# Bowling Game Project – Summary Report

## Overview

This project was about testing, fixing, and improving a Python program that calculates scores for a ten‑pin bowling game. The main goal was to check if the program followed the bowling rules, find any bugs, write tests, and make the code easier to read and use.

## Testing Approach

I wrote a test plan that included 10 different situations that can happen in a bowling game:

1. All gutter balls (score = 0)  
2. All ones (score = 20)  
3. One spare with bonus roll (score = 16)  
4. One strike with bonus rolls (score = 24)  
5. Perfect game – 12 strikes (score = 300)  
6. Spare in 10th frame with bonus roll (score = 17)  
7. Strike in 10th frame with two bonus rolls (score = 19)  
8. No rolls allowed after game is finished (should raise an error)  
9. Negative roll is not allowed (should raise an error)  
10. Rolling more than 10 pins is not allowed (should raise an error)

These tests were written in game\_testing.py using pytest. All tests passed successfully.

## Debugging & Fixes

When running the tests, I noticed the code could sometimes crash with an index error when checking for strike or spare bonuses. I fixed this by making the strike and spare bonus calculations safer so they don’t go outside the list of rolls.

## Refactoring

I made the code easier to read and safer by:  
- Adding type hints (like int and bool)  
- Writing clear comments and docstrings  
- Moving strike and spare logic into helper methods  
- Using underscores for helper methods to show they are internal  
- Organising the code into clear sections

## Version Control (Git)

I used Git to track changes. I made commits after writing tests, fixing problems, refactoring the code, and adding documentation. Commit messages explained clearly what was changed.

## Documentation

I added docstrings to all methods and generated documentation using pydoc with the command:  
python -m pydoc -w bowling\_game  
This created a file bowling\_game.html that explains the code in a simple way.

## Conclusion

The bowling game program is now fully tested and works correctly. It follows the bowling rules, handles edge cases, and has clear documentation. The code is easier to understand and ready for future improvements like adding a user interface.