The King of the Hot Dog

Software Testing Plan

Team members: Qin Shuang

Zhou Manman

Li Huixiang

Lin Qiang

Content

[1.Introduction 2](#_Toc7362680)

[1.1 Purpose of Writing 2](#_Toc7362681)

[1.2 Expected Reader 2](#_Toc7362682)

[1.3 Definition of Terms 2](#_Toc7362683)

[1.4 Test Target 3](#_Toc7362684)

[1.5 Reference Material 3](#_Toc7362685)

[2.Test Resource 3](#_Toc7362686)

[2.1 Testers 3](#_Toc7362687)

[2.2 Test Environment 3](#_Toc7362688)

[2.2.1 Hardware Environment 3](#_Toc7362689)

[2.2.2Software Environment 4](#_Toc7362690)

[2.2.3Safety environment requirements 4](#_Toc7362691)

[2.3Bug Management Tool 4](#_Toc7362692)

[3.Test Range 4](#_Toc7362693)

[4.Test Content 5](#_Toc7362694)

[4.1Start/Interrupt/Completion Criteria 5](#_Toc7362695)

[4.2 Test Type 5](#_Toc7362696)

[4.2.1 Unit Test 6](#_Toc7362697)

[4.2.2 Integration Testing 7](#_Toc7362698)

[4.2.3 Function Test 7](#_Toc7362699)

[4.2.4 UI Test 8](#_Toc7362700)

[4.2.5 Compatibility Test 9](#_Toc7362701)

[4.2.6 Security and Access Control Testing 10](#_Toc7362702)

[4.3 Testing Technology 11](#_Toc7362703)

[5.Schedule 11](#_Toc7362704)

[5.1 Test time progress 11](#_Toc7362705)

[5.2 Test Milestone 13](#_Toc7362706)

[6.Risk Statement 14](#_Toc7362707)

[6.1 Risk Assessment 14](#_Toc7362708)

[6.2 Risk Management 14](#_Toc7362709)

[6.3 Problem severity description 15](#_Toc7362710)

# 1.Introduction

## 1.1 Purpose of Writing

The purpose of this document is to develop a system test plan for President Wang to eat hot dog games. The main contents include test purpose, test scope, test content, test plan, personnel arrangement, risk analysis and test progress, etc., for subsequent test work. Provide a clearer process to ensure that testing is done efficiently.

## 1.2 Expected Reader

Developers, project managers, testers, project instructors.

## 1.3 Definition of Terms

* **Functional Testing:**  System-level testing of the system in accordance with the functional definitions in the system requirements definition
* **Non-functional testing:** System-level testing of the system in accordance with the non-functional definitions in the system requirements definition
* **Test Case: ：** A situation designed by testers to test a certain function of the software.
* **Black box test: ：** Black box test is also called function test. It is tested to check whether each function can be used normally. In the test, the program is regarded as a black box that cannot be opened. It is tested at the program interface without considering the internal structure and internal characteristics of the program. It only checks whether the program function is normally used according to the requirements specification. Whether the program can properly receive input data to produce correct output information. The black box test focuses on the external structure of the program, regardless of the internal logic structure, and is mainly tested for software interfaces and software functions. The black box test is based on the user's perspective and tests from the correspondence between the input data and the output data.
* **Integration Testing:** Integration testing is also called assembly testing or joint testing. Based on the unit test, all modules are assembled into subsystems or systems according to design requirements (such as according to the structure diagram) for integration testing.

## 1.4 Test Target

Test whether the various functional modules in the game can meet the requirements in the requirements analysis and test whether there are bugs. Expected to achieve rapid improvements and improvements in the system.

## 1.5 Reference Material

Introduction to software testing technology. By gu le shi jiu Lin/tsinghua university press

Software testing: second edition. By Paul C.Jorgensen

# 2.Test Resource

## 2.1 Testers

Qin Shuang、Lin Qiang、Li Huixiang、Zhou Manman

## 2.2 Test Environment

### 2.2.1 Hardware Environment

|  |  |
| --- | --- |
| Processor： | Dual-core processor and above |
| RAM： | More than 32MB |
| Hard Disk Space： | 1GB or more |

### 2.2.2Software Environment

Microsoft Windows 7

### 2.2.3Safety environment requirements

The security of the operating system, the security of the test tool, and the security of the test software.。

## 2.3Bug Management Tool

1.Trac

Trac is an application platform that integrates wiki and issue tracking management systems for software development projects. It is an open source software application. Trac is a project-oriented model based on a schedule-oriented model. The obvious feature is that it manages projects in a milestone manner. The specifics of each milestone are used to define, track, and so on.

2. Bugzilla

　 Bugzilla is an open source defect tracking system that manages the entire lifecycle of bug submission, repair, shutdown, etc. in software development.

# 3.Test Range

This document defines the main functions of the game. The characters of the game can move left and right, eat the hot dog to increase the satiety, eat the bomb to reduce the health and other functions and the test of the project operation.

a. This test plan is a detailed description of the entire project test work in the project's Elaboration Phase, which will be the basis for all future test work.

b. The test of the game project includes the following types of test activities:

* unit test ;
* Integration Testing ;
* function test ;
* UI test ;
* Compatibility test ;
* Security and access control testing.

# 4.Test Content

First, use the software to determine whether the software's function meets the user's requirements, refer to the requirements specification, and then use the system test to combine it with computer hardware, peripherals, some supporting software, data and personnel and other system elements. In the actual operating environment, a series of tests on the computer system, and the test personnel simulate the user to test.

## 4.1Start/Interrupt/Completion Criteria

|  |  |
| --- | --- |
| Start/Interrupt/Complete Test | Standard Description |
| Start Testing Standard | The hardware environment is available and the software is properly installed. |
| Interrupt Test Standard | The installation failed to complete properly or the program's documentation had quite a few errors or system service exceptions or found Block Bug |
| Complete Test Criteria | Complete the test plan in the test plan and meet the program and test quality objectives, and be confirmed by Test Lead/R&D Manager |

## 4.2 Test Type

|  |  |  |
| --- | --- | --- |
| Test Type | Whether to Adopt | Description |
| Unit Test | Yes | A small piece of code written by the developer to detect if a small, well-defined function is correct |
| Integration Testing | Yes | Integration testing identifies problems that occur when combining units. |
| Function Test | Yes | Depending on the system requirements, check that the product is correctly implemented by the developer. |
| UI Test | Yes | Test user interface (such as menus, dialogs,windows,and other configurable controls) whether the layout, style meets customer requirements, text is correct, the page is beautiful, the text and image combination is perfect, and the operation is friendly. |
| Compatibility Test | Yes | Verify that the data and operation of the system are correct under various network connections, and whether it is compatible with various devices. If there is cross-system support, it is necessary to check whether the various behaviors are consistent under each system. |
| Security and Access Control Testing | Yes | From the local storage of data to the security features of data transmission, processing and remote access, the system's input validity check, authentication,authorization, sensitive data storage, data encryption, detection system user authorization level, data leakage, illegal Authorized access. |

### 4.2.1 Unit Test

Unit testing is a small piece of code written by a developer to detect if a small, well-defined function is correct.

In general, a unit test is used to determine the behavior of a particular function under a particular condition (or scene).

|  |  |
| --- | --- |
| Test Target | Prove that the behavior of a piece of code is exactly the same as the developer's expected effect. |
| Method | Test module for unit testing by unittest.main()  The first step in the argument is its own module \_\_main\_\_;  The second step extracts the test methods from all the test classes in the module and generates a test suite.  Finally, pass the test suite to testrunner for specific testing. |
| Completion Criteria | * All required unit tests were completed in accordance with the unit test plan. * The requirements for coverage as specified in the unit test in the test plan are met. * The software unit functions consistently with the design. * The errors found in the unit tests have been modified and the defect repair rates at all levels have reached the standard. |
| Special Matters to Consider | No |

### 4.2.2 Integration Testing

|  |  |
| --- | --- |
| Test Target | On the basis of unit, all modules are assembled into subsystems or systems according to design requirements for integration testing |
| Method | Test the integrated functionality using the black box test method. And regression testing of previous integrations. |
| Completion Criteria | * All the problems encountered in the test have been recorded * All test cases are running * 90% of test cases have been successfully passed * All test cases have been run at least three times and all errors have been modified * The test results have been recorded and the test analysis report has been submitted to the project manager for inspection. |
| Special Matters to Consider | No |

### 4.2.3 Function Test

Functional testing is a functional validation test of test requirements. The goal of these tests is to confirm that the use case or use case scenario can be successfully implemented. This type of test generally uses a black box test method that interacts with the application through a graphical user interface (GUI) and analyzes the output to confirm the correctness of the application. Verification of each function of the product, according to the functional test cases, item by item test, to check whether the product meets the functions required by the user.

|  |  |
| --- | --- |
| Test Target | **Numerical function:**  Whether the score of the game character is increased after touching the hot dog  Whether the game character has reduced the health after touching the bomb  Whether the time in the time-limited mode is normal or not  Can the normal conversion to the health value when the satiety reaches the highest value in the infinite mode? |
| **Control function:**  Whether the keyboard can control the movement of the game characters left and right |
| **Mode selection function:**  Whether the player enters the game mode normally after making a selection |
| **Input function:**  Can the user enter the game name into the game normally? |
| **Interface display module:**  All pictures can be displayed normally |
| **Sound function:**  Whether the sound button normally switches music |
| Method | Design functional test cases using black box testing techniques, develop test procedures for each test case, and record and edit scripts for regression testing. Execute test cases to verify each use case, use case scenario, and use case flow. Mainly verify the following:   * Get the expected results when using valid data. * Displays an appropriate error or warning when using invalid data. |
| Completion Criteria | * The functional test case design has been reviewed. * Functional testing was completed in accordance with the functional test plan. * The coverage requirements specified in the functional test plan for functional testing have been met. * The system achieves the functions and performance defined by the detailed design. * The errors found in the system test have been modified, and the defect repair rate at all levels has reached the standard. * The defects found have all been resolved. |
| Special Matters to Consider | No |

### 4.2.4 UI Test

User interaction with the software is verified through user interface (UI) testing. The goal of UI testing is to ensure that the user interface provides the user with the appropriate access and functionality to navigate through the test object functionality. In addition to this, UI testing also ensures that objects within the UI functionality meet the expected requirements and follow industry standards.。

|  |  |
| --- | --- |
| Test Target | * Trace Debugging correctly reflects functionality and requirements by browsing test objects, including browsing between windows and windows, between modules and modules, and the use of various access methods (Tab, mouse, and shortcuts). * The objects and characteristics of the window (for example: settings, size, position, health status, and score indicators) are all compliant. |
| Method | Create or modify tests for each window to verify that individual application windows and objects are properly viewed and in a normal object state. |
| Completion Criteria | Make sure each window is consistent with the baseline or meets acceptable standards. |
| Special Matters to Consider | Not all features of custom or third-party objects are accessible. |

### 4.2.5 Compatibility Test

Compatibility testing refers to tests that can be tested on a specific hardware platform, between different application software, on different operating system platforms, and in different networks.

|  |  |
| --- | --- |
| Test Target | Ensure that the project is operating properly in the specified hardware environment, on different operating system platforms, and in different network environments. |
| Method | First of all, manual testing is carried out. The test engineer uses the main browser and the common operating system to test the main process and the main interface to see if there is a problem in the main process and the main interface. If there is a problem, record the bug, and the browser model and version. And the operating system, accurately locate the cause of the bug, submit a bug, and inform the developer to modify.  This is followed by third-party testing tools such as IEtester, SuperPreview, and Browsershots: browsershots.org. With third-party testing tools, you can test IE compatibility and compatibility with major browsers, including Google, 360 browser, Sogou browser, QQ browser, etc., find the location of the bug, analyze the test results, and inform the programmer to adjust. |
| Completion Criteria | * The planned test can cover the use case event flow (the elementary stream and all candidate streams) and the planned tests are all executed. * The defects found have all been resolved. |
| Special Matters to Consider | No |

### 4.2.6 Security and Access Control Testing

Security and access control testing focus on two key aspects of security:

* Application-level security, including access to data or business functions;
* System level security, including login and run access to the game.

Application-level security ensures that users can only access specific features or use cases in the expected security situation, in other words, users can only access limited data. For example, you might be allowed to enter data and create a new account, but only the background has the right to delete or modify those data or accounts. If there is data-level security, the test ensures that User Type One can see all user information (including financial data), while User Type Two can only see the same user's information data.

System-level security ensures that only users with system access can access the application and can only access it through the appropriate gateway. Users who have registered to log in can enjoy permanent access to the system.

|  |  |
| --- | --- |
| Test Target | Make sure that the software does not perform functions that are not pre-designed, and all expected hazards have been eliminated or mitigated. |
| Method | Application level security:  Identify and list each user type and the features or data that it is authorized to access.  Create tests for each user type and verify their access rights by creating transactions specific to each user, such as a login account.  Modify the user type and rerun the test for the same user. For each user type, be sure to provide or deny these additional features or data correctly.  System level security:  After the user registers to log in to the system, the program is run using the function test script, and all functions are running normally. |
| Completion Criteria | Complete the planned test. |
| Special Matters to Consider | The access control of the system must be checked and discussed with the system administrator. Since this test may be a system administrator's function, you may not need to perform this test. |

## 4.3 Testing Technology

|  |  |  |
| --- | --- | --- |
| Testing Technology | Whether to Adopt | Description |
| Milestone technology | Yes | Milestone achievement criteria and acceptance methods are developed after testing |
| Write test cases | Yes | Write test cases during the product coding phase |
| Black box test | Yes | Run in a simulator or actual device for testing |
| Integration Testing | Yes | It is detected whether the integrated system of the module meets the requirements, whether the processing of the business process and the data flow conforms to the standard, whether the system has logic rigorous errors in the processing of the service flow, and whether there are unreasonable standards and requirements. |
| Confirmation test | Yes | Before the system is released, confirm the basic requirements against the feature list to confirm that the product is functioning correctly. |

# 5.Schedule

## 5.1 Test time progress

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test phase | Test Task | Work Estimate | Staff assignments | Start Time | End time |
| First stage  Unit Testing | Object collision detection function, display update function, lifeboard class, scoreboard class, Bomb class, Hotdog class, Settings class, global properties, character health,  Check\_bottom (groups),  Check\_collide (player, groups), update\_screen (ai\_settings, screen, stats, sb,life, player, hotdogs, bombs,  play\_button), etc | 7days | Lin Qiang | 2019.4.18 | 2019.4.26 |
| Unit test bug review |
| second stage  Integration Testing | Control module, display module, numerical module, input module and other modules within the module and module compatibility | 5days | Lin Qiang | 2019.4.26 | 2019.5.1 |
| The third phase  function test | Detecting the system's numerical function, mode selection function, control function, input function and sound function can meet user requirements | 3days | Li Huixiang、  Zhou Manman | 2019.5.1 | 2019.5.3 |
| Fourth stage  UI test | It is true that each object is properly displayed and the objects and features of the window (e.g., settings, size, position, operating status, and score indicators) are met. | 2days | Qin Shuang | 2019.5.1 | 2019.5.2 |
| Fifth stage  Review bug | Audit bugs other than unit tests | 2days | Lin Qiang | 2019.5.3 | 2019.5.4 |
| Sixth stage  Compatibility test | The detection system can operate normally in different application software, on different operating system platforms and in different network environments. | 2days | Zhou Manman | 2019.5.3 | 2019.5.4 |
| Seventh stage  Security and access control testing | Make sure that the software does not perform functions that are not pre-designed, and all expected hazards have been eliminated or mitigated. | 2days | Qin Shuang | 2019.5.3 | 2019.5.4 |
| Eighth stage  Acceptance Test | Imitate the test of the user's use process | 2days | Li Huixiang | 2019.5.4 | 2019.5.5 |
| Test summary | Test summary and analysis, problem feedback | 2days | Qin Shuang、Lin Qiang、Li Huixiang、Zhou Manman | 2019.5.7 | 2019.5.9 |

* according to the actual situation to make certain adjustments

## 5.2 Test Milestone

|  |  |  |  |
| --- | --- | --- | --- |
| Test phase | Starting time | Complete time | Stage completion sign |
| Develop a test plan | 2019.4.16 | 2019 .4 .18 | Finished |
| Design test case | 2019.4.18 | 2019.4.20 | Finished |
| Test environment preparation | 2019.4.18 | 2019.4.20 | Finished |
| Test implementation | 2019.4.18 | 2019.5.5 | ongoing |
| testing report | 2019.5.6 | 2019.5.7 | Unfinished |
| Test summary | 2019.5.7 | 2019.5.9 | Unfinished |

# 6.Risk Statement

## 6.1 Risk Assessment

|  |  |
| --- | --- |
| Potential Problem | Description |
| System design | Lack of rationality |
| Database risk | Unreasonable database structure design |
| Improper control of database operations leads to data inconsistency |
| Improper  response time | The time is too long, so the user is frustrated; |
| Too short a time causes misoperation; |
| Lack of test environment | Possible test defects |
| System robustness check | An operation exception occurred |
| cyber security | Lack of stability |

## 6.2 Risk Management

L=Low (risk and processing priority is low) M=Middle (risk and processing priority is medium) H=High (risk and processing priority is high)

|  |  |  |
| --- | --- | --- |
|  | Functional testing phase | Document test |
| Correctness | H | H |
| File integrity | H | H |
| Continuity of processing | M | M |
| Access control | M | M |
| Compliance | H | H |
| reliability | H | H |

## 6.3 Problem severity description

|  |  |
| --- | --- |
| Problem severity | Description |
| Fatal defect | 1. Illegal exit due to crash caused by the program  2. Infinite loop  3. Database deadlock  4. Program interruption due to incorrect operation  5. Major function loss or serious function error  6. Connection error with the database |
| Serious flaws | 1. Program error  2. Program interface error  3. Constraints such as tables, business rules, default values, and integrity of the database |
| General defect | 1. Operation interface error (including the definition of column names in the data window, whether the meaning is consistent)  2. Simple input restrictions are not placed in the foreground for control  3. Delete action did not give a prompt  4. There are too many empty fields in the database table |