**Project I Gameplay Programming I (Due 12th April 2018 @ 11:00 am 20% of 100%)**

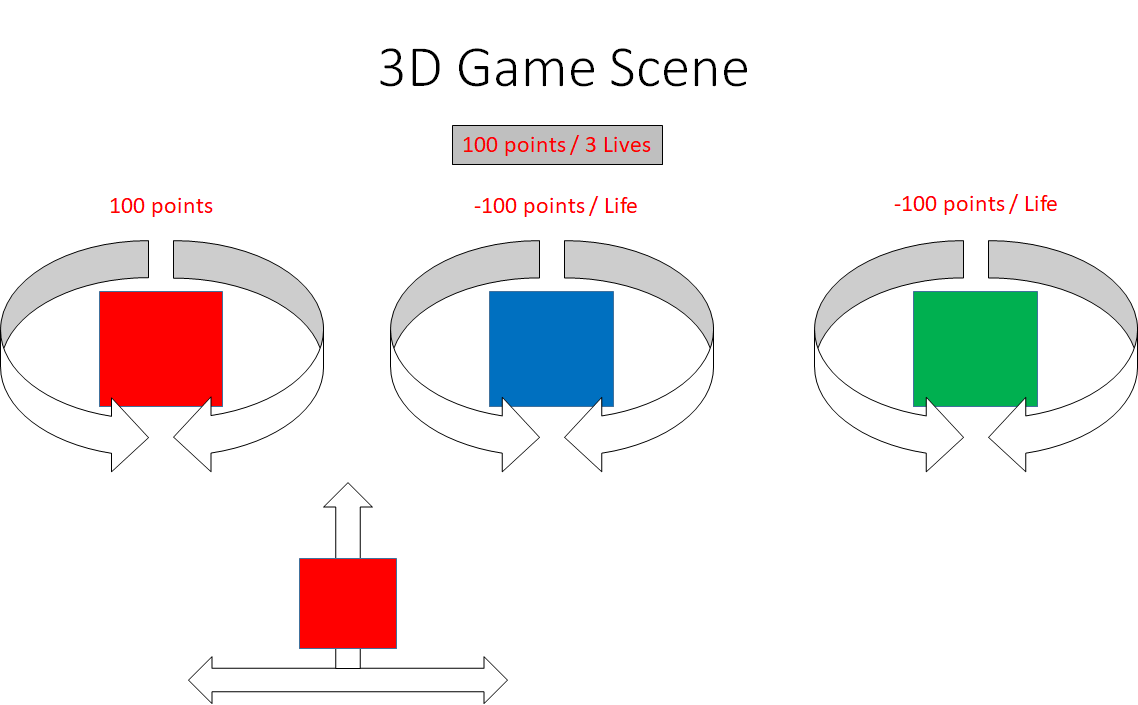
Produce a SFML Game which implements rotation of shape(s) (cubes) within Game loop. Each of shape face will have an individual colour and/or texture.

There are 3 Sample Labs that can be utilised to build game:

* [Lab 12 Starter Kit](https://bitbucket.org/MuddyGames/gameplay-programming-i-lab-12)
* [Lab 13 Starter Kit](https://bitbucket.org/MuddyGames/gameplay-programming-i-lab-13)
* [Lab 14 Starter Kit](https://bitbucket.org/MuddyGames/gameplay-programming-i-lab-14)

Using either arrow, WASD or Mouse a smaller colour cube (the player) will launch / shoot towards the rotating shapes, and if colour matches the face colour, that’s a hit. If the cube and shape face colour match the player gains points, if the colours don’t math then the player loses a life or points.

The player launch/shoot should be implemented like <https://gamemechanicexplorer.com/#bullets-2>



***Project Notes:***

1. Create a Player class which holds the current transformation position of player.
2. Create a NPC class which holds the current transformation position of the NPC cubes.
3. Create a Goal class which holds the current transformation position of the win / add points cube.
4. Initialise the VBO
   1. Create buffer
   2. Bind buffer
   3. Fill buffer with vertex data
5. Create shaders which colour the Player, NPC and Goal cubes.
6. During update loop alter the transformations for the character cubes.
7. During the draw loop switch the shader program (Player Shader, NPC Shader and Goal Shader) and draw the at least 3-character cubes on screen.
   1. Bind buffer Data
   2. Use the Shader Program
   3. Call Draw
   4. Repeat a to c for each Object
8. Add collisions and points and the game is complete.

|  |  |  |
| --- | --- | --- |
| **0 -35** | **35-75** | **75-100** |
| * A selection of the basic game requirements have been implemented to a basic level * Game implementation will achieve minimum functionality * Game implementation may contain some syntax and/or run-time errors * Game implementation code will be poorly commented and/or formatted * Game implementation will contain basic features; application will not be tested properly * Game implementation code will not follow applicable coding conventions * Game implementation will have basic gameplay * Implementation utilises Vertex Array to render(s) | * Game implementation requirement have been implemented to an acceptable level * Game implementation will achieve expected functionality * Game implementation will not contain syntax and/or run-time errors * Game implementation code will be reasonably commented and/or formatted * Game will be tested to a reasonable degree * Game implementation code will follow appropriate coding conventions * Game implementation will have gameplay as specified. * Implementation utilises Vertex Array to render(s) cubes | * Game implementation requirement have been implemented to an advanced level * Game implementation will not contain syntax and/or run-time errors * Game implementation code will be well commented and/or formatted * Game will be expertly tested * Game implementation of code will follow coding conventions * Game implementation will include novel gameplay such as [Cubefield](http://www.cubefield.org.uk) / VFX / SFX. * Implementation utilises Vertex Buffer Object (VBO) to render cube(s) |