

Chapter 1

TEST PLAN (TP) for Game Rules and Play

Version 1.1, April 2014

TEST PLAN (TP)

FOR

Swords and Sorcery

Version 1
April 30, 2014

Prepared for:
Clinton Jeffery

Prepared by:
Tao Zhang, Cameron Simon, Matthew Brown,
Tyler Jaskowiak, Ian Westrope, ChiHsiang Wang
(Rule Team)
University of Idaho
Moscow, ID 83844-1010
CS383 TPD

RECORD OF CHANGES (Change History)

[illegible]

A - ADDED M - MODIFIED D - DELETED

Swords and Sorcery

TABLE OF CONTENTS	
Section	Page

1	TEST PLAN (TP) for Game Rules and Play	1
1	IDENTIFIER	1
2	REFERENCES	1
3	INTRODUCTION	2
4	TEST ITEMS	2
5	SOFTWARE RISK ISSUES	2
6	FEATURES TO BE TESTED	2
7	FEATURES NOT TO BE TESTED	2
8	APPROACH	3
9	ITEM PASS/FAIL CRITERIA	3
9.1	Unit Testing Pass/Fail Criteria	3
9.2	Intergration Testing Pass/Fail Criteria	3
9.2.1	Character	3
9.2.2	Spells	3
10	SUSPENSION CRITERIA	3
11	TEST DELIVERABLES	4
12	REMAINING TEST TASKS	4
13	ENVIRONMENTAL NEEDS	4
14	STAFFING AND TRAINING NEEDS	4
15	RESPONSIBILITIES	4
16	SCHEDULE	4
17	PLANNING RISKS AND CONTINGENCIES	4
18	APPROVALS	4
19	APPENDIX A. [insert name here]	1
20	APPENDIX B. [insert name here]	1

1 TEST PLAN IDENTIFIER

This test plan has no unique identifier other than Spring 2014 CS 383 sworsorc. This particular test plan is in conjunction to the rules and game play group test plans. There are many test plans for this group due to the size and span of the group's contents. This test plan, for example, will be known as the Units test plan.

2 REFERENCES

Use Cases and State diagrams are available at <https://github.com/cjeffery/sworsorc/tree/master/doc>

3 INTRODUCTION

The purpose of this test plan is to state the processes used by Game Rules and Play Team in testing the Spells and Characters for the Swords & Sorcery project. (Tao, Cameron)

This test plan covers the creation and implementation of the Moveableunit class and its sub class Armyunits. (Matt)

4 TEST ITEMS

The army units will be tested for the proper member variable values as well as proper results from the member function of the Moveableunit and Armyunit classes. Variables will be tested at creation and members variable will be tested for proper results when applicable. Some example tests will be that the location member variable is properly set during movement. Also another type of test that will be preformed is that conjured units are properly created and placed in the proper place on the game board.

- Class Interfaces
- Class Interactions
- Spells Implementation
- Character

5 SOFTWARE RISK ISSUES

- Improper casting of units
- Unit ID's interpreted incorrectly

6 FEATURES TO BE TESTED

- conjured unit appear when casted
- units are properly represented on HUD
- movement location is correct on map
- moral status is properly updated after combat
- Select character on GUI
- Select Spell
- Effects of Partial Spells
- Manna Costing

7 FEATURES NOT TO BE TESTED

What is not tested is the count of the units on the map that correspond to the scenario loaded. Also the type of unit is not test, this is assumed correct.

8 APPROACH

We will use JUnit tests as well as manual tests to determine the effectiveness and accuracy of the methods and variables tested.

9 ITEM PASS/FAIL CRITERIA

Items will receive a PASS once the variables/methods have been test multiple times and under different circumstances and the results of the test were within expected parameters. A FAIL will be received if the test results are not within expected parameters.

9.1 Unit Testing Pass/Fail Criteria

9.2 Intergration Testing Pass/Fail Criteria

9.2.1 Character

Rule Description	Test Description	Expected Result
Create new character object with its information.	createCharacter function is called with specific character name in Character-Maker.java.	A new character object is created and returned that contains all of the specified characters information.
Potential spells for selected character displayed.	Select character in GUI, then click cast spell button on sidebar panel.	List of spells that can be cast by selected character are displayed.

9.2.2 Spells

Rule Description	Test Description	Expected Result
Character with Power Level should have a spell book.	Generate a spell book for the character.	A frame with a list of spells should be shown on the screen.
Show spell description.	Click on the Spell button.	A frame will be displayed with all information about the spell.
A target need to be selected for most of the spells.	Click on cast button on the frame of displaying information about the spell. Then select a target by right click the target on the map.	A target is selected to cast the spell.

10 SUSPENSION CRITERIA AND RESUMPTION REQUIREMENTS

Tests will be suspended if a bigger piece of code that directly or indirectly effects the code is compromised. Tests will also be suspended once a FAIL has be received for that test. The tests will be resumed once the effected code fixed.

11 TEST DELIVERABLES

- Test plan document
- Test cases
- Relevant error logs or problem reports

12 REMAINING TEST TASKS

This is a multi-group project in which many test plans are created for the respective group. Also since the full project has not been completed there will be more tests developed in the future.

13 ENVIRONMENTAL NEEDS

There are no special requirements for this test plan.

14 STAFFING AND TRAINING NEEDS

Training in JUnit test is required to understand sections of the code.

15 RESPONSIBILITIES

This is a co-operative, no one person is in charge. Tests are divided by group type developers.

16 SCHEDULE

Realistically most tests will not be completed due to the rapid approach of the end of the semester.

17 PLANNING RISKS AND CONTINGENCIES

Too many tests will cause a halt in development and since the project won't be completed by the end of the semester might as well get as much working functionality as possible.

18 APPROVALS

Dr. J is the evil genius behind the project and is the only one capable of approving this process.

19 APPENDIX A. [insert name here]

Include copies of test examples, etc. supplied or derived from the customer. Appendices are labeled A, B, ... n. Reference each appendix as appropriate in the text of the document.

[insert appendix A here]

20 APPENDIX B. [insert name here]

[insert appendix B here]