## **Spike:** Task 22
## **Title:** Collisions

**Author:** Hayden Whiteford, 104001272

### Goals / deliverables:

Either extend your Sprites and Graphics spike or create new 2D entities which you can implement a collision management data structure around:

1. Implement at least box-based (axis-aligned rectangles) intersection testing, and circle-circle intersection testing
    - Display two or more boxes, one of which is a fixed position box and the other(s) moving. Detect, using axis-aligned rectangle testing, one box overlaps another (collision detected) and visually display the "collided" status to the user
    - Display two or more circles where one or more of the circles are moving. Detect, using circle-circle collision testing, when one circle overlaps another, and change the visual appearance to represent that a collision has occurred.
2. Incorporate extensibility in your collision data structure, so that you would be able to swap out collision detection method for another

### Technologies, Tools, and Resources used:

- Xcode
- SDL2
- Some of the code in AI4G class used as inspiration for Circle and Square

### Tasks undertaken:

- Created a Box and Circle class, each able to handle collision
- Created a DrawCircle function
- Made 2 fixed and 2 moving squares and circles each, with moving circles and square coming from the corners
- Change the colour of the respective shape to red upon collision

### What we found out:

In order to figure out the math regarding how circle collision works, I ended up digging through the AI for games agent code, to find the solution: you need to compare the squared distance between the centre points and compare it to the squared sum of the radii (we keep it squared as it's functionally the same result as if we Square rooted them).