Doom3

- we will be using the dhewm3 fork of the doom3 code with heavy modifications which include:
  - gore, speed ups
  - python bots!

- about 681075 total lines of code
  - however this includes many tools and libraries

- actual engine is closer to 137905 lines of code

Game engines

- there are over 50 open source game engines
  - of differing size and quality

- require a balance between: complexity and resource requirements

- resource requirements are not always obvious

- for example the time to recompile Panda3D was 50 minutes on a 4 GHz AMD black
  - dhewm compiles in 3 minutes
  - pge compiles in 1 minute

Doom3 advantages

- code base is very clean implemented in C++

- it does not use the STL
  - most libraries are implemented from scratch
  - highly portable codebase and fast

- Doom3 lineage is strong!
  - many eyeballs keep bugs shallow

- Doom3 codebase can be extended and makes an effective teaching tool

- all maps and models are stored in text format! (Excluding images/sounds)

- uses the MAP and BSP format which is the Rosetta stone to game engine design

Morloc Tower

- Morloc Tower ⟨http://www.mobygames.com/game/dunjonquest-morlocs-tower⟩ was a game written in 1980

- the tower had six stories and consists of 30 rooms total

- the wizard Morloc was the boss enemy which you had to defeat to complete the game
  - the quicker you killed him the higher your score

- it had adventure elements to the game (pick up magic sword or hand grenade)
  - which would take time to find

Morloc Tower

- there were smaller monsters to kill before you reached Morloc
  - single player real-time adventure game

Penguin tower

- Penguin tower is a multiplayer 2 dimensional game which was inspired by Morloc Tower

- Penguin Tower is a very different game it does retain a similar screen layout and many of the key commands are the same

- the goal of the game is to stay alive as long as possible and to inflict the most damage on other players
  - genre is a graphical multiuser dungeon with a limited graphical interface and limited number of objects and weapons

Penguin tower

- architecturally it consists of three main components, a client, a server and a protocol

- the client is written in Python and it utilises the pygame libraries

- the server is mostly written in Modula-2 and a small amount of C
  - the protocol is entirely character and string based

Penguin tower

- the penguin tower server code was written during two Augusts in 1985 and 1986 and originally ran on a 6 Mhz PC clone connected by two Visual 200 terminals (making it a three player game).

- the maps were drawn with simple ASCII characters, and it was quite playable (for those days!)

- the 6 Mhz PC ran the server code quite comfortably, it occasionally slowed down, which perhaps added to its charm.
  - normally when someone pulled the hand grenade

Penguin tower demo

- demo screenshots ⟨`http://floppsie.comp.glam.ac.uk/Southwales/gaius/games/ptower.html`⟩

Doom3 mod

- how difficult would it be to produce a Penguin Tower mod for Doom3?
  - utilise the monsters from Doom3
  - generate very simple maps in the style of (Morloc Tower and Penguin Tower)

- very simple maps have the advantage of only having 90° corners
  - ideal teaching vehicle and also it should be possible to generate maps quickly

Penguin tower maps

- there are a number of penguin tower maps and a tool to randomly create large maps

- the Penguin tower file format is simple and easily extensible

Penguin tower map: star.pen

-
```
ROOM 1

WALL 1 1 1 20
WALL 1 20 20 20
WALL 20 20 20 1
WALL 20 1 1 1

DOOR 9 20 12 20   STATUS CLOSED LEADS TO 5
DOOR 20 9 20 12   STATUS CLOSED LEADS TO 5
END
```

- etc

Chisel

- Chisel ⟨`https://github.com/gaiusm/chisel`⟩

- is a github project containing command line tools to create doom3 maps

- the tools allow anyone with basic computer skills to generate small doom3 maps

Map: one.txt

-
```
define 1 room 1
define s worldspawn
define o monster monster_demon_imp
define n monster monster_demon_hellknight
define i light
define a ammo ammo_shells_large 16

##########################################################
# 1    i       i                    i
#
# s        a                                             n
#
#                          o
#                i                              i
##########################################################
```

Map: two.txt

- Chisel ⟨`https://github.com/gaiusm/chisel`⟩

```
define 1 room 1
define 2 room 2
define s worldspawn
define o monster monster_demon_imp
define n monster monster_demon_hellknight
define i light
define a ammo ammo_grenade 16

###############################
# 1         i   # 2      i    #
#         o     #             #
# s             .            #
#               .             #
#               .             #
#               #     g   n   #
#      i        #   i         #
###############################
```

Compiling a map

-
```
$ txt2pen -o one.pen one.txt
$ pen2map -o one.map one.pen
```

- notice the txt file is compiled into a pen file
  - the pen file is compiled into map file

Obtaining chisel

-
```
$ cd
$ mkdir -p Sandpit
$ cd Sandpit
$ git clone https://github.com/gaiusm/chisel
```