Chisel floor levels and doom3 botlib experimentation, documentation and extension

- for your coursework you need to alter chisel and also extend the doom3 botlib API
  - during this tutorial we will attempt to modify the floor level in chisel
  - make a start on changing the botlib API
    - this section will start with some experimentation, to gain confidence about the botlib API

Step 1: new emacs configuration file

- download and install a new emacs configuration file

- ```
  $ cd
  $ rm -f skeleton-doom3-data.tar.gz
  $ wget http://floppsie.comp.glam.ac.uk/download/targz/skeleton-doom3-data.tar.gz
  $ tar zxvf skeleton-doom3-data.tar.gz
  ```
  ```

Step 1: new emacs configuration file

- this new emacs configuration has the cheat sheet built in via, F10
  - other useful features (shown in the cheat sheet)

- open up a terminal and update your dhewm3 code

- ```
  $ cd $HOME/Sandpit/git-doom3/pybot-dhewm3
  $ git pull
  $ cd $HOME/Sandpit/git-doom3
  $ rm -rf build
  ```
  ```

Step 2: chisel floor levels

- currently `pen2map` converts a penguin tower file into a doom3 map file
  - however map floor is completely level
  - it would be good aesthetically to introduce minor floor level changes between rooms
- start up `emacs` and press F7 and then press F10
- in `pen2map.py` search forward for `floorLevel`
- notice how it is initialised to zero in the `roomInfo` constructor

Step 2: chisel floor levels

- modify this so that it is 0 for an even room number and -0.25 for an odd room number
- test these changes on a two room map
  - test these changes on a three room map
  - test these changes on a four room map
- a unit of 1 in a penguin tower map represents 48 inches in the doom3 world
- homework: see if you can think of a better algorithm in which to change floor levels

Step 3: understanding more about the Python Bot API

- quit `emacs`
- make sure that dhewm3 runs in a window (not full screen)
- now set the environment variable `DEBUG_PYBOT` from your command line, then restart emacs
  - this will allow you to debug the python bot and the doom3 game engine
-

```
$ export DEBUG_PYBOT=yes
$ emacs &
```

Step 3: understanding more about the Python Bot API

- use emacs to load the file `$HOME/Sandpit/git-doom3/pybot-dhewm3/python-bot/python_doommarine.py`
  - remember `$HOME` is shorthand for `/home/yourusername`
  - recall that you can use the <tab> key to complete filename and directory names in emacs

Step 3: understanding more about the Python Bot API

- now press F12 and when this has completed F5
  - this will compile dhewm3 (F12) and then debug dhewm3 (F5)
  - press F10 for help

- make sure that dhewm3 has been configured to run in a window (not full screen)
  - if not reconfigure it and quit dhewm3 and then press F5 in emacs

Step 3: understanding more about the Python Bot API

- open up another terminal

- ```
  $ cd $HOME/Sandpit/chisel/python
  $ ./developer-txt2map ../maps/onebot.txt
  ```
  ```

Step 3: understanding more about the Python Bot API

- open up another terminal
-
```
$ cd $HOME/Sandpit/git-doom3/pybot-dhewm3/python-bot
$ python3 python_doommarine.py 0
```
- this will run the python bot from the command line and allows you to see any debugging output
- return to the dhewm program and pull down the in game console (using ~)
- now type:
  - `dmap tiny.map`
  - `map tiny.map`

Step 3: understanding more about the Python Bot API

- you should see Python bot appear and run in a circle
  - the game engine is being run under the debugger
  - python bot is being run from the command line

Step 3: understanding more about the Python Bot API

- see if you can change `python_doommarine_1.py` to make Python bot walk around in a circle rather than run
- create two functions `walkCircle` and `runCircle`
- finally change the program to make Python bot turn without walking
  - see if you can change botlib.py so that an turn angle of 0 degrees is straight up in the penguin map
- you will need to read and study the file `$HOME/Sandpit/git-doom3/pybot-dhewm3/python-bot/botlib.py`

Step 3: understanding more about the Python Bot API

- homework, write out a list of functions implemented in `botlib.py` together with their functionality
  - complete the walkCircle/runCircle and turn exercises from above

- consider what extra basic movements are desirable in `botlib.py`