John Romero Programming Proverbs

- 2. “It’s incredibly important that your game can always be run by your team. Bulletproof your engine by providing defaults (for input data) upon load failure.”

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

Work flow for touchmap, chisel, penguin tower, doom3 usage

- `touchmap` should be able produce a text map which is suitable for penguin tower, isometric penguin tower and doom3
- all maps are in text, doom3 maps and penguin tower maps can be produced using touchmap and chisel

Work flow for touchmap, chisel, penguin tower, doom3 usage

- using a terminal in J109 type:

- ```
  $ j109-d3
  ```

- this will take a few seconds and will:
  - create a set of configuration files for doom3
  - download the chisel source code from github and place it into: `$HOME/Sandpit/chisel`

- it then starts up doom3
  ```

chisel

- after the doom3 window appears open up another command line terminal and type:

```
$ cd
$ cd Sandpit/chisel/python
$ ls ../maps
$ ./developer-txt2map ../maps/three.txt
```

Chisel map: three.txt

$HOME/Sandpit/chisel/maps/three.txt

```
define 1 room 1
define 2 room 2
define 3 room 3
define s worldspawn
define o monster monster_demon_imp
define h monster monster_demon_hellknight
define S monster monster_demon_tick

################################
# 1              # 2           #
#                #             #
# s         o    .             #
#                .             #
#          #######         h   #
#          #                   #
#          #                   #
##################..############
# 3                            #
#    S                         #
#                              #
################################
```

Touchmap

- should eventually generate text files similar to `three.txt`
- `chisel` will produce doom3 and penguin tower equivalents from your text map

Touchmap

- now return to the doom3 window and press the tilde key ~
  - this enables the in game doom3 console
- in this console type: `dmap tiny.map`
  - `dmap` bsp compiles the tiny.map
- once this is complete, type in the doom3 console: `map tiny.map`
  - which loads in the bsp compiled map `tiny.map`
- notice that the output from `chisel` is always `tiny.map`
  - this is for convenience and configuration
  - `chisel` can output the file into any named file if necessary

Penguin Tower

- is a multiplayer 2 dimensional game
  - inspired by Morloc Tower ⟨`http://www.mobygames.com/game/dunjonquest-morlocs-tower`⟩ although Penguin Tower is very different

- however the screen layout and many of the key commands are the same

Penguin Tower

- you can download a copy of the game from here ⟨`http://floppsie.comp.glam.ac.uk/download/penguin-tower/penguin-tower-1.0.tar.gz`⟩.
- you then need to extract the archive using the following command line:
- ```
  $ mkdir $HOME/Sandpit
  $ cd $HOME/Sandpit
  $ wget http://floppsie.comp.glam.ac.uk/download/penguin-tower/penguin-tower-1.0.tar.gz
  $ tar zxvf penguin-tower-1.0.tar.gz
  ```
  ```

Creating important directories

- these directories need to created before penguin tower can be built

- ```
  $ mkdir -p $HOME/Sandpit
  $ cd $HOME/Sandpit
  $ mkdir -p $HOME/opt/bin
  $ mkdir -p build-ptower
  ```
  ```

Building Penguin Tower

- you should be able to build it by typing:

```
$ cd $HOME/Sandpit
$ mkdir -p build-ptower
$ cd build-ptower
$ ../penguin-tower-1.0/configure --prefix=$HOME/opt
$ make
$ cd ..
```

- you need to check that the make program above exited with no error messages

- note the build will recreate all images in the build directory.

Installing your own copy of Penguin Tower

- you can install your own copy of Penguin Tower into your directory (`$HOME/opt`) specified by the `--prefix` to the `./configure` command

- ```
  $ cd $HOME/Sandpit
  $ mkdir -p build-ptower
  $ cd build-ptower
  $ make install
  $ cd ..
  ```
  ```

Penguin Tower keyboard controls

- keyboard controls ⟨`ptower.html`⟩.

Running the client of Penguin Tower

- you should be able to run the client like this:

```
$ $HOME/opt/bin/penguin-tower mcgreg.comp.glam.ac.uk:7000
```

- also consider running it in fullscreen by:

```
$ $HOME/opt/bin/penguin-tower -f mcgreg.comp.glam.ac.uk:7000
```

Configure notes

- you only need to execute `../penguin-tower-1.0/configure --prefix=$HOME/opt` once ever in this directory (unless you modify the package)

- check out the documentation here ⟨`ptower.html`⟩.

Configure notes

- you can also run the system installed version by typing:

- ```
  $ penguin-tower mcgreg.comp.glam.ac.uk:7000
  ```
  ```

Running your own Penguin Tower server

- can be done by opening a terminal and typing the following:

```
$ cd
$ ./opt/bin/ptower-server ./opt/share/ptower/maps/star
```

- to stop the server, type `^C` (press down the control key and then press the `c` key, now release both keys)

- there are a number of maps in the `$HOME/opt/share/penguin-tower/maps` directory
  - `m1`, `star` and `glover`
  - `star` is the smallest (5 rooms)

Tutorial work

- examine the chisel file input file
  `$HOME/Sandpit/chisel/maps/three.txt`
- examine the chisel file output
  `$HOME/Sandpit/chisel/python/tiny.pen`
- try making a simple change to `$HOME/Sandpit/chisel/maps/three.txt` and bsp compile the map and load it into doom3
- examine many of the files in `$HOME/Sandpit/chisel/maps` what mapping features does the chisel program seem to provide
  - write a list of `chisel` features you would like to utilise in your `touchmap` tool