2011

Ercan Özdemir
Hasan Faruk Çoban
İsmail Ceylan

[TEST SPECIFICATION REPORT]

MasterMind
Contents

1. INTRODUCTION .................................................................4
   1.1. Goals and Objectives .................................................4
   1.1.1. User Environment Test ........................................4
   1.1.2. Developer Environment Test ................................4
   1.2. Statement of the Scope .............................................4
   1.3. Major Constraints ..................................................5
   1.4. Definitions, Acronyms and Abbreviations ...................6
   1.5. References ..........................................................6

2. TEST PLAN .....................................................................6
   2.1. Software to be tested ...............................................6
   2.2. Testing strategy ......................................................7
       2.2.1. Unit Testing ..................................................7
       2.2.2. Integration Testing ..........................................9
       2.2.3. Validation Testing .........................................9
       2.2.4. High-order Testing .........................................10
   2.3. Test metrics .........................................................10
       2.3.1. Functionality .................................................10
       2.3.2. Performance ................................................10
       2.3.3. Reliability ...................................................10
       2.3.4. Validity ......................................................10
       2.3.5. User-friendly ...............................................11
   2.4. Testing Tools and Environments ................................11

3. TEST PROCEDURE .......................................................12
   3.1. Unit Test Cases ....................................................12
       3.1.1. Emotiv .......................................................12
       3.1.2. Games .......................................................12
       3.1.3. Database ..................................................13
   3.2. Integration Testing ................................................13
   3.3. Validation Testing ................................................13
   3.4. High-order Testing (a.k.a. System Testing) .............14

4. TESTING RESOURCES AND STAFFING .......................14

5. TEST WORK PRODUCTS .............................................14

6. TEST RECORD KEEPING AND TEST LOG ....................14
7. ORGANIZATION AND RESPONSIBILITIES .................................................................15
8. TEST SCHEDULE ....................................................................................................15
1. INTRODUCTION

Project MasterMind is a BCI projects and due dependencies of BCID device (Emotiv) it has to implement incrementally. For this reason, Team MasterMind collaborates while developing. Any new developed features are added project after it is ensured about its correctness. Other than that, any modules that can be done independently are done division by work. These independent developed modules are tested by its developer then inserted the main project. However, testing or debugging processes will continue until final release.

1.1. Goals and Objectives

Our testing strategy contains two different phase:

1.1.1. User Environment Test
Since MasterMind project mainly targets people who suffer explicit attention disorder one of main goals of testing process is to ensure providing its users a user-friendly environment.

1.1.2. Developer Environment Test
Like any serious software product MasterMind project must satisfy some quality criteria such as:

- Being error-free
- High performance
- Logically correctness
- Compatibility between modules

1.2. Statement of the Scope

This document is prepared for description of the testing process of MasterMind project. Although testing process lasts from beginning till ending, this document focus on full-scale
testing process not small-time tests which were done during implementation of modules. At this point, it is important to express that every step of our test plan is decided by specific properties of MasterMind Project and the major constraints that we may have encountered (i.e. Lack of time). We tried to make suitable plan for testing our product. This report also defines our criteria and approaches for encountered problems.

1.3. Major Constraints

TIME: MasterMind project is aimed for helping sufferers of explicit attention disorder. Since it is a real time application, it is intended to implement fast interaction between user, Emotiv Device and game. Serious run-time calculations (i.e. FFT of brainwaves) are made during interactions. Interpreting waves and affecting game play are also important tasks of our product. Hence, it must be ensured speed of software should be as high as possible. Our testing process mostly focuses on speeding up application. Beside that due to time planning we have nearly a month to finish testing phase. As a result of this time is the greatest constraint for us to face.

DATA: In order to get accurate data pattern of Brainwave interpretation, our software should be tested on as many user as possible. However, collecting this amount of data costs serious time and human resource. Our aim in this case is try to keep optimum balance between data collected and spent time.

SPEED: As it mentioned before speed of the program is our greatest concern. It is aimed to provide user with product works smoothly. Which features are going to integrate on our project are rely on base speed of program. In other words, if any feature causes too much burden on program it may have been sacrificed to maintain acceptable performance. Speed is our biggest concern in this project.

HARDWARE: The architecture of computers used in beta testing should have enough capacity to run the Emotiv device. Memory usage, performance of the graphics card is the other factors limiting tests of the project.
1.4. Definitions, Acronyms and Abbreviations

**EEG**: Electroencephalography

**GUI**: Graphical User Interface

**BCID**: Brain-computer interface device

**BCI**: Brain-computer interconnection

1.5. References

- Emotiv EPOC SDK user manual
- Test Specification Template

2. TEST PLAN

To begin with, in this document whole testing process of MasterMind is described. In other words, in general terms past, current and future actions about testing strategy of MasterMind is mentioned in this document. Two general strategies are agreed by team MasterMind. First, if module or section is developed individually due to division by work, responsibility of testing components is up to developer. On the other hand, in system testing these parts are checked by all team members. Second, if team collaborates to implementation testing process is done cooperatively.

2.1. Software to be tested

MasterMind project with all its sub-components will be tested in order to provide users with error-free, user-friendly software product. MasterMind focus on attention increase via its neuro-feedback features. It has three sub-components which are Emotiv, Game and
Database. All components and their integration and finally all system will be tested in testing phase.

- Emotiv Module
- Database Module
- Pong Game
- Dots and Boxes Game
- Text reader Application
- User Interface Application
- After-game Graph Application

2.2. Testing strategy

MasterMind team is developing software which focuses on brain-computer interconnection. It will be tested bottom-up approach. Testing phase begins with smaller parts. After this part is finished components which are made of already tested smaller parts are tested then main components and finally all system are going to be tested. This approach is accepted because our project mostly developed incrementally.

2.2.1. Unit Testing

MasterMind project has three main components as it mentioned before. For each unit there are functionality tests which are being made. Emotiv and Game components’ unit tests have already been done. Only database component are not finished but its testing strategies are decided. Following subsections have more details on unit testing:

2.2.1.1. Emotiv
Since it is hardware –dependent module it is decided that black-box method would be appropriate. Only outputs are checked in this unit. Emotiv is responsible for collecting
brainwaves and transforms them into usable data. Procedures will be described later, but list of units’ tests is given:

- Emotiv signal (from brain) collector
- Emotiv signal processing
- Emotiv EEG data controller
- Sending data to Games
- Sending data to Database

#### 2.2.1.2. Games
For testing games there are two main subjects: User-friendliness and functionality. In MasterMind project games are merely phase of collection of data. While user playing game his/her brainwaves, reflexes and responses are gathering. Hence game phase must draw attention of user; user-friendliness; and performs correctly; functionality; to avoid any distractions for user. Because success rate of software depends on correctly gathered data. For this reason, game component must work error-free. Testing procedures will be discussed in “Testing” procedures section. Following list is depicts for unit tests of games:

- Interconnection with Emotiv
- Interconnection with Database
- Physical engines
- Performance of AI sub-components

#### 2.2.1.3. Database
To provide Experts who are going to evaluate our result collected data should be saved. Database component are responsible for this task. Every Brainwaves and game result which includes responses of users are stored in database. Data should store accurately in order to provide experts reliable subjects for healthy evaluations. This module mostly concern about experts. However, it has another task to do; database also keeps user personal information and records. To prevent abuses of information database should secure and reliable. In other
words, securing unauthorized data accesses are also important point of our testing strategy. Procedures of testing will be discussed later but for now list of unit test is shown below:

- Interconnection with Emotiv
- Interconnection with Games
- Checking accuracy of received data from Emotiv
- Checking accuracy of received data from Games
- Testing for login security
- Testing for permissions of accesses for each user and role

2.2.2. Integration Testing

Since communications between modules are essential for system, integration testing is also essential while integrating all modules to system. As soon as, a new sub-component or a new object is defined developed an integration test is applied to check correctness of communication between modules. In our project order of integration is Emotiv, games and database. The reason of this order is Emotiv is the key module of our product it is the first module implemented. Other modules are integrated to it. In other words, Emotiv is the foundation of our project. Because of that every integration test begins with Emotiv integration. Database is last because to store data first data must be produced.

2.2.3. Validation Testing

Validation testing is the testing phase that is enabling compatibility between implementation and the other previous reports i.e. Software Requirement Analysis; Initial and Detailed Design Reports. Validation testing is done while developing. Development of any module are guided by previous reports especially design reports. Features which are going to be implemented are decided due to promised proposition. Its details will be covered Procedures section.
2.2.4. High-order Testing

After finishing the development phase it is planned that high-level tests for our product.

2.2.4.1. Performance Tests
Each modules speed and reliability will be tested. Especially, Game module will be tested in different levels of load.

2.2.4.2. Stress Test
This tests concern of game and database modules. In a different runtime lengths it as checked that if any crashes or data loss happens.

2.2.4.3. User Tests
Because our products targeted market is health clinic we will planning to run some beta tests to understand what user wants from our product. As result of these tests, user-friendly identity of our product will be improved.

2.3. Test metrics

A description of all test metrics to be used during the testing activity is noted here.

2.3.1. Functionality
This metric stands for if components are able to perform its required task.

2.3.2. Performance
Performance metric expressed if components run with acceptable speed.

2.3.3. Reliability
Concerns data transfer. To satisfy this metric component must provide dependable data transfer with other components which is interconnected.

2.3.4. Validity
This metric is to check if components accomplish propositions from previous design reports.
2.3.5. User-friendly
The hardest metric to test is user-friendliness because these tests are highly subjective and hard to evaluate.

2.4. Testing Tools and Environments

Testing tools are important for the success of testing phase and naturally the success of product. In our unit testing phase NetBeans, Emotiv EPOC SDK test bench and MySQL Graphical User interface are used. Following paragraphs have detailed information about testing tools.

NetBeans debugger features is used for every module development. By tracing and correcting bugs and errors our system’s stability is increased. Also it has great role on testing Game components. Since our codes are written in NetBeans environment it is easy to keep tracking any mistakes especially in unit testing phase.

Emotiv EPOC SDK test bench enable us checking validity of acquired data from Emotiv device. In every phase of testing it helps keep tracking condition of Emotiv device, situation of Brainwaves and values of EEG data. To sum up, Emotiv EPOC SDK is testing environment for hardware.

MySQL GUI provides us easy access to database. In any tests which includes database module MySQL GUI is essential part of testing process. To checking both brainwaves and user records it presents easy-to-use environment.

3. TEST PROCEDURE

In this section of document testing procedures and tactics will be explained. Testing approaches and test cases will be expressed.
### 3.1. Unit Test Cases

MasterMind Project has three main components as it mentioned before. For each component there are serious units tests are or will be done. Due to time constraint each feature or preferences cannot be tested. Our approach focuses on functionality. Especially data transfer between modules.

#### 3.1.1. Emotiv

Unit tests on Emotiv targeted on reliable connection and brainwave data transfer.

- Unit constructors
- Wave class constructors
- Fast Fourier transformations methods reliability
- Raw data storage
- Processed data storage and transfer
- Connection constructors(to other modules)

For Emotiv module test cases are signal quality testing, signal length testing etc.

#### 3.1.2. Games

Unit tests on Games targeted on high performance.

- Unit constructors
- Visual units constructors
- Data acquire and send methods reliability
- Physical and logical correctness
- AI performance

For Game module test cases are AI response testing, performance testing etc.
3.1.3. Database
Unit tests on Database targeted on security and reliable data storage

- Unit constructors
- Database initialization
- Permissions
- Parsing methods reliability
- Data storage and acquire reliability

For Database module test cases are: Security testing, recovery testing etc.

3.2. Integration Testing

After unit test, integration of components need additional test because of compatibility. To overcome this problem we choose incremental integration. We insert each newly developed module in order to main project. When any module integrated to system combined test cases of new module and older modules which already integrated are applied to system. After satisfying results are achieved another module will be integrated. In other words, we run small time high-order unit test each time in integration testing.

3.3. Validation Testing

Since developing any new module and features implementation is shaped from propositions from design reports, hence validation tests begin during development process. After integration of module with presence of all team member last general revisions are made for validation of related module. These revisions are done just after integration tests of related module. By this approach, we ensure validity of our product each step.
3.4. **High-order Testing (a.k.a. System Testing)**

High order testing procedures are still in planning phase because Project has not finished yet. On the other hand, partial system tests are constantly ongoing in order to protect unity and integrity of project. It is safe to say that our system tests primarily targeted on increasing performance, stress management and user-friendliness features.

4. **TESTING RESOURCES AND STAFFING**

As can be inferred from above, testing is done by the developers in a collective manner. Then error-free parts are going to be integrated to the whole system. Finally, system tests are done by all development team members.

We are also going to test the system with the end-users. We will take some children which are known to suffer from attention deficit disorder between the ages 10-12 and allow them to play the games. We will make more precise adjustments to the wave calibration (this maps to the levelization in game) according to the result.

5. **TEST WORK PRODUCTS**

We did not produce any test work products because there are already functional products that contributes to testing.

6. **TEST RECORD KEEPING AND TEST LOG**

For debugging purposes we plan to use NetBeans debugger which is a plug-in for the NetBeans IDE. The debugger includes necessary user interfaces, therefore we do not need any extra record keeping and logging. PHPMyAdmin will be used to check correctness of the
data kept in the database. It is intended to handle the administration of MySQL and allows users to view tables and their records in a database, modify or delete tables, databases. These two tools will help to make our tests and log them in chronological order and evaluate them at the end with pre-determined metrics.

7. ORGANIZATION AND RESPONSIBILITES

Altough the total work will be done collaboratively we have some distinct responsibilities. Database component and the integration of the database to our game environment will be checked by Ercan Özdemir. İsmail Ceylan will be responsible from the stabilities of the games and the effects of neuro-feedback mechanism. Faruk Çoban will be responsible to combine our results and preparing possible bugs in our integrated design. Finally all of our team members are required to attend to the final tests to check the system fully and in all parts.

8. TEST SCHEDULE

<table>
<thead>
<tr>
<th>Test Description</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Specification Report:</td>
<td>24/04/10</td>
</tr>
<tr>
<td>Unit and Integration Tests:</td>
<td>05/05/10</td>
</tr>
<tr>
<td>Validation Tests:</td>
<td>15/05/10 – 25/05/10</td>
</tr>
<tr>
<td>Performance and Stress Tests:</td>
<td>20/05/10 – 25/05/10</td>
</tr>
<tr>
<td>Beta and Alpha Tests:</td>
<td>25/05/10 – 30/05/10</td>
</tr>
<tr>
<td>Tracing and Correction:</td>
<td>30/05/10</td>
</tr>
</tbody>
</table>