

TTG Team - Still No Name Yet: (Insert Undecided Game Name Here)

Connor Blakey Matthew Burling
C.Blakey@2014.hull.ac.uk M.Burling@2014.hull.ac.uk

Marc Mardare
M.Mardare@2014.hull.ac.uk

Richard Smith
Richard.Smith@2014.hull.ac.uk

October 23, 2014

Contents

1	Instructions for Compilation	1
2	Gameplay Instructions	1
3	Interesting Bits of the Internal Workings	1
3.1	Life System	1
3.1.1	Bugs	2
4	Idea Documentation	3
5	Tips/Mistakes When Using C#	3

1 Instructions for Compilation

2 Gameplay Instructions

3 Interesting Bits of the Internal Workings

3.1 Life System

The player is to receive 3 lives to play with when the game starts. The `Lives` variable is initialised in the preamble and in the `LoadGame` method. The Life functionality is provided in the `Update` method and is triggered by an `if` statement when the object is exiting the bounding box or ‘Arena’:

```

        if (SpritePOS.X > (GraphicsDevice.Viewport.Width
            - SpriteWidth) || SpritePOS.X < 0 ||
            SpritePOS.Y > (GraphicsDevice.Viewport.Height
            - SpriteHeight) || SpritePOS.Y < 0)

```

We make use of the Boolean **LeftArena** to make sure that this code is somehow no run without us foreseeing and to enable a simple while loop for the two case scenarios of the amount of lives left:

```

        LeftArena = true;

        while (LeftArena)
        {
            if (Lives == 0)
            {
                Console.WriteLine("You_have_no_lives
                    _left.");
                gameStates = GameStates.StartMenu;
                LeftArena = false;
                break;
            }
            else if ( (Lives > 0) && (Lives <= 2) )
            {
                Lives--;
                Console.WriteLine("You_lost_a_life , _
                    you_have_" + Lives + "_lives_left
                        ");
                LeftArena = false;
            }

            if (LeftArena == false)
            {
                ResetGame();
            }
        }

```

and to keep evaluating an **if** to check if we lost a life and reset the game with our **ResetGame** method:

```

        if (LeftArena == false)
        {
            ResetGame();
        }

```

3.1.1 Bugs

It was found that we actually interpreted the **Lives** variable in a binary format (0, 1, 2, 3, 4 etc., where we count from zero first) instead of base 10 so when **Lives** was originally set to the value of 3, the user was able to have 4 goes of the game instead. This was simply fixed by initialising **lives** to 2 instead since 0 is counted as a number. We left the algorithm as-is as that's how Mathematics and algorithms should be done.

4 Idea Documentation

5 Tips/Mistakes When Using C#

Name	Mistake	Solution	Time Taken to Resolve
Matt	Incorrectly nested <code>if</code> statements	Think about the ordered sequence of your program (because Rob Miles reminded us)	2:00