GD1P03 **2D Game Programming**	**Summative Assignment** **OO Sprite-Based Game Code Project**	**Hand-Out Date:** **24th May 2018**
		Submission Date: **5:00pm, 21st June 2018**

Submission filename:
`YYYY-MM-DD – GD1P03 – OO Sprite Game Project – Student Name(s).zip`

Object Oriented 2D GDI Sprite Game:
Create a simple 2D "space-invader" game utilizing object oriented programming techniques with Windows GDI sprites. The game must run in a window. The application must report when the user loses the game.

General rules about Space-Invaders

"*Space Invaders* is a 2D fixed shooter game in which the player controls a laser cannon by moving it horizontally across the bottom of the screen and firing at descending aliens. The aim is to defeat five rows of eleven aliens that move horizontally back and forth across the screen as they advance towards the bottom of the screen. The player defeats an alien, and earns points, by shooting it with the laser cannon. As more aliens are defeated, the aliens' movement and the game's music both speed up. Defeating the aliens brings another wave that is more difficult, a loop which can continue indefinitely.

The aliens attempt to destroy the cannon by firing at it while they approach the bottom of the screen. If they reach the bottom, the alien invasion is successful and the game ends. A special "mystery ship" will occasionally move across the top of the screen and bonus points will be awarded if it is destroyed. The laser cannon is partially protected by several stationary defence bunkers—the number varies by version—that are gradually destroyed by projectiles from the aliens and player."[1]

[1] "Space Invaders." Wikipedia: The Free Encyclopaedia. Wikimedia Foundation, Inc. 11th October 2015. Web. <https://en.wikipedia.org/wiki/Space_Invaders>

Team-based or Individual:
This assignment can be done individually or in groups of two (at most). If you are working with a partner, you must include the names of both the team members in the submission file.

Build Quality:
The source code is required to display the following features:

- Compiling code:
  - **Code must build as submitted** in both Debug and Release modes.
  - No warnings or errors present at Warning Level three for all build targets.
- A folder containing an electronic source code must be included with the submission.
  - Visual Studio 2017 solution file, project file, and source files are required.
  - Required external game data, such as INI files (if used).
  - All other files must be removed.

GD1P03 **2D Game Programming**	**Summative Assignment** **OO Sprite-Based Game Code Project**	**Hand-Out Date:** **24th May 2018**
		Submission Date: **5:00pm, 21st June 2018**

Coding Standards:
The source code is required to adhere to the Media Design School's Game Development Faculty's Coding Standard.

Runtime Quality:
The application must not have the following issues:

- Memory leaks.
- Bugs.
- Crashes.

Technical Features:
The game is required to demonstrate the following features:

- Appropriate, effective and correct usage of:
    - C++.
    - Object Oriented Programming.
    - Windows GDI.
    - Sprites.
    - Debug Window.

Debug Window:
Create a dialog window, which can be opened at run time when the user presses the Escape key. The purpose of this window is to allow a tester to modify in-game values, such as the speed of the bullets, the speed of the aliens etc. This will allow the tester to change the state of various important variables in the game, allowing gameplay to be tweaked without having to rebuild the code.

Interface Features:
The executable is required to provide an intuitive interface with the following features:

- Clear instructions are provided both in game and an external readme file.
- Controls are clearly identifiable and intuitive while playing.
- In game interface design and layout makes effective use of screen space.
- At the end of the game, a new game can be started without exiting the executable.

Release Build Zip:
A release build executable must be zipped and included with the submission. This is equivalent to the final build of the game which is about to be mastered for release. Ensure that project settings are set to Release when creating this build.
Name the Release build zip as **`Space Invaders – Student Name.zip`**.

GD1P03 2D Game Programming	**Summative Assignment OO Sprite-Based Game Code Project**	**Hand-Out Date: 24th May 2018**
		Submission Date: 5:00pm, 21st June 2018

Submission Checklist:

Source code folder:

- Solution file (.sln).
- Project file (.vcxproj).
- Source files (.cpp, .h).
- Library files, if any (.lib).
- Resource script files (*.rc)
- Intermediate files have been removed.

Release build zip:

- Stand alone executable (.exe) file.
- Readme file (.txt).

The work is to be placed in a zip file and submitted via the “Web Drop Box.” The file structure and file names of the submission must follow the file hierarchy listed below. Replace the underlined portions with the appropriate values; italic text identifies the required folders.

🗐 **`YYYY-MM-DD – GD1P03 – OO Sprite Game Project – Student Name(s).zip`**

    🗐 **`Space Invaders – Student Name(s).zip`**

    🗁 ***`Source – Student Name(s)`***

        🗋 **`Space Invaders.sln`**

        🗋 …Project and source code, etc.

ASSESSMENT CRITERIA:

Grade D:

- No work submitted **OR**
- Work submitted but the source code does not build **OR**
- The executable works but it does not implement any of the features of the Space Invaders game.

Grade C:

The development of an application that has the following features:

- Ability to draw the five rows of aliens and the player ship on the screen.
- The player ship can be moved by the player laterally across the screen.
- The aliens move across the screen, dropping down one space and reversing direction as they reach the end of the screen.
- The player ship is able to fire bullets at the descending aliens.
- The aliens are able to shoot back at the player.
- Coding standards are adhered to in a few places in the code.

Grade B, as per grade C and:

The application has the following additional features:

- There are destructible bunkers near the bottom of the screen.
- A mystery ship occasionally appears near the top of the screen and shooting the ship awards bonus points to the player.
- The current score and player's lives are clearly displayed in-game. The player should have three lives in each round of play.
- The application is able to report when the player loses the game.
- The Debug Window as specified in the brief has been implemented.
- Coding standards have been adhered to in most places in the code.

Grade A, as per grade B and:

The application has the following additional features:

- The game has multiple rounds of play with a new and faster wave of aliens appearing in each round.
- The game has a title screen and game-over screen.
- The game is able to record high scores across gameplay sessions.
- The game is polished, smooth to play and does not lag or hang.
- Coding standards have been adhered to in all places in the code.

Grade A+, as per grade A and:

The game exhibits evidence of additional work beyond the limits set in the assignment brief, in an attempt to make the game more interesting for the user. This could be in terms of the visual appeal of the game and/or enhanced gameplay features that add more fun to the game or better code design that improves the performance of the game.