Yahtzee!

Test and Validation Plans

CUP O' JAVA

Connor King

Hunter Banks

James Vuong

Course: CPSC 224 - Software Development

Instructor: Aaron S. Crandall

I. Introduction

I.1. Project Overview

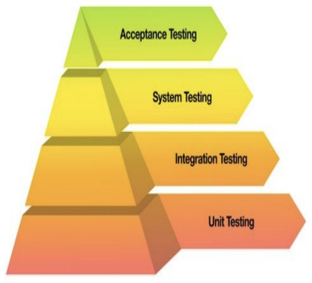
The purpose of this project is to create a Java based application which is operated from a graphical user interface (GUI) to play the game of Yahtzee. The project will be a complete version of the game which includes the ability for multiple users to play together. The project will be true to the official Yahtzee rules and point system. Other features include the ability to change how many sides each die has, the total number of dice, and the number of rolls each player is allowed. Being able to easily test the program allows for efficient debugging and ensuring all aspects of the game are working as expected.

I.2. Scope

This document discusses the testing plans used to test our functions created for the Yahtzee GUI, using Java Swing, and the game functions themselves, using JUnit. Java Swing is an API for providing a GUI for Java programs. JUnit is a unit testing framework for Java and is very important in the development of test-driven development. This document will discuss our testing strategy, test plans, and the different kinds of tests (e.g. unit testing, integration testing, system testing, etc.) that will be implemented to test the functionality of Yahtzee!

II. Testing Strategy

The program which is to be tested will be a fully functional Yahtzee program which uses a GUI for all operations in the game. The variety of tests will help prevent and find errors from small mistakes (i.e. wrong side of die) to larger mistakes where the game isn't being correctly played. The importance of unit tests is being able to automate this process without having to play countless games of Yahtzee to find bugs. This will initially be more time heavy, when writing tests, but will allow the majority of the program to be tested automatically.



The High priority functional requirements are all very high-end tests, which will fall in between system testing and user acceptance testing. Included in this is, "Play an accurate game of Yahtzee, true to the game rules" and "Game is played on a graphical user interface (GUI)". These requirements rely on the foundational Unit & Integration tests to provide a framework for the user testing.

Being able to test the simple actions in the program is what will keep the more complex tasks operable and accurate. We must be able to rely on the basic operations, and testing with good tests is what will make this possible. The tests should fail when they're supposed to and indicate what went wrong. A log will be kept to keep track of when tests have been completed and if any fail.

III. Test Plans

The testing will follow a bottom up path. First, the core simple tests will happen. The program wouldn't be able to operate without these. Next the tests will begin to get more complicated and start testing whole operations in the game. Finally, the whole game of Yahtzee will be tested to ensure all aspects of the program are working as expected.

III.1. Unit Testing

To perform unit testing, we will be testing the reading from the configuration file. We will also be testing the correctness of the scores. Connor will write these unit tests and all group members will review these tests before they are considered "complete".

III.2. Integration Testing

To perform Integration Testing, our group will be performing tests on things like generating the upper scorecard based on the user's configuration. This integration test will test multiple components: validating the user's entered configuration for the number of sides on a die (i.e. properly allocating space for 1's Score, 2's Score, 3's Score, … , N's Score), making sure that the upper scorecard has generated the right amount of die scores, and validating that the scores for each side of the die are correctly computed. Our overall integration tests will cover testing the upper scorecard, lower scorecard, creating a die hand, and more.

III.3. System Testing

To perform System Testing, our group will come up with a list of use cases that could possibly happen for a user while they're playing Yahtzee on the GUI. Essentially, our developers will test the program of GUI Yahtzee but from the user-end perspective, searching for potential bugs and errors. Our team will then run through each of the possible scenarios and make sure that those scenarios are functional and follow the project requirements/guidelines. All exposed bugs will be reported to the developing team and fixed as soon as possible.

III.3.1. Functional testing:

To perform functional testing, we will be testing the GUI to see if it works the way we intend it to. We will do this by having all members of the group run the program and discuss if the GUI looks and functions the way we want it. The requirements of how the GUI should look like and function will be taken from the functional requirements that we have agreed upon. This test will also be finalized by our final test, which is user acceptance testing. This will allow us to see how a user outside of the group uses and interacts with the GUI.

III.3.2. Performance testing:

To perform performance testing, we will be pushing the limits of the program. One test we will perform is when a user inputs a large number for each configuration. This will test how the program handles excessively large numbers of dice, turns, and sides on the dice. We will also be testing if the user enters too many or too little y's or n's when asked which dice to keep.

III.3.3. User Acceptance Testing:

To perform User Acceptance Testing, we will have randomly selected 1-3 people, who are not members of our team, play a full game of Yahtzee on the GUI that is generated from our code. The testers will run through the game checking for any display errors or any functional errors that they may come across. All exposed bugs will be reported to the developing team and fixed as soon as possible.

IV. Glossary

GUI	Graphical User Interface.
Java	Programming language which the project will be developed in.
JUnit	The testing framework which will be used to run the tests.
Java Swing	The interface used to develop the graphical user interface

V. References

Sefan, Petr, Vojtech Horky, Lubomir Bulej, and Petr Tuma. “Unit Testing Performance in Java Projects: Proceedings of the 8th ACM/Spec on International Conference on Performance Engineering.” ACM Conferences, April 1, 2017. https://dl.acm.org/doi/abs/10.1145/3030207.3030226.

Crandall, Aaron. “Unit Testing II, Slide 8.” Blackboard, January 2022. https://learn.gonzaga.edu/ultra/courses/_966960_1/cl/outline.

Indeed Editorial Team. “What Is System Testing? (plus How to Perform a System Test).” Indeed Career Guide, July 22, 2021. https://www.indeed.com/career-advice/career-development/system-testing.

“Unit Testing Swing Applications with Marathon/Javadriver.” Test Automation Tools for Java and Web. Accessed April 5, 2022. https://marathontesting.com/blog-posts/unit-testing-swing-apps/.