COMP 8051 – Advanced Games Architecture
Assignment 4

Work in groups of two

Total marks: 200

1. [10 marks] Create a simple iOS game that uses the Bullet library. Initially a ball should be shown at the top of the window, and a "ground" (could be just a horizontal line) should be shown near the bottom of the window. When the user taps the display, the ball should drop until it hits the ground, then bounce until it comes to a stop. You may want to use the HelloWorld example from the Bullet SDK.

2. [10 marks] Extend the game above as follows. Instead of a ground object, have the ball fall on an incline plane. When the ball hits the incline, it should start bouncing and/or rolling down the incline. The incline should be attached to a box, so when the ball reaches the end of the incline it falls into the box. In addition to tapping to release the ball, the user should be able to change the angle of incline of the plane by dragging it (only one degree of freedom, since the incline is attached to the box at one end).

3. Implement a 2D iOS game that mimics Pong, using the Box2D library and OpenGL ES. You do not need to worry about a 100% faithful reproduction in terms of content, colours, levels, etc. Only the core game essence is required. Use your own drawing functions instead of the drawing systems or debug drawing of Box2D.

    a. [30 marks] The game should work as expected. For the paddles, use dynamic objects that are manipulated by the player(s) or AI. They have collision, but not too much in the way of dynamics because they must be responsive. For this part, you should have the ability to move the user's paddle in a responsive, predictable way.

    b. [30 marks] You will need to implement a play surface and ball dynamics. You will find dynamics for spheres in the "Polygon Shapes" and "Varying Restitution" Box2D demos. Additionally, the "Apply Force" Box2D demo has an illustration of a contained environment with a dynamic object that may be of interest. For this part, you should have a basic environment where an animated sphere or cube serves as the ball and bounces around the environment.

    c. [30 marks] When the ball leaves the play area (end of the display), you will need to detect this and reset or otherwise trigger a game event. This does not need to be elaborate and can be math-based (position of the ball) or use a *Sensor Shape* (consult Box2D manual) for detection.

4. Implement a 2D iOS game that mimics Breakout (Arkanoid), using the Box2D library and OpenGL ES. You do not need to worry about a 100% faithful reproduction in terms of content, colours, levels, etc. Only the core game essence is required. Use your own drawing functions instead of the drawing systems or debug drawing of Box2D.

    a. [30 marks] The game should work as expected. For the paddles, use dynamic objects that are manipulated by the player(s) or AI. They have collision, but not too much in the way of dynamics because they must be responsive. For the blocks, use static objects that are turned off after the ball hits them. For this part, you should have the ability to move the user's paddle in a responsive, predictable way.

    b. [30 marks] You will need to implement a play surface and ball dynamics. You will find dynamics for spheres in the "Polygon Shapes" and "Varying Restitution" Box2D demos. Additionally, the "Apply Force" Box2D demo has an illustration of a contained environment with a dynamic object that may be of interest. For this part, you should have a basic environment where an animated sphere or cube serves as the ball and bounces around the environment.

    c. [30 marks] The "Collision Processing" demo in Box2D shows an example of detecting and responding to collision events. You will need to build a similar system for disabling blocks when they are hit by the ball on screen. When the ball leaves the play area (bottom of the screen), you will need to detect this and reset, or otherwise trigger a game event. This does not need to be elaborate and can be math-based (position of the ball) or use a *Sensor Shape* (consult Box2D manual) for detection.

Submit your entire project, including documentation (at least a README file with any notes and a description of the user controls for each part) to the D2L dropbox. You are to work in groups of three and all will receive the same mark. Your submission should be in a single ZIP file using the naming convention A00ABC_A00DEF_A00XYZ_Asst4.zip, where ABC, DEF and XYZ are your student numbers.