Module code	CI123
Module title	Introduction to Games Development
Coursework hand out date	13th February 2014
Coursework hand in date	4th May 2014

Coursework feedback will be provided within three term-time weeks unless otherwise notified.

General description of the coursework:
Your assignment is to implement a game level for a PacMan style game. It is based on the sample code in `StarshipFontana` which is available on GitHub.
The current code creates a simple game level where

- static enemies are placed on screen,
- the protagonist can move left and right in order to shoot aliens.

You should modify this code such that there are static barriers, such as walls, and collectable items, such as coins.
You have a wide remit to provide a new level. Such a level could have more difficult obstacles. Furthermore, it could have coins that move, rather than are static. A really interesting level would have some kind of basic enemy. An excellent project must extend the automated testing to other components. Finally, you should do something to notify the user that they have won the game.

Module outcomes assessed by this piece of work: Learning outcomes 1, 2 and 3.

Names of markers: Aidan Delaney

Indicative marking criteria:

Percent of total	**criteria**
25%	Clean code structure - subclassing where required, no hacks.
25%	Working game level - whether I can actually play your game level.
25%	Use of more complex structures, such as `std::list` and using standard tools such as `git`.
25%	Implementation of game idea - you may wish to provide original sketches of your idea as evidence for this.

Note that the above criteria are indicative. This means that if you do something interesting, or have an interesting game concept then I can give you marks for that.

Submission procedure:
Your code will be tested on my Intel based Fedora machine (Fedora 20 on an Intel i5). Your code should be in a git repository that I can pull from. A plain-text document called `README.md` (i.e. markdown formatted) should be in the root of your archive and detail exactly how to build your code. If you have used any 3^{rd} party code, you should document this in the README file and ensure that you're using the code under a licence that allows it. An astute student may automate the build using `make` or GNU autotools. If your documentation for building your code is not correct and concise, then it will be harder for me to build your product. If I can't build your product, it's even harder for me to award marks.
To submit your project you should do two things:

- Ensure the latest version of your source code has been pushed to some repository that I can pull from, and
- submit the repository URL and any supporting documentation as a PDF (I only accept PDF for technical reasons) on StudentCentral.

1. A copy of your coursework submission may be made as part of the University of Brighton and School of Computing, Mathematical and Information Sciences procedures which aim to monitor and improve quality of teaching. If a copy is made, it will be kept only for this purpose and will be destroyed once this purpose has been fulfilled. You should refer to your student handbook for details.
2. All work submitted must be your own (or your teams for an assignment which has been specified as a group submission) and all sources which do not fall into that category must be correctly attributed. The markers may submit the whole set of submissions to the JISC Plagiarism Detection Service.