**Spike:** 11

**Title:** Collisions

**Author:** Ryan Chessum, 102564760

**Goals / deliverables:**

* Code, see /22 – Spike – Collisions /Collisions/
* Spike Report

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

List of information needed by someone trying to reproduce this work

* Visual Studio 2019
* C plus plus reference (<https://www.cplusplus.com/reference/>)
* SDL2

**Tasks undertaken:**

* Download and install Visual Studio
* Create a new C++ project
* Set up SDL window init
* Create Class for object to move
* Create collision detection system

**What we found out:**

In this spike I made a basic program that calculates the collisions between circles and circles and rectangles and rectangles.

You can detect if two rectangles are colliding by getting positions for the top, bottom, left and right of the two rectangles. Then you check if the position of the first rectangles bottom is less than the position of the second one’s top (top to bottom y cords). Then you check if the first rectangles top is greater than the bottom. Then if the firsts right is less than the second ones left and last if the first ones left is greater than the second ones right. If all of these fail then the objects must be within each other. It also doesn’t matter on the order, just as long as they all fail.

Text

Description automatically generated

For detecting collisions with circles, you get the difference in each coordinate squared and add them together. Then if the result is less than both circles’ radiuses squared the two circles are inside each other.

Text

Description automatically generated

For creating this in working code I made an object that moves around.

Text

Description automatically generated

Text

Description automatically generated

If the object is supposed to be moving it will update its position using a global instance of a collision checker. If it detects a collision the object will be moved back out of the other object and it’s velocity in that direction will be reversed.

The shape of the object can be changed with 1 and 2. The collision checker is a state machine. If the collision state is on Square it will use rectangle on rectangle collision detection. If it is on circle it will use circle on circle collision detection. This can be expanded on in the future for more shapes.

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

The state enum will also decide what render method to call for the object.

Text

Description automatically generated

In the game loop the screen will first be cleared. Then it will update the moving object. Then it will render each object to the screen.

Chart

Description automatically generated

Graphical user interface, application

Description automatically generated