Joint Project 1 with Arrays

(Programming Project Only)

**This project involves the following:**

* You are to put **two** arrays into your Joint Project 1. Most projects involved firing a bullet, you should now implement an array of bullets instead of an individual bullet object. You could also implement an array of one of your types of enemies, so instead of having 2-3 individual enemy objects you could now have an array of enemies. You are NOT allowed to use a **foreach** loop statement when manipulating the arrays. You can use **for** or **while** loops, though it is more common to use a **for** loop.
* You are also to make your instance variables private to the class unless they are constants or public data shared between classes. You will need to add properties to your project to make it possible to set and get the instance variable values outside of the class.
* If you have time make some of the changes recommended to you by Peter or I when marking your project. These changes should improve your project. For example look at your project to see if you can make any of your methods more flexible and reuseable. This is good programming practice. For example can your collision detection method(s) be more reuseable when handling arrays. Look at the example on page 28 of my notes “Chapter 4 Methods – Console” called **DetectRectCollision( )** for a good example.

# It is absolutely prohibited to use anyone else’s code in your project. You can ask for help with a particular problem from friends, colleagues, the lecturers in the lab etc but you must code the fix to the problem yourself.

## Submission

Project code due: **Friday the 19thof February (put on your M drive).**

Submit to the M drive. Make sure your project is stored in a folder called **JointProject1withArrays**UserID i.e. **JointProject1withArrays**C00115099 on your M drive. Also sign the Plagiarism Declaration (in the project folder of your common drive) and copy it to your project folder.

**Very Important:**You're required to demonstrate the project in class and to submit it to the M drive. **If you do not demonstrate the project in class you will NOT receive any marks for the project. Late entries will not be considered.** A part of your demonstration you will be asked a **few technical questions** on how you designed and coded parts of your project. You should be able to answer these questions in a clear and concise manner. Failure to answer our questions adequately will result in further questions being asked and you may have to be interviewed about your project.

**This project (code) is worth 5-10% of your final mark in Programming.**

## Coding

Pay attention to the programming standards (given out in class and on your common drive). To do well here, you will need:

* proper structure and layout of classes & methods in your project, including correct indentation of code.
* well commented code - used to explain pieces of code throughout methods where necessary.
* well named, appropriate classes & variables (remember to keep your variables as local as possible.)
* appropriate use of conditional expressions and loops
* the code should be divided into well named methods, each of which do one thing; you should be able to pass parameters to the methods and return values from them.

## Improving Your Project

If you have time make some of the changes recommended to you by Peter or I when marking your project. These changes should improve your project. If no recommendations were made you should talk to Peter or I to ask what we recommend. For example look at your project to see if you can make any of your methods more flexible and reuseable. If no recommendations are given, look to see what extra functionality you can implement.

**Examples of extra functionality:**

* Introducing new enemy behaviours (enemy types) to the game. Different enemies must have different behaviours to count as extra functionality.
* Using different types of weapons (bomb, guided missile, beam etc.).
* The enemies can fire a weapon also.
* Program eye(s) that tracks an enemy.

**Marking Scheme**

|  |  |  |
| --- | --- | --- |
|  | **Games** | **Core** |
| **Add two arrays** | 35% | 55% |
| **Private instance variables** | 15% | 25% |
| **Code Quality** | 20% | 20% |
| **Improving your Project** | 30% |  |
| **Total** | **100%** | **100%** |