv.png wallh-bw.png wallv-bw.png \
  newname.png

newname.png: $(srcdir)/images/newname.png
°convert -resize 100x100 $< $@
```

- notice that ° needs to be replaced by a single tab character
  - you might need to alter preferences in gedit to allow you to add a tab character

Loading a map into touchmap

- **$HOME/Sandpit/touchmap-0.2/touchmap.py**

```
def load_map (name):
    f = open (name, "r")
    f = read_map (f)
    f.close ()

def myimport (name, tap):
    global clicked
    pygame.display.update ()
    load_map (current_map_name)
    clicked = True
    pygame.display.update ()
```


Loading a map into touchmap

- **$HOME/Sandpit/touchmap-0.2/touchmap.py**

```
def read_floor (lines):
    seen_start = False
    y = 0
    ypos = 0
    for line in lines:
        if len (line) > 0:
            if len (line.split ("#")) > 0:
                seen_start = True
            if seen_start:
                add_xaxis (line, y, ypos)
                y += 1
                ypos += cell_size
```

Loading a map into touchmap

- **$HOME/Sandpit/touchmap-0.2/touchmap.py**

```
def read_map (f):
    lines = f.readlines ()
    read_assets (lines)
    read_floor (lines)
    return f
```


Loading a map into touchmap

- **$HOME/Sandpit/touchmap-0.2/touchmap.py**

```
#
#  add_xaxis - adds a line of buttons.
#              y is the index on the yaxis.  posy is the screen coo
#

def add_xaxis (line, y, ypos):
    global cell_array, button_array
    xpos = 0
    x = 0
    for ch in line:
        b = button (xborder + xpos, yborder + ypos, cell_size)
        if ch == "#":
            cell_array.set_contents (xoffset+x, yoffset+y, "v")
            b.to_wall ()
        elif ch == " ":
            cell_array.set_contents (xoffset+x, yoffset+y, " ")
        button_array.set_contents (xoffset+x, yoffset+y, [b])
        xpos += cell_size
        x += 1
```


Loading a map into touchmap

- **$HOME/Sandpit/touchmap-0.2/touchmap.py**

```
def read_assets (lines):
    for line in lines:
        words = line.lstrip ().split ()
        if (len (words) > 2) and (words[0] == "define"):
            include_asset (words[1], words[2])
```


Thoughts on Checkpointing and forward/next

- assuming that basic saving and loading is complete
  - we notice that it saves to a file `current_map_name`
  - therefore we can take advantage of this and create temporary filenames
  - save periodically
- might be good for `touchmap` to create a directory `$HOME/.cache/touchmap`
  - under which the checkpoint files might be kept
  - notice that many applications keep their file cache contents under `$HOME/.cache` (including `touchgui`)

Thoughts on Checkpointing and forward/next

- it would be possible to have a *forward* and *next* button to cycle through the checkpoint files
  - maybe the application should create a new checkpoint file every 5th action?
- might implement a naming scheme
  - for example: `touchmap-%3d.txt`

Thoughts on Checkpointing and forward/next

- using
- ```
  cp_name = os.path.join (".cache", "touchmap")
  cp_name = os.path.join (cp_name, "touchmap-%3d.txt" % cp_
  cp_name = os.path.join (os.getenv ("HOME"), cp_name)
  ```
- notice the `cp_num` which is the check point number and this should cycle `0-999`
  ```