



Assessment Cover Sheet and Feedback Form 2018-19

Module Code: CS2S566	Module Title: Tool Development for Computer Games	Module Team: Gaius Mulley
Assessment Title and Tasks: Produce and evaluate a content creation tool for a simple game engine		Assessment No. 2
Date Set 7/1/2019	Submission Date 29/3/2019	Feedback Date 25/4/2019

IT IS YOUR RESPONSIBILITY TO KEEP RECORDS OF ALL WORK SUBMITTED

<p style="text-align: center;">Marking and Assessment</p> <p>This assignment will be marked out of 100%</p> <p>This assignment contributes to 50% of the total module marks.</p> <p>Learning Outcomes to be assessed (as specified in the validated module descriptor https://icis.southwales.ac.uk/):</p> <ol style="list-style-type: none">1) To identify the functional and non-functional requirements of a game engine / game design2) Apply relevant software engineering techniques to develop applications to generate data for use in a game engine <p style="text-align: center;"><i>Provisional mark only: subject to change and / or confirmation by the Assessment Board</i></p>

Your task is to implement a Python based GUI tool which will produce maps suitable for chisel and isometric penguin tower.

Your tool should initially allow users to click on tiles which can be chosen to be either a wall, door or space.

It can be extended to include other attributes such as pickups, monsters and lights if desired. Your report must also include a user guide and line by line commentary. The semantic checking of a limit of walls per room and door sizes required for isometric penguin tower would be beneficial.

Marking Scheme:

	Fail	Narrow Fail	3rd Class / Pass	Lower 2nd Class / Pass	Upper 2nd Class / Merit	1st Class / Distinction
semantic checking of data entry 20%	<ul style="list-style-type: none"> • Very poor semantic checking of data entry 	<ul style="list-style-type: none"> • Poor semantic checking of data entry 	<ul style="list-style-type: none"> • Satisfactory semantic checking of data entry 	<ul style="list-style-type: none"> • Good semantic checking of data entry 	<ul style="list-style-type: none"> • Very good semantic checking of data entry 	<ul style="list-style-type: none"> • Excellent semantic checking of data entry
line by line commentary 20%	<ul style="list-style-type: none"> • Very poor line by line commentary 	<ul style="list-style-type: none"> • Poor line by line commentary 	<ul style="list-style-type: none"> • Satisfactory line by line commentary 	<ul style="list-style-type: none"> • Good line by line commentary 	<ul style="list-style-type: none"> • Very good line by line commentary 	<ul style="list-style-type: none"> • Excellent line by line commentary
user guide 20%	<ul style="list-style-type: none"> • Very poor user guide 	<ul style="list-style-type: none"> • Poor user guide 	<ul style="list-style-type: none"> • Satisfactory user guide 	<ul style="list-style-type: none"> • Good user guide 	<ul style="list-style-type: none"> • Very good user guide 	<ul style="list-style-type: none"> • Excellent user guide
human computer interface 20%	<ul style="list-style-type: none"> • Very poor human computer interface 	<ul style="list-style-type: none"> • Poor human computer interface 	<ul style="list-style-type: none"> • Satisfactory human computer interface 	<ul style="list-style-type: none"> • Good human computer interface 	<ul style="list-style-type: none"> • Very good human computer interface 	<ul style="list-style-type: none"> • Excellent human computer interface
code quality and use of Python 20%	<ul style="list-style-type: none"> • Very poor code quality and use of Python 	<ul style="list-style-type: none"> • Poor code quality and use of Python 	<ul style="list-style-type: none"> • Satisfactory code quality and use of Python 	<ul style="list-style-type: none"> • Good code quality and use of Python 	<ul style="list-style-type: none"> • Very good code quality and use of Python 	<ul style="list-style-type: none"> • Excellent code quality and use of Python