**Spike:** Task 22.P

**Title:** Collisions

**Author:** Mitchell Wright, 100595153

**Goals / deliverables:**

To demonstrate .

Items created during task:

* Code, see: \22 - Spike – Collisions\SDL2

**Technologies, Tools, and Resources used:**

* Visual Studio 2022
* SourceTree
* GitHub
* SDL2
* Lecture 6.2 – 2D Collision Tests

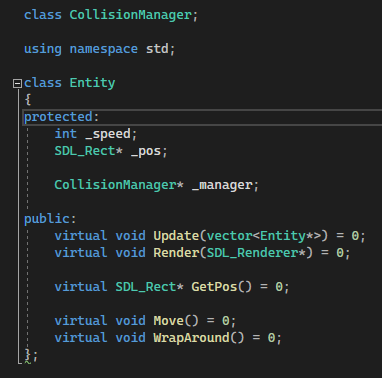
**Tasks undertaken:**

* Implement Moving Entities
* Implement Collision Data Structure
* Implement the Collision Tests
* Commit to Git

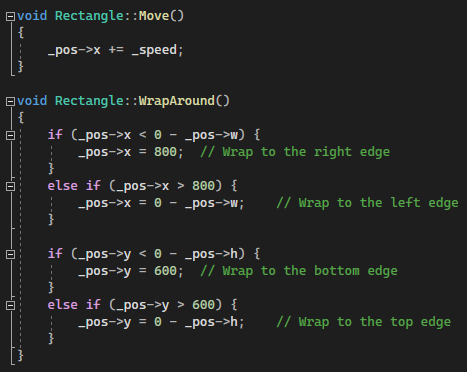
**What we found out:**

1. Implement Moving Entities:

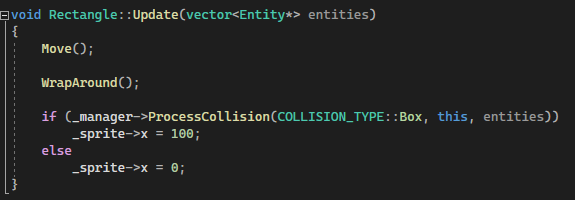
In order for everything to work smoothly later on, I created an Entity super-class with most fo the main functionality needed, then created a Circle and Rectangle class.



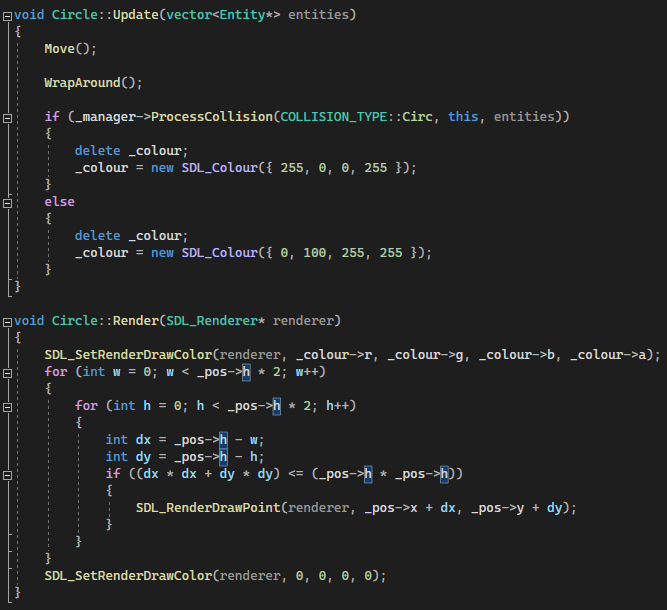
Move and Wrap Around are pretty simple, basically if the entity goes beyond the bounds of the screen, move it back to the other side, otherwise it moves from left to right at whatever speed is set on instantiation.



Update and render for the rectangles work basically the same as in the previous Sprite task.



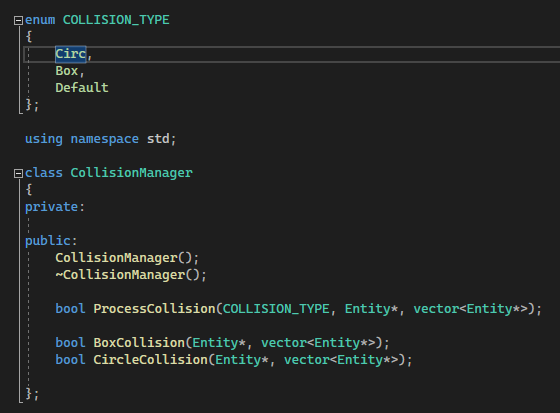
Circles work a bit differently, being drawn rather than a rendered sprite. This also means that update changes colours rather than changing sprite positioning.



Objects are then run in an update/render loop in main as part of a vector.

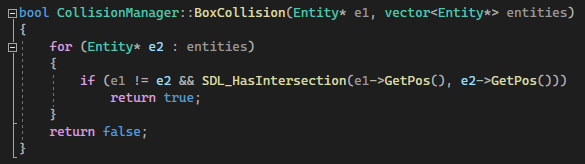
1. Implementing the Collision Manager:

The task sheet for this spike said something about making it extensible. Didn’t really understand what that meant, so I made a collision manager with different functions for circle or box collision and an enum to select which one was appropriate at any given time.

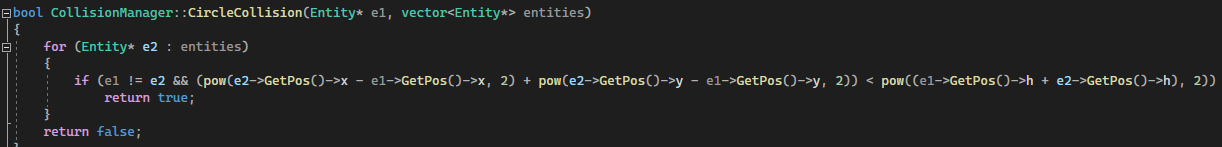


1. Implementing Collision Tests:

The actual collision tests themselves weren’t too tricky. For Rectangles, SDL2 has a built-in function allowing for finding an intersection.



Circles are a bit more fiddly, and I essentially converted the pseudo code provided in the lecture into C++ and used that to check if the distance between circles was less than their combined radii.



1. Commit to Git:

