

# Breakout Game Report

Diarmuid Ryan – 11363776

For Assignment 5, I built a Breakout Game for iOS using Swift. This app consists of a single tabbed view controller which allows the user to switch between the gameplay, the high scores list and the game settings. All views are rotatable, however the game will be reset (as is the case when the user tabs away from the game).

In the game settings the user may modify the number of balls in the game, as well as the number of bricks, the bounciness of the balls and the size of the slider. Each of these settings automatically updates values in the `NSUserDefaults`.

The High Scores tab shows the ordered list of the user's score history. This allows the user to keep track of their scores and try to beat their past scores.

The game view allows the user to play Breakout. The user attempts to bounce the balls to hit the blocks. When a block is hit it disappears. The game ends when either the user has broken all blocks or all balls have left the game view. The user is then presented with an Alert View Controller with their score. The score is calculated as the square of the number of blocks broken, divided by the time taken to finish.

Balls are prevented from exiting the game view, except through the bottom, by boundaries along the top, left and right of the game view. Blocks and the Slider are also inserted to the collider as boundaries. Each block is assigned a numeric id so that when a block is hit, the correct block is removed from the game.

During development an interesting bug was noticed whereby the balls would "stick" to the side of the view. This is due to the fact that collider is not built for complex physics. Very small vector components are assigned the value of zero. This results in the behaviour whereby balls travelling slowly will stop when they touch a boundary and balls touching a boundary at an acute angle will have the vector component that is perpendicular to the boundary zeroed and will continue to travel alongside the boundary. To counteract this behaviour, whenever a ball collides with a game boundary which results in a vector component of zero, a small velocity (equal to  $1/10^{\text{th}}$  gravity) is added to the ball in the direction perpendicular to the boundary.

Tapping the view also results in the closest ball to the tap being accelerated upwards. This behaviour allows users to redirect a chosen ball, however it is not strong enough to stop a ball from exiting the view, preserving the utility of the slider.

The figure below shows a general architecture diagram for the operation of the app. We can see that the `NSUserDefaults` are used by all three tabs (although only the Settings VC writes to it). The Breakout VC is the central VC in the system. It creates the Breakout Behaviour and assigns itself as the Breakout Behaviour's Collision Delegate (a protocol). Some static methods contained in the Breakout Model are also used for the sake of code separation. The Breakout Behaviour has two child behaviours, the Collider and the Ball Behaviour. These are controlled by the Breakout Behaviour to maintain consistency with the displayed game view.

