

## Assessment Cover Sheet 2019-20

Module Code:	Module Title:	Module Team:
CS4S765	Game Engine Optimisation	<a href="#">Gaius Mulley</a>
Assessment Title:		Assessment No.:
Implement game engine optimisation: extending chisel.		2
Date Set:	Submission Date:	Return Date:
23-Sep-2019 23:55	03-Apr-2020 23:55	29-Apr-2020 23:55

**IT IS YOUR RESPONSIBILITY TO KEEP RECORDS OF ALL WORK SUBMITTED.**

### Marking and Assessment

This assignment will be marked out of **100%**.

This assignment contributes to **50%** of the total module marks.

### Learning Outcomes to be assessed

As specified in the validated module descriptor <https://icis.southwales.ac.uk>

- 1) Demonstrate the ability to analyse and critically evaluate techniques used to optimise game engines
- 2) Demonstrate the ability to analyse, create and evaluate game engine code

*Awarded mark is only provisional: subject to change and / or confirmation by the Assessment Board.*

# Assessment Task

The aim of this coursework is to extend the chisel free software package which allows doom3 maps to be built from the command line.

## Part 1. Chisel changes

Your extensions to this package should primarily be directed towards the tool `pen2map.py`. This will transform a simple pen map into a doom3 map. It can be extended in any way you feel appropriate. You might choose to work on any of the following (but (i) is mandatory)

- (i) implementing open doors between rooms (using visportals). Make this change switchable from the command line.
- (ii) You might also consider how more effective beams might be created.

The chisel software can be obtained using git:

```
git clone https://github.com/gaiusm/chisel
```

## Part 2: Chisel optimisations

Introduce a `brushdef` class to `pen2map.py` and record the extent of each brush in as simple brick with `x,y,z` dimensions. Now implement an optimisation procedure to

combine adjacent bricks and emit the resulting set of brushes to a map file. Generate some statistics on how many brushes are optimized and also comment on the frame rate of doom3. Also generate statistics from turning on the generation of visportals for generated chisel map and compare the same map without visportals. How does the frame rate differ using these techniques?

Your report should include a git diff and also a commentary of the changes you made and justification. You should include relevant screenshots within your report. Finally your report should comment on the success or otherwise of using this tool for the creation of maps for doom3 and future improvements you might make. Your report should not exceed 3000 words (excluding code)

# Marking Scheme

	<b>Fail (0/29)</b>	<b>Narrow Fail (30/39)</b>	<b>3rd Class / Pass (40/49)</b>	<b>Lower 2nd Class / Pass (50/59)</b>	<b>Upper 2nd Class / Merit (60/69)</b>	<b>1st Class / Distinction (70/100)</b>
Chisel functionality changes (part 1) (40%)	<input type="checkbox"/> Very poor Chisel functionality changes (part 1)	<input type="checkbox"/> Poor Chisel functionality changes (part 1)	<input type="checkbox"/> Satisfactory Chisel functionality changes (part 1)	<input type="checkbox"/> Good Chisel functionality changes (part 1). visportals implemented	<input type="checkbox"/> Very good Chisel functionality changes (part 1). visportals implemented and other features	<input type="checkbox"/> Excellent Chisel functionality changes (part 1). visportals implemented and features which have been independently created
Chisel optimisation changes (part 2) (30%)	<input type="checkbox"/> Very poor Chisel optimisation changes (part 2)	<input type="checkbox"/> Poor Chisel optimisation changes (part 2)	<input type="checkbox"/> Satisfactory Chisel optimisation changes (part 2). A good start and the direction of travel is correct	<input type="checkbox"/> Good Chisel optimisation changes (part 2). Functionality complete	<input type="checkbox"/> Very good Chisel optimisation changes (part 2). Relevant optimisation statistics generated	<input type="checkbox"/> Excellent Chisel optimisation changes (part 2). Well crafted code produced which also generates useful optimisation statistics
commentary on the usefulness and future improvements (30%)	<input type="checkbox"/> Very poor commentary on the usefulness and future improvements	<input type="checkbox"/> Poor commentary on the usefulness and future improvements	<input type="checkbox"/> Satisfactory commentary on the usefulness and future improvements. The commentary addresses some of the areas with errors and omissions	<input type="checkbox"/> Good commentary on the usefulness and future improvements. The commentary addresses the majority of areas with a few errors or omissions	<input type="checkbox"/> Very good commentary on the usefulness and future improvements. The commentary addresses the majority of areas with no major errors or omissions	<input type="checkbox"/> Excellent commentary on the usefulness and future improvements. The commentary contains a high amount of independent thought and also all the major areas are covered without errors
Global:						