**Fall 2021 Honors Project**

* [**Topic selection**](https://forms.gle/Z9b4PUs57bpWiBei9) **[DUE NOV 18th 2021]**

**PART 1**

**Part 1 (Project description and proposal) and 7 (References) is DUE NOV 18th 2021**

**-**[**SUBMIT Proposal FORM HERE**](https://forms.gle/9K4y38pWFiGJGguQ7) **: Thursday NOV 18th 2021 and no later than Nov 23rd 2021**

**-MIDREPORT DUE: Thursday DEC 2nd 2021**

**-Prototype Mockup Demo: Dec 9th 2021**

**-Final Project Due: Dec 14th 2021**

[**https://sites.google.com/view/bmccmakeathon/spring-2021-virtual-make-a-thon-computing-for-social-good-innovation**](https://sites.google.com/view/bmccmakeathon/spring-2021-virtual-make-a-thon-computing-for-social-good-innovation)

**CHOOSE ONE OF THE FOLLOWING THEME**

**1. ZERO HUNGER**

**2. CLEAN WATER & SANITATION AND RESPONSIBLE PRODUCTION**

**3. GREEN CONSUMPTION**

**4. RACIAL JUSTICE**

**5. GENDER EQUALITY**

**6. QUALITY EDUCATION**

**7. GOOD HEALTH AND WELL BEING**

**Submission:**

* **Youtube video of your demo (3 minutes]**
* **Github page with your code** 
  + **Inspiration**
  + **Project Description**
  + **Algorithm**
  + **Challenges**
  + **Accomplishment (what you have learned)**
  + **Future Direction**
    - [**Sample Github**](https://github.com/weidajiang/BMCC-Make-a-Thon)
    - **(Here is a tutorial-** [**https://guides.github.com/activities/hello-world/**](https://guides.github.com/activities/hello-world/)**) .** 
      * [**https://guides.github.com/features/pages/**](https://guides.github.com/features/pages/)**)**

| **You Name** | **Marcos Ortega** |
| --- | --- |
| **Your ID** | **14065613** |
| **Instructor:** | **Dr. Azhar** |
| **Date of Submission:** | **11/ 23/ 2021** |
| **Your Topic:(e.g, RACIAL JUSTICE):** | **Quality Education** |
| **Your new C++ Topic is: (Game or C++ GUI Application or Database or Networking or OpenCV):** | **Game creation using SFML** |

**A. Project Description and Problem Statements:**   
Hoping to create a dungeon crawler-type game. The player goes through the dungeon, defeating monsters with math-based actions/solutions. Arithmetic problems. The player rescues historical figures (mathematicians). I want players to learn not just math but its history and progenitors.

Unfortunately, I wildly underestimated the process of creating a game but I was able to create a simple game collecting balls that includes a health bar and score. While not the original vision, it did allow me to learn the basics and become more comfortable with SFML. SFML includes numerous classes to render, display, create shapes, set font and display text, account for movement, keystrokes, etc.

**B. Proposal:**

Weekly plan

Week 1- Research and learn the environment. Code layout for level one.

Week 2- Create a syllabus/layout of learning material and add characters to the code

Week 3- Implement lessons to level (addition problems, subtracting) and do testing

Week 4- Polishing. Continuously testing and adding features/code to build further.

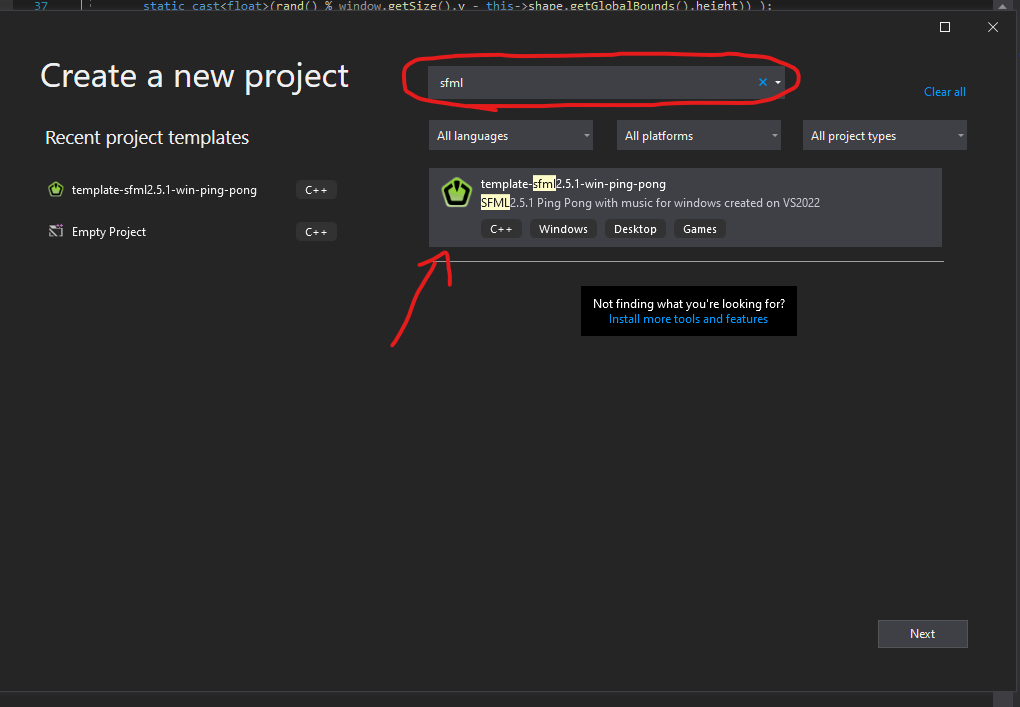
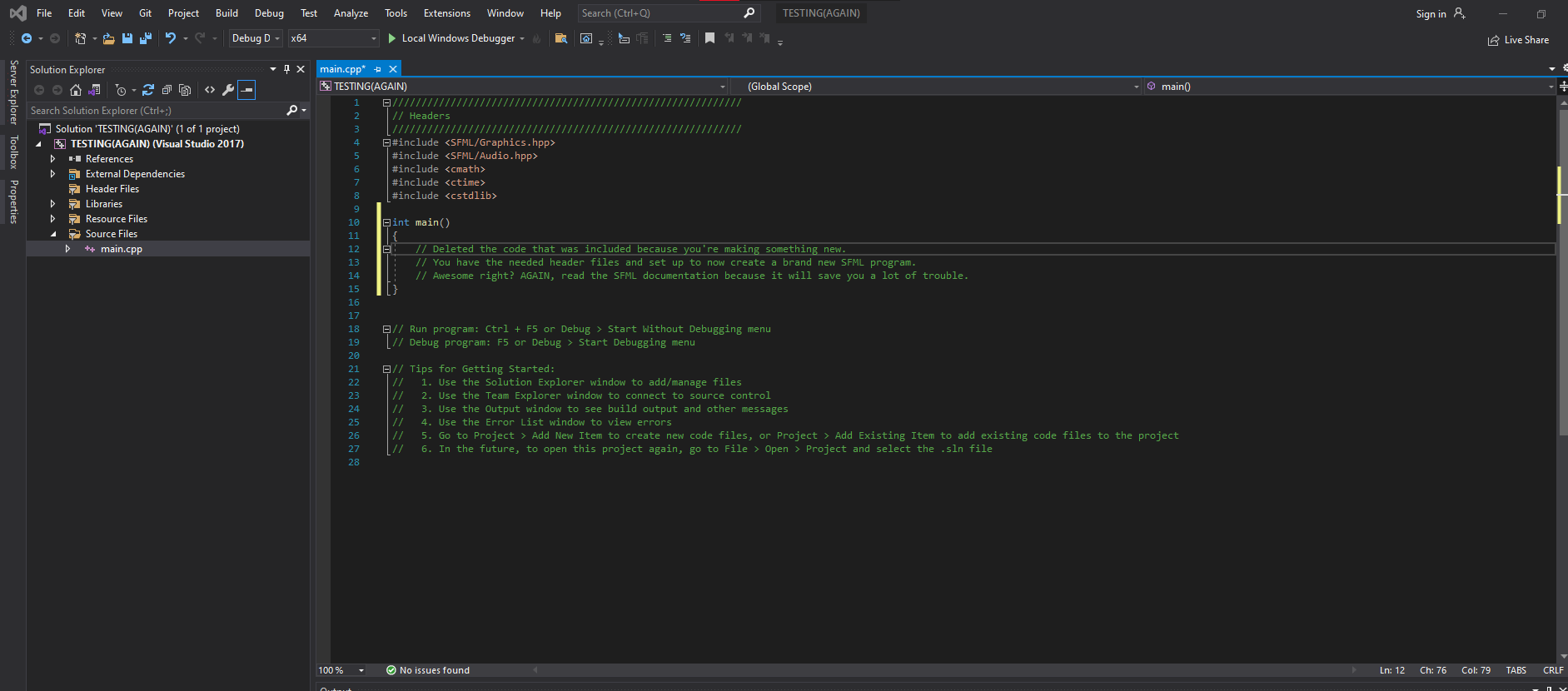
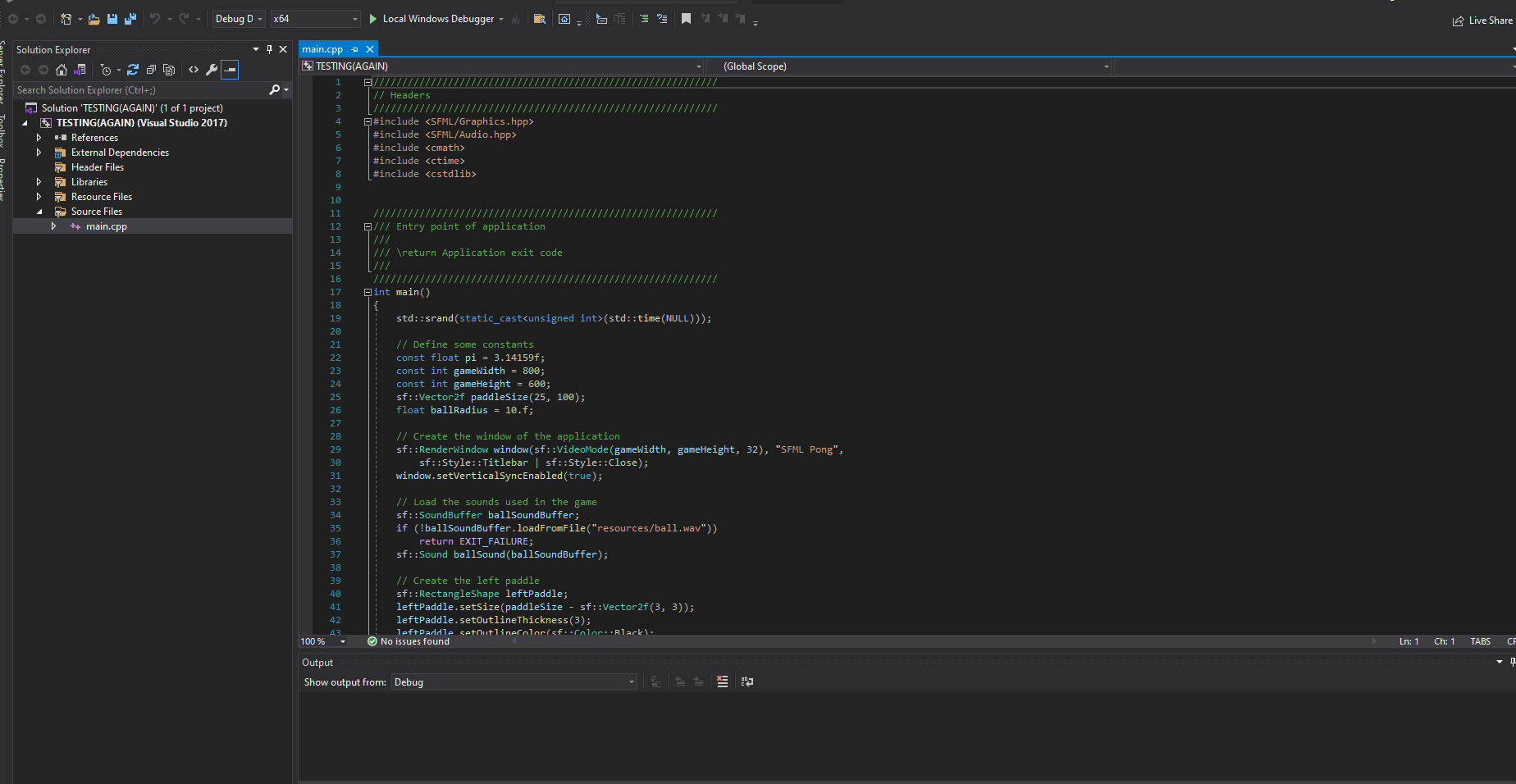
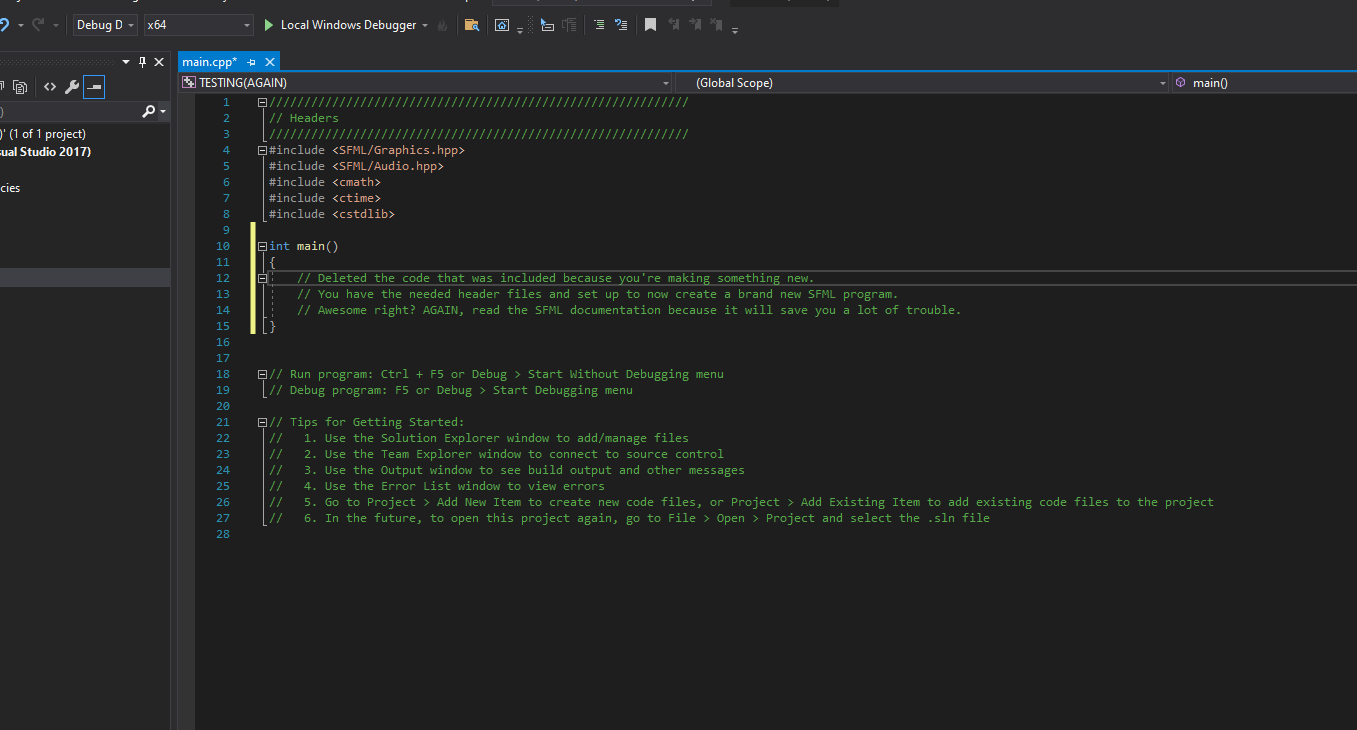
|  | To-Do List | Completed? |
| --- | --- | --- |
| Week 1 | Learn environment. Level design. Research. | Yes |
| Week 2 | Create a lesson plan. Create player and mathematician NPCs. | NO |
| Week 3 | Lesson | NO |
| Week 4 | Polish the (a?) program | Yes(?) |

2. Software Setup:

[Installation description, links, and screenshots]

* Used **Visual Studio 2019** (don’t use 2022, it’s a hassle). Here’s the link below for VS 2019: [Link for Visual Studio 2019](https://visualstudio.microsoft.com/thank-you-downloading-visual-studio/?sku=community&rel=16&utm_medium=microsoft&utm_source=docs.microsoft.com&utm_campaign=download+from+relnotes&utm_content=vs2019ga+button)

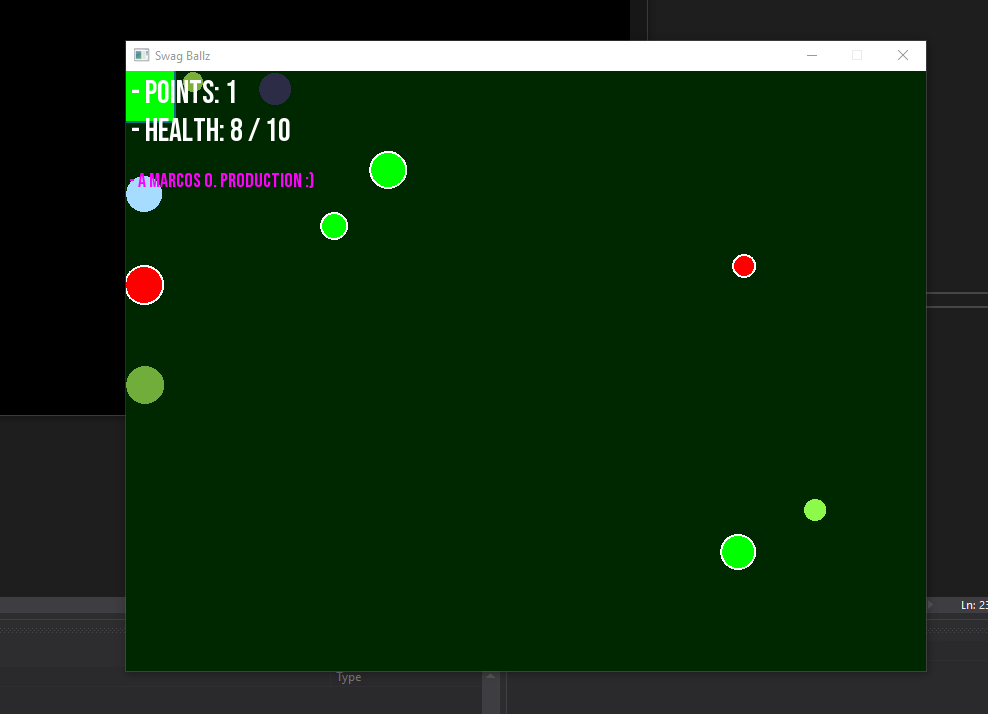
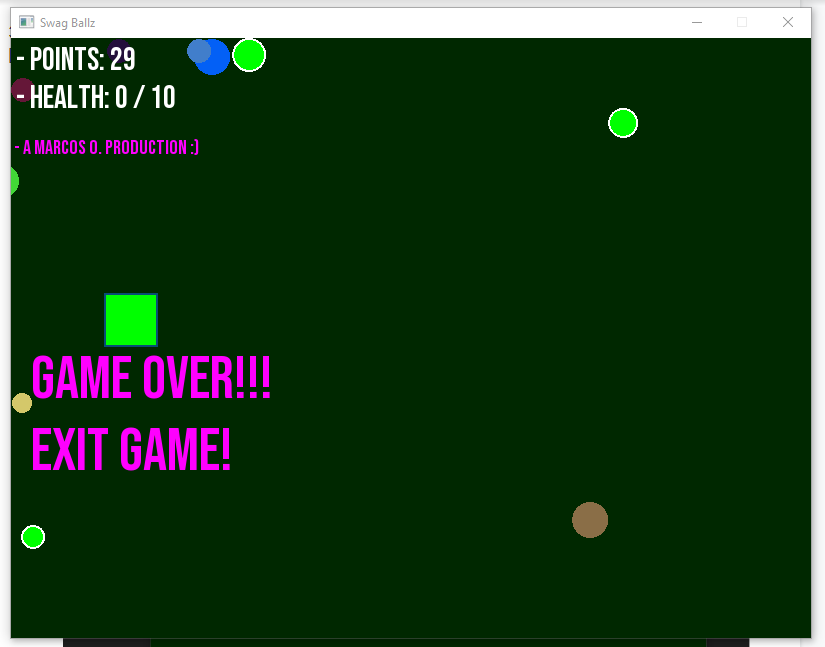
**Note**: This is NOT Visual Studio Code. That’s a whole other thing. This is Visual Studio

* SFML was a bit tricky to set up but the video below made it a breeze.  
  [How To Install SFML with Visual Studio 2019 In 2021 [C++]](https://www.youtube.com/watch?v=neIoDQ71yb0&list=FLldtBAV-EjujSYA69IV_Z1Q&index=1&t=188s&ab_channel=RoguesDivision)
* Assuming you set up VS and SFML, you want to create a new project and look for a template SFML called ping-pong and download it. 
* While the template is a game by itself, it essentially sets up a project for you to erase the code and then create your own work. 
* 

Cool, huh?

3. Algorithm/ Software Design/Interface Design:

[Algorithm for your program, Interface Design (How does your interface look like),

* The game starts with a player able to move their character on the screen. The object is to collect the balls while avoiding red ones with a white border to earn points. You can move the player using keys WASD. Red balls damage the player while green balls (also with borders) heal the player. All other balls will earn the player points. The game is over when player health reaches zero.
* To quit the game you can close the window (clicking on X on the window) or press the ESC key. The game looks simple:  
  
* Game over screen is simple:  
  

4. Code (Later) github

All the topics covered in the class (Classes, Inheritance, Pointers, Exception Handling, File I/O)

Must give credits to other’s code.

| #include “Game.h” **using** **namespace** std; // Game.h file includes everything needed to run so just on file **void** **main**() {  // Initialize random seed; used for random number generators  srand(static\_cast<unsigned>(time(0)));  // Initialize game object  Game game;  // Game loop  while (game.running())  {  game.update();  game.render();  }  // End of application  return 0;  } |
| --- |

5. Screenshots: (I wasn’t sure what was being asked so I included a short description of each class in the game).

[Outputs]

| Descriptions | Screenshots |
| --- | --- |
| Game.h file. Contains all the needed header files for vectors, string streams( for text). Game class will display text and on-screen for variables such player health and score.  Thus, game hfile contains player.h and  Swagball.h file too.  Game class renders and display window. Uses an update and render function to draw images to screen. These functions will be used in main.cpp in a while loop. Loop ends when player ends the program by closing window or pressing ESC key. |  |
| SwagBall.h file. Contains SFML header files for graphics and audio. I wasn’t able to implement a sound file, unfortunately, but it was set up (#include <SFML/Audio.hpp>).  Left some code to implement direction for the ball movement. To be used with update function for swagball; will use a switch statement for direction and move accordingly and change direction 180 to “bounce” off the window border (if going N, hits N border, then ball direction set S). |  |
| Player.h file. Holds variables for health, movement speed as well as functions for gaining health and taking damage. These functions are to be used in Game class’s collision function. |  |

6. Challenges and Solutions:

* The first challenge I had was implementing and using SFML. I had no real grasp on where to start so I looked up videos. The biggest help was a classmate. She helped install the right Visual Studio (2019) and how to set up SFML too. Asking for help is key when you’re stuck.
* The second challenge was understanding SFML. I used a tutorial to learn the basics. It is time-consuming to watch a bunch of videos so you want to commit time to learn it.
* Testing, testing, testing. With so many changes to your code, you want to make sure every step along the way is compiled/working before adding more code on top.
* Learning SMFL was a hassle but tutorials help and reading the SFML documentation on its modules and classes is essential for a deeper understanding.
* One of the biggest issues near the end was the movement of the balls themselves. Stationary balls felt boring so I wanted to add movement to the balls to engage the player. The balls move but only towards the NE corner. Could be resolved using enumeration (N, S, E, W) to move the balls in different directions. Also, ball movement needs to account for the borders of the window/game.
* I was not able to implement a pause for the game. TBC
* The biggest setback was implementing this towards a Quality Education theme. While I could not achieve that, I was able to make a working game. Small steps but forward steps, I guess :)

7. References:

* [Link for Visual Studio 201](https://visualstudio.microsoft.com/thank-you-downloading-visual-studio/?sku=community&rel=16&utm_medium=microsoft&utm_source=docs.microsoft.com&utm_campaign=download+from+relnotes&utm_content=vs2019ga+button)9
* Guide: [How To Install SFML with Visual Studio 2019 In 2021 [C++]](https://www.youtube.com/watch?v=neIoDQ71yb0&list=FLldtBAV-EjujSYA69IV_Z1Q&index=1&t=188s&ab_channel=RoguesDivision)
* CodeMaid is greatly useful for cleaning up and formatting code for you. Saves a lot of time with indentation and organization: [CodeMaid](https://www.codemaid.net/documentation/)
* Essential reading: [Learn SFML](https://www.sfml-dev.org/learn.php)