**Test Plan**

**For MasterMind**

Devin Taniguchi

Rodolfo Diaz

Zijun Zhao

Contents

[**1.** **Test Plan Identifier** 4](#_Toc453337682)

[**2.** **Glossary** 4](#_Toc453337683)

[**3.** **Introduction** 4](#_Toc453337684)

[**4.** **Test Items** 4](#_Toc453337685)

[**5.** **Features to be Tested** 4](#_Toc453337686)

[**5.1** **Combo Generation (H)** 4](#_Toc453337687)

[**5.2** **Menu (H)** 4](#_Toc453337689)

[**5.3** **User input (H)** 4](#_Toc453337691)

[**5.4** **Results output (M)** 4](#_Toc453337693)

[**6.** **Features not to be Tested** 5](#_Toc453337695)

[**7.** **Approach** 5](#_Toc453337700)

[**8.** **Item Pass/Fail Criteria** 5](#_Toc453337702)

[**9.** **Test Deliverables** 6](#_Toc453337712)

[**10.** **Remaining Test Tasks** 6](#_Toc453337717)

[**11.** **Environmental Needs** 6](#_Toc453337719)

[**12.** **Responsibilities** 6](#_Toc453337720)

[**13.** **Schedule** 6](#_Toc453337721)

[**14.** **References** 6](#_Toc453337724)

[**14.1** **Project17A\_1 document** 6](#_Toc453337725)

[**14.2** **Mastermind System Requirements Specifications** 6](#_Toc453337726)

1. **Test Plan Identifier**

MasterMind

Test Plan :: Version Number < 0.0.1 >

Version Date :: 6/02/2016

Authors:

Devin Taniguchi // Email: [sukiyaki2@gmail.com](mailto:sukiyaki2@gmail.com)

Rodolfo Diaz // Email: [RudyDiaz02@gmail.com](mailto:RudyDiaz02@gmail.com)

Zijun Zhao // Email: [gre9986@gmail.com](mailto:gre9986@gmail.com)

1. **Glossary**

A.I.: Artificial Intelligence

Module: A Class within the source code

Memory: Physical memory component within the hardware

NetBeans: Program used to develop source code in various programming languages

PVP: Player Versus Player; A functionality which enables various users to interact within the system

1. **Introduction**

The purpose of this version of the Test Plan is to explicitly state which items are to be tested within MasterMind. The system will be tested thoroughly based on the MasterMind Requirement Specification document.

The key points of testing will be primarily based on the class relationship between modules. Furthermore, testing will be conducted through the user interaction with an artificially made input-output values alongside user input as well. Stress testing will also be conducted to test the reliability of the memory capacity and data processing.

1. **Test Items**

The following items will be tested to ensure proper functionality within the system specification:

1. Module Testing: Verify that the expected input through parameters generated desired output
2. Null Testing within modules: Testing that the desired output is generated when passing a nothing through parameters.
3. Module to Module Testing: Ensure that each module interacts accordingly with other modules.
4. Stress Testing:
   1. Data Processing: Test processing capabilities to ensure reliability.
   2. Memory: Test to ensure that the memory allocated satisfies the memory needed to run the program.
5. **Features to be Tested**
   1. **Combo Generation (H)**

As the main part of the system/game, the combo generator that makes the set of colors the user has to guess is extremely important. If the generator is too simple then the game would be too easy for the user. In addition, if the generator doesn’t work then the entire game does not work.

* 1. **Menu (H)**

The menu is the skeleton of the system/game. If this feature does not work then the user will not be able to navigate to any part of the system/game thus rending it unusable.

* 1. **User input (H)**

User input ties in with everything for the system/game and is absolutely needed. Without the game cannot function.

* 1. **Results output (M)**

The results are a list of the user entered combinations for guessing the computer generated code. This is outputted whether the user wins or loses so while it is nice it has no bearing on the end result of the game.

1. **Features not to be Tested**
   1. **AI combo guessing (L)**

As it is not fully implemented/completed it will not be part of the features used. Will be completed later and included in a later release.

* 1. **Instructions (L)**

As the instructions are simply text being outputted to the terminal there is no need to test this function.

1. **Approach**

To begin with the system can be managed on most compliers but we will be mainly using Netbeans for the management. This means that the testers need to be able to code in C++ while having basic knowledge of the use of Netbeans. Additionally a basic computer, at least to current standards, is required to run it. As each unit/feature is tested it the overall system will over go a regression test to ensure that any changes in the progress of testing the unit have not effected the system. After the system is properly tested as a whole the game will then be tested for any bugs that might occur in the process of playing it.

1. **Item Pass/Fail Criteria**
   1. **Pass**

* Features with little to no bugs
  + Able to navigate the menu properly
  + Combination generated randomly
  + User is able to input everywhere it is needed
* System Processes Properly
  + Can identify/analyze user’s input
  + Follows the rule of the game
  + Outputs proper responses/results

1. **Test Deliverables**

* Test plan document
* Problem reports and corrective actions
* Test outputs
* Test cases

1. **Remaining Test Tasks**

Being a game, this test plan covers all of the needed testing for the system. The only additionally task would be to test the AI game function when it is completed. Addition a PVP function could be tested and added in a later release.

1. **Environmental Needs**

An environmental need is the data that will be processed within MasterMind will rely in the input the user is inputting to guess which colors are generated.

1. **Responsibilities**

Devin T., Rodolfo D., Zijun Z.:

Making final decision to determine if the system has met all necessary requirements

Selecting features to be tested and which are not to be tested.

Providing resolutions of any conflicts within the development process.

Devin T.:

Responsible for ensuring that all required elements are set to enable testing.

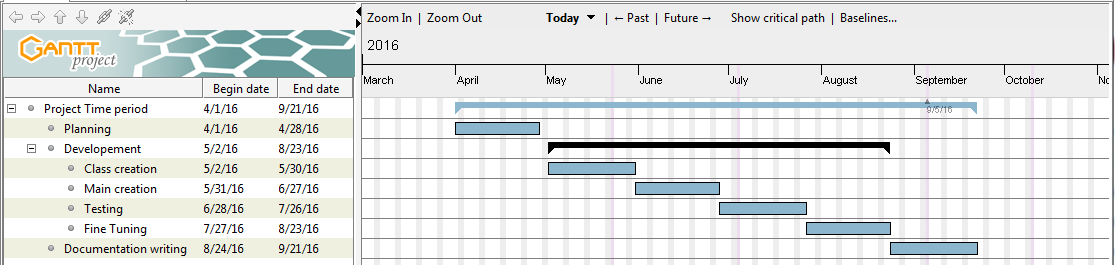
Rodolfo D. :

Responsible for detecting and setting risk that the system may encounter.

Zinjun Z.:

Responsible for setting the overall strategy for all testing phases.

1. **Schedule**

****

Here is a basic schedule for the system in addition to the testing time period.

1. **References**
   1. **Project17A\_1 document**
   2. **Mastermind System Requirements Specifications**