

Coursework support

- it is worth making sure you have an up to date version of chisel from github
- you will need a selection of test maps of varying size which will depend upon the performance of your desktop
- there are a number of maps inside chisel, feel free to adapt them or create new ones
 - a good starting place is to use the random map tool we looked at last week
 - alternatively check out:
chisel/maps/complex.txt,
chisel/maps/complex2.txt,
chisel/maps/complex3.txt and
chisel/maps/two.txt

Running chisel

```

$ cd
$ cd Sandpit/chisel/python
$ ./developer-txt2map ../maps/complex3.txt
txt2pen: pass
Total rooms = 28
Total cuboids = 4760
Total cuboids expanded (optimised) = 0
Total entities used = 417 entities unused = 3679
Total brushes used = 4760
pen2map: pass

```

- notice the total cuboids optimised

Running chisel

- once you have completed the chisel/python/student/chcuboid.py file you can rerun chisel

```

$ ./developer-txt2map ../maps/complex3.txt
Total rooms = 28
Total cuboids = 4647
Total cuboids expanded (optimised) = 162
Total entities used = 375 entities unused = 3721
Total brushes used = 4647
pen2map: pass

```

- notice it has combined 162 cuboids

Coursework support (chisel)

- you need to complete chisel/python/student/chcuboid.py to implement the optimisation of the cuboids
 - search for `--complete me--` markers in the source code
- here we will complete `_canExtend` as an example

canExtend

chisel/python/student/chcuboid.py

```
#
# canExtend - returns True if we can enlarge self to
#             contain pos, size. We can only do
#             this if pos, size joins or overlaps with s
#             We have already tested for a superset or
#             subset elsewhere so this routine just
#             handles extending a cuboid.
#
```

canExtend

chisel/python/student/chcuboid.py

```
def _canExtend (self, pos, size):
    if chcuboid_enable_optimise:
        #
        # if the self x and y limits are the same as (po
        # then
        # see if we can extend the z axis to combine
        # do the same for the z, x and y axis
        # do the same for the z, y and x axis
        #
        if self._xlimits (pos, size) and self._ylimits (p
            return self._Zextend (pos, size)
        if self._zlimits (pos, size) and self._xlimits (p
            return self._Yextend (pos, size)
        if self._zlimits (pos, size) and self._ylimits (p
            return self._Xextend (pos, size)
    return False
```

xlimits

chisel/python/student/chcuboid.py

```
#
# _xlimits - return True if self has the same x start, e
#           defined by pos and size.
#
def _xlimits (self, pos, size):
    return self._limits (pos, size, 0)
```

- you need to complete the remainder of the
--complete me-- functions