University of South Wales
Prifysgol De Cymru

Faculty of Computing, Engineering and Science

Assessment Cover Sheet and Feedback Form 2018-19

Module Code:	Module Title:	Module Team:
CS3S665	Game Engine Design	Gaius Mulley
Assessment Title and Tasks: Enhancing pge and implementing frozen bubble		Assessment No. 2
Date Set **24/9/2018**	**Submission Date** **05/04/2019**	**Feedback Date** **03/05/2019**

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 critically evaluate the techniques that underpin modern game engines 2) To be able to justify techniques used in the design, development and evaluation of game engine and gameplay code
Provisional mark only: subject to change and / or confirmation by the Assessment Board

The aim of this coursework is to extend the PGE and Python interface API code. You should start by incorporating per object inelastic collisions. You can extend PGE in any way you wish and then you should implement a simple 2D game which uses your changes. The frozen bubble game is an idea but you can choose your own game if you prefer.

You can obtain pge from github via:

git clone https://github.com/gaiusm/pge

Once you have implemented these changes you should consider making any other improvements based on your own research.

For each improvement you make you should generate simple Python test cases to demonstrate your code is working.

Your report must consist of a program listing, a line by commentary of any changes/improvement that you make and appropriate screen shots.

Marking Scheme:

	Fail	Narrow Fail	3rd Class / Pass	Lower 2nd Class / Pass	Upper 2nd Class / Merit	1st Class / Distinction
Inelastic collisions and per object gravity 30%	• Very poor Inelastic collisions and per object gravity	• Poor Inelastic collisions and per object gravity	• Satisfactory Inelastic collisions and per object gravity. A single feature was changed. Some obvious code weaknesses exist, but the overall direction was sensible	• Good Inelastic collisions and per object gravity. Sensible changes attempted, code contains some errors but is along the correct path	• Very good Inelastic collisions and per object gravity. Interesting and effective changes made	Excellent Inelastic collisions and per object gravity. Code contains independent ideas and is well crafted
Any other improvements 30%	• Very poor Any other improvements	• Poor Any other improvements	• Satisfactory Any other improvements	• Good Any other improvements	• Very good Any other improvements	• Excellent Any other improvements. The improvements contain a high amount of independent thought
Test programs 20%	• Very poor Test programs	• Poor Test programs	• Satisfactory Test programs	• Good Test programs	• Very good Test programs	• Excellent Test programs. Very thorough tests produced
Documentation/Commentary 20%	• Very poor Documentation/Commentary	• Poor Documentation/ Commentary	• Satisfactory Documentation/Commentary. The commentary addresses some of the areas with errors and omissions	• Good Documentation/Commentary. The commentary addresses the majority of areas with a few errors or omissions	• Very good Documentation/Commentary. The commentary addresses the majority of areas with no major errors or omissions	• Excellent Documentation/ Commentary. The commentary contains a high amount of independent thought and also all the major areas are covered without errors