

ECE1778 Final Report

Cognitive Game

Jiaxing Li 999169670

Tong Zou 1001913815

(word count: 2095)

Table of Content

Introduction	2
Statement of functionality	2
Overall Design	7
Reflection	9
Individual Contribution.....	10
Future Work.....	11
Announcement.....	11

Introduction

Psychiatrists usually use paper-based cognitive ability tests to measure the cognitive impairment for depression patients. Based on the measurements, psychiatrists are able to determine whether depression treatment will be successful or not. Cognitive ability is the most effective indicator for depression treatment prediction. Another indicator is the patients' motivations on taking risks.

However, problems occur that paper tests take several hours and patients might lose concentration, which greatly affect the results accuracy. Moreover, it is also hard to measure patients' motivations using paper tests. Therefore, our Role Playing Game (RPG) Cognitive Game is built to act as the depression treatment predictor. There are two main reasons to build this mobile application. The first reason is by making a background story and embedding the memory tests into game challenges, we can fit patients into an interesting but intense story which makes them concentrate and actively participate into the memory tests. And the second reason is to measure patients' motivations of taking risks, through their playing behaviours.

Statement of functionality

There are four main functionalities in Cognitive Game – game Interface, animation, memory tests, scoring system and result display.

a. Game interface

Figure 1 Game Interface and Figure 2 are the main game interfaces of Cognitive Game. There are several elements and functionalities contains in the game interface.

- Main character animation

We implemented walking animations for the main character (the girl with blonde hair in the screenshot), which is controlled by player.

- Gamepad

The bottom left gamepad is used to control the movement of the main character

- Non Player Character (NPC)

The characters built in the game which are not controlled by any of the users are called Non Player Character, and they are used to provide interactions between game and users.

We build two NPCs in our game. One has the “click me” notation in the screenshot to

give users instructions, and another is designed to lead player to result page once game is finished.



Figure 1 Game Interface (1)

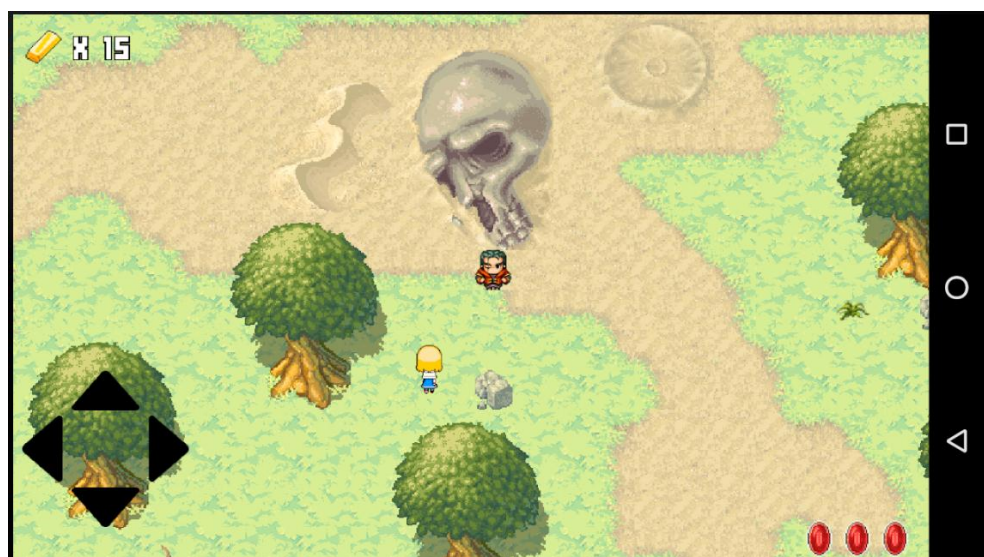


Figure 2 Game Interface (2)

- Background story and game instruction

The background story and game instruction pages are triggered by clicking the NPC when the user first enters the game.

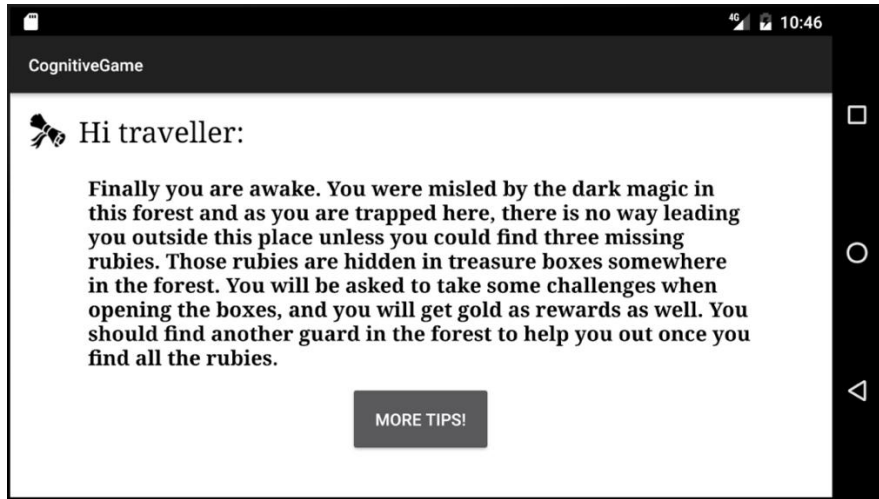


Figure 3 Game Instruction (1)

The first instruction tells the background story of the game and explains the tasks for the users – players need to find three rubies by answering questions and take rubies to a guard to leave the magic forest.

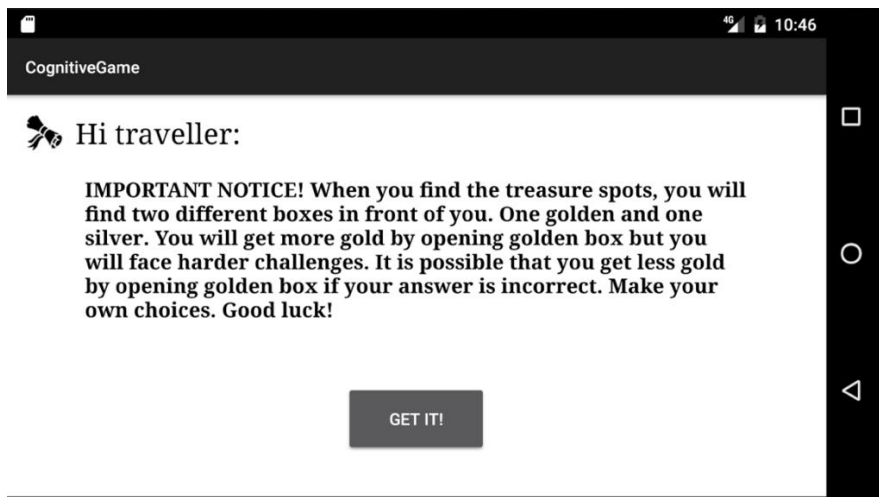


Figure 4 Game Instruction (2)

The second instruction describes that for each treasure box with ruby, they can choose either gold or silver box. Challenges in the gold boxes are harder but have more reward.

- Question difficulty selection

When approaching the treasure boxes, players will be automatically led to a question difficulty level selection page, the description and instruction of this page are included in the instruction page. The detailed mechanism of the scoring system will be described in section c.



Figure 5 Question Difficulty Selection

b. Cognitive Ability Test Questions

There are three main question sets used in the game, for learning and memorizing ability measurement purpose.

- **N-back task**

The N-back task is a continuous performance task that is commonly used as an assessment in cognitive neuroscience to measure a part of working memory and working memory capacity. The player is presented with a sequence of number which is shown one after another, and the task consists of indicating when the current number matches the one from n steps earlier in the sequence. The load factor can be adjusted to make the task more or less difficult. In Cognitive Game, both correctness and time consumption are measured and recorded. The load factor was set to 2 as easy mode and 3 as hard mode.

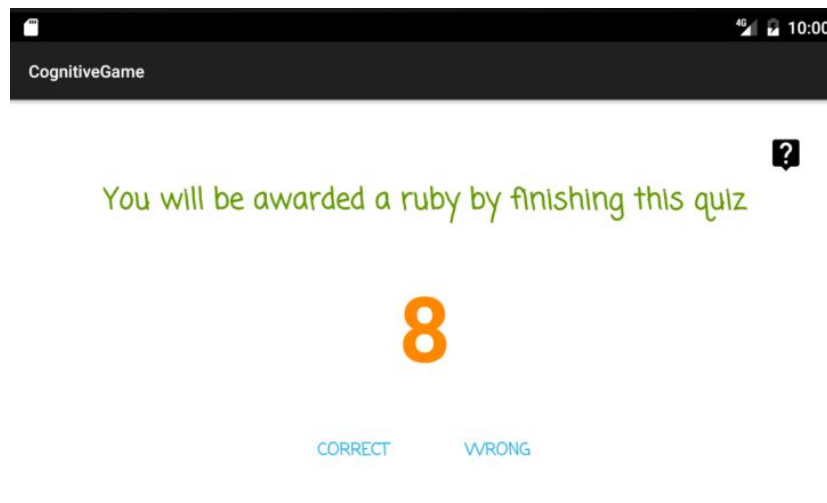


Figure 6 N-back task

- Digital Symbol Substitution Test (DSST)

DSST is a neuropsychological test sensitive to brain damage and depression. It is commonly used on short term memory and learning ability measurement. DSST consists of ten digit-symbol pairs followed by a list of digits. Under each digit the player should select the corresponding symbol as fast as possible. The number of correct symbols selected within the allowed time is measured and recorded.



Figure 7 Digital Symbol Substitution Test

- Visual Recognition Memory Test

Visual recognition memory is the ability to recognize previously encountered events. The player is presented a list of randomly generated icons, and the task is to identify which list is in the same sequence as the previous icon lists on another screen, among three choices. The easy mode list consists of four icons and hard mode consists of five.

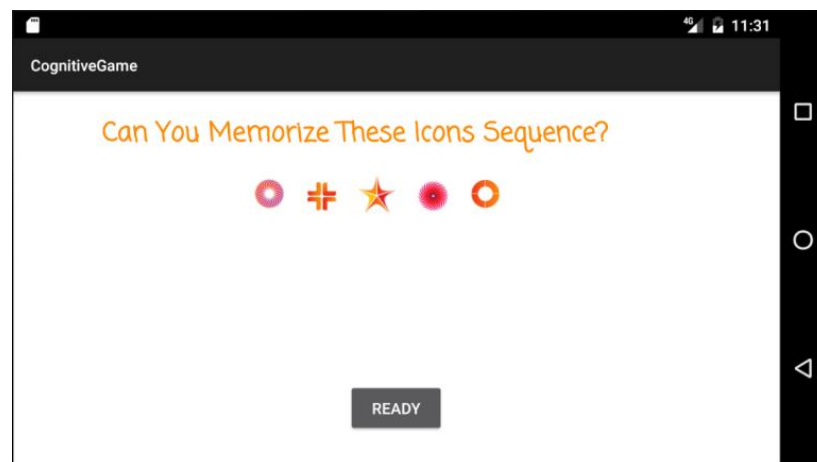


Figure 8 Visual Recognition Test (1)

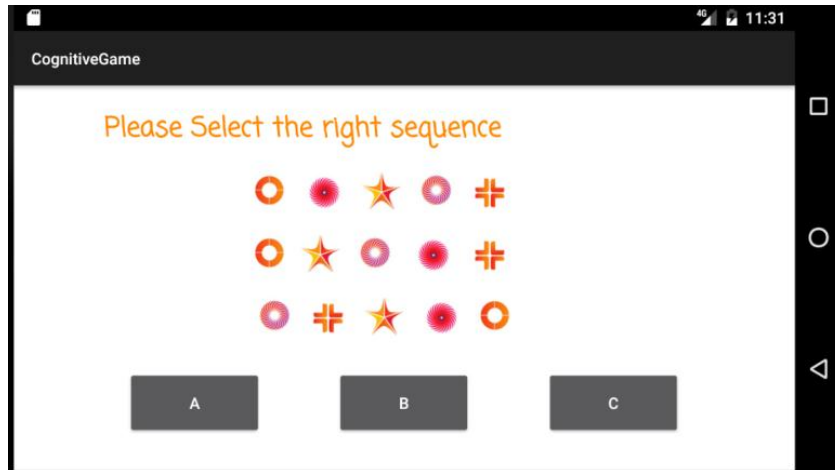


Figure 9 Visual Recognition Test (2)

c. Scoring system

It can be seen in the main game interface screenshot. There is a gold icon on the top left corner and several ruby icons on the bottom right corner. The scoring system consists of these two elements. The rubies are used to finish the game and three of them are required. Users will get a ruby regardless of type of boxes they choose and accuracy of their test result. This is because the main goal of this application is to lead the users through all the memory tests and collect the results. Ensuring players to finish all the tests is the main concern. Therefore, once users finish the required amount of tests, we should no longer keep them in trap.

In terms of gold, if users take harder challenges and accuracies are more than 50%, they will get 10 gold for each test. If they fail to reach the threshold, they will get 0 gold. However, if they take easy challenges, they will get 5 gold regardless of correctness. This is the risk we set for users, get more gold or get punished. This mechanism is not quantitatively described in instruction and will be found by user along the way, which could make them put more thoughts into their choice and facilitate the measurement of motivation.

Overall Design

As shown in Figure 10, the block diagram clearly presents the software structure and relations among each functioning part in the game.

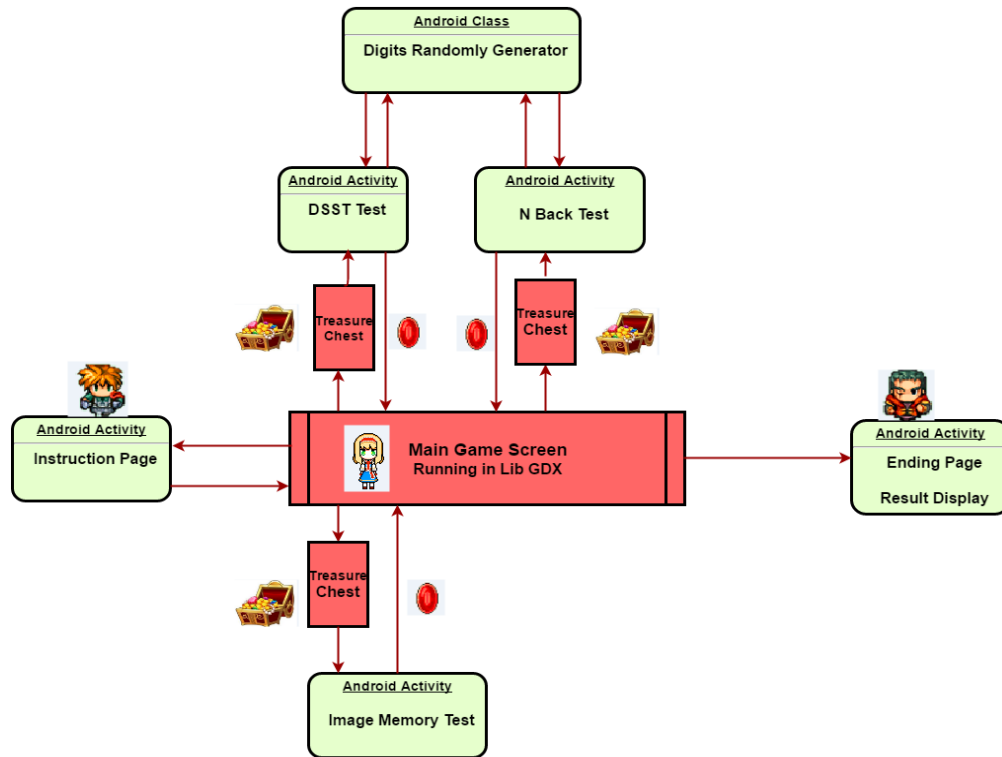


Figure 10 Cognitive Game Block Diagram

- **Android**

All the red parts are built in LibGDX and green parts are built in Android. Android activities will be activated when the player approaches a treasure chest or in the ending page when the adventure is finished.

Question sets are built in various Activities and Fragments. Each of the activity and fragment has its own layout design. A Random Number Generator class is built as utility class, since both DSST and N-Back task require randomly generated numbers as test contents, and the Visual Recognition Memory Test needs random numbers as icon indexes as well. All the player's question results include correctness, time consumption and difficulty level, and all these data are saved locally, for the purpose of result check when adventure is finished. When the quiz is finished, corresponding Android activity and fragment are destroyed and software is back to LibGDX.

- **LibGDX**

All the game user interfaces and elements are built in LibGDX.

For the animation part, the first rendered object is the map. Map in LibGDX has a special format and is rendered layer by layer. The map is pre-drawn in a stand-alone software called Tiled with 50 x 80 grids and each grid has 48 x 48 pixels. The grassland will be rendered first, then the flowers and trees and finally the NPCs. After map is rendered, a player is generated on top of the map. There are 12 pictures for a player - one state for each direction and three pictures for each state. The movement direction is tracked and pictures switch in sequence according to the direction at the rate of 60 frames per second. This realizes the moving animations of the character. Gamepad is implemented by image click buttons and controls the state and direction of the character. Collision detection is used to prevent player from bumping into roots of trees, rocks and other landscapes. For the static display part, rubies, gold and treasure box display are built into a special object layer in the map. After answering a question, the object layer is called and all the positions of the static elements are retrieved and updated. The display of the box selection page is also implemented by LibGDX using image buttons and bitmap fonts.

Reflection

- **What we learned**

The most important thing we learned is teamwork and multidiscipline collaborating skills. This project gives us the chance to work with the specialist from another discipline, he brings the knowledge from his specialized area and we help him to realize the idea. We learned how to work effectively with team members from different discipline.

Good communication and presentation skills are also essential during this process. The project progress needs to be updated at a high intensive pace, not only to specialist but also to all the class mates and professor. The team members' communication and presentation skills are greatly improved though this three-month project experience.

On technical field, team members' developing skill on object oriented language is greatly enhanced, and testing and debugging skills are also get improved.

- **What we will do differently**

More interesting story line: the background story line could be designed differently to make the game be more tightly connected with memory test questions. Such as when

finished collecting three rubies, the player can be transferred to map and continue more challenge.

More Effective Project Management: although we followed the agile processes during the project and got all the work done on time, we still make some mistakes on time scheduling and task management. We did not start early research on game framework and did not see the big picture of the application, especially ignored the importance of the integration between game part and memory tests, which made us start over and built everything from scratch again in the middle of the project and wasted a huge amount of time. We will figure out the overall picture and focus more on game part at the very beginning if do the project again.

More user friendly game interface: we will try another method to improve the game user interface if doing the project again. Current map rendering method is fine but still has some tiny issues. The new method can build the character into map rendering, thus placing the character between layers on the map. This will make the character move beneath trees, clouds and other big objects and enhance the experience of the game. The collision detection algorithm is based on grids on the map now, which is rough detection. It can be improved by splitting obstacle grid into more cells to boost detection precision. Finally, DSST display can be animated by highlighting the current match pair and feeding in unmatched pattern in a streamline-like feeling.

Individual Contribution

Jiaxing is in charge of all the Android work. The DSST, N-Back task and Visual Recognition Memory Test design and implementation, time recording and correctness calculation in quiz, Random Number Generator algorithm design and implementation, and the final result check.

Tong mainly focused on the LibGDX game framework implementation which includes game user interface design, character moving and animations, map rendering, character collision detection, game instruction and treasure box selection animations.

Team members worked together on necessary knowledge researches on depression treatment, overall story line and game layout design, data transition between Android and LibGDX, as well as the final game testing and debugging.

Future Work

- One function can be added to save playing history and test result statistics on server. It would be ready for doctor's remote access. Right now the test result data is only stored locally.
- Memory test results analysis function can also be embedded into Cognitive Game. This functionality can generate a more advanced conclusion regarding to the cognitive ability measurement based on NAB test marking scheme, rather than only the correctness and time consumption information.
- Make the background story line more integrated, from a higher level of view. A complete story line that makes more sense to players.
- After a more integrated story is designed, more game levels and maps can be designed to embed various types of memory test. This will make the game more interesting and more test results will be available for analysis.

Announcement

- Video of final presentation (OK to post)
- Report (OK to post)
- Source code(OK to post)