GD2P02 Physics Programming	Summative Assignment 2D Physics Project Submission	Date: 13th March 2018 Submission Dates: 5:30pm, 10th Apr 2018

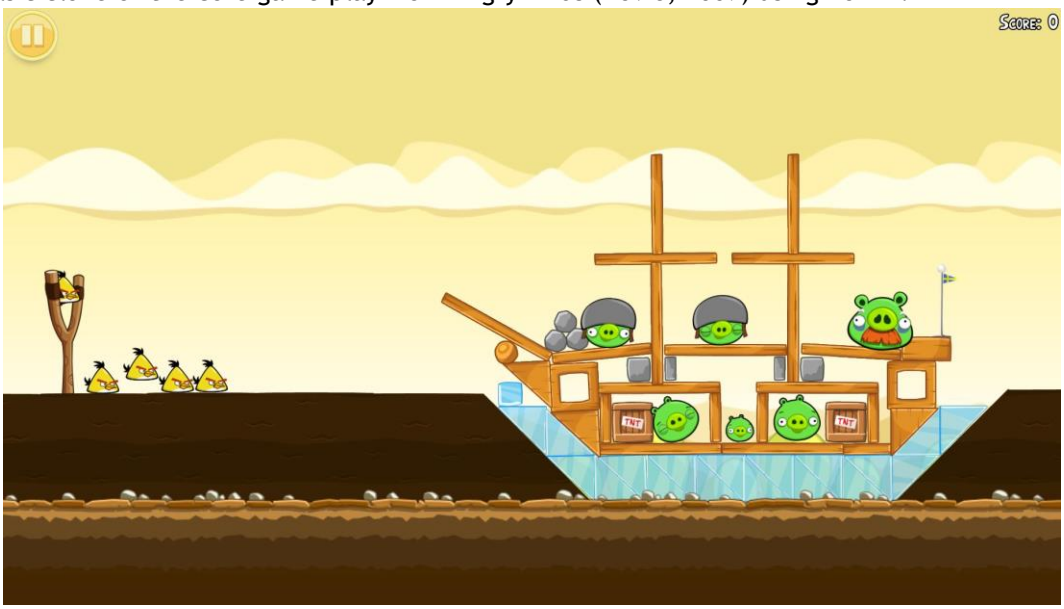
Submission filenames:

`YYYY-MM-DD – GD2P02 – PhysicsProjectSub – Student/Group Name.zip`

`YYYY-MM-DD – GD2P02 – PhysicsProjectPostMortem – Student Name.zip`

2D Physics Project:

Create a basic clone of the core game play from Angry Birds (Rovio, 2009) using Box2D.



This project can be worked on either individually or in groups of 2. Setup at least two scenes of items to knock over, and allow the player to fire at least three different "grumpy birds" at the items in the scene.

The goal of the game is to fling player entities into the scene to destroy all enemy entities. Upon destroying all enemy entities, the player wins the game! If the player uses all of their fling-able player entities without destroying all enemy entities, they lose the game!

Ensure the game can restart appropriately based upon win/lose conditions.

Each scene will contain:

- Destructible block/plank items with physics behaviour
- Indestructible block/plank items with physics behaviour
- Enemy entities
- Indestructible ground
- Fling-able player entities

Employ appropriate usage of:

- Gravity
- Forces
- Rigid bodies

Utilise at least three different types of 'joints' (joints referring to the joints in Box2D) to create different elements in the scenes. These elements can be but not limited to: springs, levers, seesaws, pulleys, ropes, chains etc.

Utilise basic rendering.

GD2P02 **Physics Programming**	**Summative Assignment** **2D Physics Project Submission**	**Date:** **13th March 2018** **Submission Dates:** **5:30pm, 10th Apr 2018**

Release Build Zip:
A release build executable must be zipped and included with the submission. Ensure that project settings are set to Release when creating this build.

Include a readme.txt detailing the general user controls and the gameplay features.

Post Mortem (Individual Submission):
Write a post mortem analysis of the implementation and the software engineering approach applied to the project. The post mortem document should be submitted individually.

Address the following areas:

- Known issues with the release build of the project
- Problem solving strategies
- Software Engineering approaches for verification and validation

Programming Practices:
Utilise the following good programming practices in the development of the project:

- Commenting:
  - Proper header, source and functionality comments
- Build Quality:
  - 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 solution file, project file, and source files are required
    - Required external game data, such as .ini files
    - All other files must be removed
- Runtime Quality:
  - Contains no bugs, does not crash and no memory leaks are present
- Coding Standard Guidelines:
  - The source code is required to adhere to the Media Design School's Game Development Faculty Coding Standard.

Document Formatting Guidelines:
The documents are to be in the form of an Adobe PDF document. The font must be Times New Roman point size twelve with 1.5 line-spacing, and the layout set to fully justified. The header must contain the author's name using left justification, and the footer must contain the page number using right justification.

Submission Checklist:
Source code folder:

- Solution file (.sln)
- Project file (.vcproj)
- Source files (.cpp, .h)
- Intermediate files have been removed

Release build zip:

- Stand-alone executable (.exe)
- Any additional files required to run the game
- Readme file (.txt)

Documentation:

- Post mortem (.pdf)

GD2P02 **Physics Programming**	**Summative Assignment** **2D Physics Project Submission**	**Date:** **13th March 2018**
		Submission Dates: **5:30pm, 10th Apr 2018**

Deliverable:
Place the submission in the Web Drop Box by the time specified for submission.

The work is to be placed in a .zip file. The file structure and file names of the submission must follow the file hierarchy listed below.

- **YYYY-MM-DD – GD2P02 – PhysicsProjectSub – Student/Group Name.zip**
  - ***Source – Angry Birds***
    - **Physics.sln**
    - **readme.txt**
    - …Project and source code, etc.
  - **Release Build – AngryBirds.zip**

- **YYYY-MM-DD – GD2P02 – PhysicsProjectPostMortem – Student Name.zip**
  - **Post Mortem – Student Name.pdf**

GD2P02 Physics Programming	Summative Assignment 2D Physics Project Submission	Date: 13th March 2018 Submission Dates: 5:30pm, 10th Apr 2018

2D Physics Project		%	A (Range: 80% ≤ x ≤ 100%) *Outstanding/Excellent*	B (Range: 65% ≤ x < 80%) *Very Good/Good*	C (Range: 50% ≤ x < 65%) *Acceptable/Satisfactory*	D (0%) (Range: x < 50%) *Unsatisfactory/Below standard*
		100%				
1.1	**Physics Project** **-Box2D middleware** **-Two 2D game world** ***Two scenes** ***Listed entities and concepts are utilised** **-Physics behaviour** **-Required gameplay**	**70%**	Box2D middleware is utilised. 2D game world created: including gravity, forces, rigid bodies, and various elements created with different joints. Newtonian physics utilised. All gameplay as specified in the brief is feature complete.	Box2D middleware is utilised. 2D game world created: including gravity, forces, rigid bodies and elements created with at least three different joints. Newtonian physics utilised. The majority of gameplay as specified in the brief is feature complete.	Box2D middleware is utilised. 2D game world created: including gravity, forces, rigid bodies and elements created with at least two different joints; there may be a few bugs but the simulation does not break. Newtonian physics utilised. An attempt to implement gameplay as specified in the brief is present, some features may be incomplete.	Box2D middleware is not utilised. **OR** 2D game world is not created.
1.2	**Programming Practice** **-Commenting** **-Build quality** **-Runtime quality** **-Coding standards** **-Release build**	**10.00%**	Code is well commented and easy to follow. Solution builds cleanly under warning level 3 in all build targets. File structure of the solution matches the requirements exactly. No intermediate files are submitted. Runtime maintains real-time frame rate and stable physics simulation. Source code follows established coding standards. No bugs present, game does not crash. The program does not exhibit memory leaks. The game release executable zip is present and contains all material required to run the game. Only the files required to run the game (including documentation) are included.	Code is commented. Solution builds cleanly under warning level 3 in all build targets. File structure of the solution matches the requirements. No intermediate files are submitted. Runtime maintains appropriate frame rate and stable physics simulation. Source code follows established coding standards. No bugs present, game does not crash. The program does not exhibit memory leaks. The game release executable zip is present and contains all material required to run the game. Only the files required to run the game (including documentation) are included.	Some commenting is present. Solution builds under warning level 3 in all build targets. Runtime mostly maintains appropriate frame rate and stable physics simulation. Source code attempts to follow established coding standards. Bugs are present; however, the game does not crash. The program does not exhibit memory leaks. The game release executable zip is present and contains all material required to run the game.	Solution does not build. **OR** Significant memory leaks are present. **OR** The game is significantly different from the game features required.
1.3	**Post Mortem** **-Known issues** **-Problem solving strategies** **-Collision detection strategies** **-Software engineering Strategies for verification and validation**	**20.00%**	All known issues are outlined and described; possible resolutions detailed. Problem solving strategies are outlined and clearly described. Collision detection strategies are outlined, clearly described and evaluated. Verification and validation strategies are clearly outlined and described.	The majority of known issues are outlined and described. Problem solving strategies are outlined and described. Collision detection strategies are outlined and described. Validation and verification strategies are outlined.	Known issues are outlined. Problem solving strategies are outlined and described. Collision detection strategies are outlined. No or poor outline of validation and verification strategies.	Post mortem not present. **OR** The post mortem is not in .pdf format. **OR** Post mortem provides poor insight into the development of the project.