# Educational Bowling Game Test Summary Report

## Test Plan Identifier

Educational Bowling Game Test Summary Report v1

## Summary

As listed in Educational Bowling Game Test Plan v1, BowlingGame.py is the test item. This test item was tested on a computer running Windows 10, and tested debugged using Python and Wing IDE.

After BowlingGame.py gets tested, debugged, and refactored, documentation regarding the game and Main.py were generated using Pydoc/Pythondoc as an HTML file. The result of the tests are summarized below.

Testing went according to plan, and thus no variances were experienced.

The tests covered 100% of BowlingGame.py, using six test cases that tests the few mechanics that the game has to offer. They were conducted within the course of 30 minutes.

When running Main.py for the first time, several errors were present. The sole cause of these errors was due to typos. After fixing these errors, all six test cases reported to be successful. Refer to “Features to be tested” in Test Plan v1 for more details.

BowlingGame.py has proven to be functional after testing, and is ready to be fully developed to have a GUI and a system to receive input data. Although there is a medium risk regarding how the game handles processing roll scores. If the size of the list of roll scores does not equate to 10 frames, which is a complete game in bowling, the game will try to grab a non-existent roll score (IndexError).

The developers must ensure that the list of roll scores must be greater than 11, and less than 22. This is to cover for two scenarios: a perfect game (only 12 balls thrown), or an ordinary game where the 10th frame scores a strike as it gives two extra throws for the player (up to 21 balls thrown, assuming no strikes were achieved before the 10th frame).

The good news is that ordinary games will automatically fall within said range, and so players will not experience any errors with how the mechanics are implemented. Developers must be careful if they decide to simulate or test a quick game.

Testing, debugging, and refactoring were conducted in one sitting, and were finished in a reasonable time. No variances, change of plans, nor losses in terms of time or money. Setting up and using Python and Wing IDE on a Windows 10 computer as the test environment proved to be easy, and free of any issue.

Verification and approval of the test results depend on the author of this document.

To view any specific comments regarding the debugging of the game, refer to the following GitHub repository: <https://github.com/RayJr-Personal/game-test-wc>