Faculty of Computing, Engineering and Science

Assessment Cover Sheet 2019-20

Module Code:	Module Title:	Module Team:
CS2S566	Tool Development for Computer Games	Gaius Mulley
Assessment Title:		**Assessment No.:**
Python/Pygame and Missile command		1
Date Set:	**Submission Date:**	**Return Date:**
23-Sep-2019 00:00	13-Dec-2019 23:55	10-Jan-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) To identify the functional and non-functional requirements of a game engine / game design   2) Apply relevant software engineering techniques to develop applications to generate data for use in a game engine
Awarded mark is only provisional: subject to change and / or confirmation by the Assessment Board.

Assessment Task

Your task is to write an implementation of missile command suitable for running on a desktop computer using Python/Pygame and mouse as its primary input.

You should ignore gui based menus as these are covered next term. Your task is to concentrate on making the game work using Pygame and Python. You should document the controls chosen and justify the design decisions.

You must also provide a line by line commentary of all code you write. Finally you should give an analysis of the effectiveness of Python/Pygame when implementing this game.

Marking Scheme

	Fail (0/29)	Narrow Fail (30/39)	3rd Class / Pass (40/49)	Lower 2nd Class / Pass (50/59)	Upper 2nd Class / Merit (60/69)	1st Class / Distinction (70/100)
discussion on the effectiveness of Python/Pygame in producing such a game (30%)	☐ Very poor discussion on the effectiveness of Python/Pygame in producing such a game	☐ Poor discussion on the effectiveness of Python/Pygame in producing such a game	☐ Satisfactory discussion on the effectiveness of Python/Pygame in producing such a game	☐ Good discussion on the effectiveness of Python/Pygame in producing such a game	☐ Very good discussion on the effectiveness of Python/Pygame in producing such a game	☐ Excellent discussion on the effectiveness of Python/Pygame in producing such a game
line by line commentary (20%)	☐ Very poor line by line commentary	☐ Poor line by line commentary	☐ Satisfactory line by line commentary	☐ Good line by line commentary	☐ Very good line by line commentary	☐ Excellent line by line commentary
use of PyGame libraries (20%)	☐ Very poor use of PyGame libraries	☐ Poor use of PyGame libraries	☐ Satisfactory use of PyGame libraries	☐ Good use of PyGame libraries	☐ Very good use of PyGame libraries	☐ Excellent use of PyGame libraries
various inputs handled (20%)	☐ Very poor various inputs handled	☐ Poor various inputs handled	☐ Satisfactory various inputs handled	☐ Good various inputs handled	☐ Very good various inputs handled	☐ Excellent various inputs handled
controls chosen (10%)	☐ Very poor controls chosen	☐ Poor controls chosen	☐ Satisfactory controls chosen	☐ Good controls chosen	☐ Very good controls chosen	☐ Excellent controls chosen
Global:						