# SOFTWARE TESTING AND MAINTENANCE

# Project

# Finn Hawkes

# 91050039

## Overview:

For this project I will run over the supplied code for the game provided by the stake holder and then from that break the code down to run tests on it and see how the game functions and if it is fully bug checked and run multiple different tests to check to see how the code works under all instances and circumstances.  
I will have the code on GitHub to see the project   
The GitHub link is: <https://github.com/boghopper2/BowlingGame>

## Test Plan:

For the code once I can confirm that it works in the correct method I’ll be running some tests on the project,   
the tests I intend to run will be in relation to how the game is run and what rolls they make and what pins they hit to confirm the final score is correct when they finish the game to ensure that the scoring maths are correct.

The tests that I will be running are:

* Hitting only one pin on every throw in a game
* Hitting no pins for an entire game
* Checking to see if it is possible to complete the perfect game
* Checking to see if getting a spare gives the correct score
* Checking to see the score is correct making different bowls
* Checking to see what the score is once hitting one strike
* Checking to see how it goes while making two strikes

These tests will be made in the unit test case to ensure that the expected outcome comes true after running the code.

For the first test *hitting only one pin on every throw in a game* because during a game there are 20 throws of the ball and the idea is to only hit one pin each throw at the end of the game the idea is to have a final score of 20.

For the second test of *hitting no pins for an entire game* at the end of the 20 throws and all being misses the final score of the game will be zero

For the third test *checking to see if it is possible to score a perfect game*, this test is a good one as there are extra factors when it comes to getting multiple strikes in a row as there is an additional multiplier, if you are to roll a strike each time then it should be a game of 300

Forth test is testing to see *the outcome from scoring a spare* the expected outcome from the spare run will be 24 after throwing one spare and then after rolling an additional throw of a 7 to see if the correct multiplier is in effect

The fifth test is checking to *see if rolling different points adds up to the correct amount* this is to test if when throwing multiple different rolls that the total score is correct over the period of the run

The sixth test is *checking the score after only scoring one strike* this is to ensure once rolling a strike and then rolling another go after to ensure the strikes multiplier works after the roll and ads up to the correct amount

And the final/seventh test is to *see what the final score is after scoring two strikes* to ensure the additional multiplier as previously mentioned in the past test to see if the extra multiplier still works properly.

## Unit tests:

Test one:

Test two:

Test three:

Test four:

Test five:

Test six:

Test seven:

## Python Doc comments:

## Summary: