**Date: 10/22/2014**

Midterm Report

**Nine Men's Morris**

**CS 471/571 Software Engineering**

**Advisor: Prof. Dianxiang Xu**

**Member: Milson Munakami, Jimmy Wang, Sung-Ju Fan-Chiang**

Team Project Midterm Report

Section I Team Organization and Buddy Rating .................................................... 3

Section II. Summary of User Stories ....................................................................... 4

Section III. Summary of Test-Driven Development and Refactoring ...................... 8

Section IV. Summary of Pair Development ........................................................... 10

Section V. Lessons Learned ................................................................................... 11

Section I Team Organization and Buddy Rating (1%)

(1) Member's Work and Tasks

|  |  |
| --- | --- |
| Member | Work and Tasks |
| Milson Munakami | Team leader  Game GUI design and programming  Testing  Scrum master  Main programmer for pair programming |
| Jimmy Wang | Coordinator for the project progress  Testing  Backend logic and algorithm  Main programmer for pair programming |
| Sung-Ju Fan-Chiang | Documentation  Testing  Minor programmer for pair programming |

(2) Buddy Ratings

|  |  |  |  |
| --- | --- | --- | --- |
| Member | Buddy | Rating | Note |
| Milson Munakami | Jimmy Wang | 1 |  |
| Sung-Ju Fan-Chiang | 1 |  |
| Jimmy Wang | Milson Munakami | 1 |  |
| Sung-Ju Fan-Chiang | 1 |  |
| Sung-Ju Fan-Chiang | Milson Munakami | 1 |  |
| Jimmy Wang | 1 |  |

Section II. Summary of User Stories (12%)

(1) User Stories and Development Tasks

|  |  |  |
| --- | --- | --- |
| |  | | --- | | Description of user story (author) | | Tasks for Each User Story |
| 1. As a program scrum master, we want to apply software development model so that the project is well organized and to follow the development step by step. (Milson) | 1. Consider different software development models and try to apply Agile |
| 2.Implementing the Test-Driven Development method to produce the project |
| 3.Class refactoring to improve the program |
| 2. As a player, we want to either play with computer or another player so that we can still have a game without another participant. (Milson) | 4.Add options on game board for either choice |
| 5.Produce program to implant AI and let computer react with gamer |
| 6.Players can move or place the men on not occupied (vacant ) nodes only |
| 7.Fix the grid with valid twenty-four intersections or points to move the Men |
| 3. As a programmer, we want to use an OOP language which every member can participate so that team member can contribute to this pair development project. (Milson) | 8.Evaluate the OOP language and have internal discuss for the possibility |
| 9.According to team member specialty, we choice Java language |
| 10.Make a schedule for team member to do pair development and code review |
| 11.Showing the game status and notification on text area |
| 4. As a player, we want to know current status so that the information can remind us for next process. (Sung-Ju) | 12.Create an information area to show text message |
| 13.Code the displayed status by each movement and show the message according to the movement |
| 5. As a game player, we want to have reasonable reaction from computer to play with so that we have certain enjoyment with the game. (Sung-Ju) | 14.Study the rules and practice the game to contribute algorithm for the game |
| 6. As the programmer, we need to implant the rules so that the phase I game could follow the correct placement and remove node. (Jimmy) | 15.Deal with different strategic method and priority to product individual case to conquer the different movement from player |
| 16. Patterns for forming a mill - a straight line, vertically or horizontally |
| 7. As the programmer, we need to implant the rules so that the phase II game could follow the correct movement, remove node, flying node, and decide the winner. (Jimmy) | 17.Removing the opponent player’s Men after making mill |
| 18.Decision based on the moves left on board |
| 19.To check if there is any legal move left or not |
| 20.Players can do “flying” after only having 3 players on board |
| 22.Winning and losing conditions for a player |

(2) Team Meetings and Meeting Minutes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Meeting # | | Time/place | Participants | Topics and Decisions |
| 1 | | 9/3  13:00 - 13:30  MEC410 | Milson  Jimmy  Sung-Ju | Team organization and study the project requirement and game rules |
| 2 | | 9/10  13:00 - 13:40  MEC410 | Milson  Jimmy  Sung-Ju | Identify the high-level scope  Looking for coding standard  Searching for useful information and resource on internet |
| 3 | | 9/12  13:00 - 13:30  MEC410 | Milson  Jimmy  Sung-Ju | Consider user story to start project  Think about Agile method and try to apply  Try to list to do items for project schedule |
| 4 | | 9/17  13:00 - 13:30  MEC410 | Milson  Jimmy  Sung-Ju | Identify initial “requirements stack”  List to do list  - Figure out basic structure of the program  - GUI and backend game details  - To draw game board and plan functions |
| 5 | | 9/19  13:00 - 13:30  MEC410 | Milson  Jimmy | Consider user story to add possible class  Setup pair programming schedule  Identify and architectural vision |
| 6 | | 9/24  13:00 - 13:30  MEC410 | Milson  Jimmy  Sung-Ju | Finalize the Classes  - Point  - Player  - Node  - Board |
| 7 | | 9/26  13:20 - 13:50  MEC410 | Milson  Jimmy  Sung-Ju | Add game board GUI  Graphics i.e. Images for the Men’s, board, and texture area |
| 8 | | 10/1  13:00 - 13:40  MEC410 | Milson  Jimmy  Sung-Ju | Implant the TDD by TFD and refactoring  Review the pair programming schedule |
| 9 | | 10/3  13:00 - 13:30  MEC410 | Milson  Jimmy  Sung-Ju | Review more TDD and refactoring |
| 10 | | 10/8  13:00 - 13:30  MEC410 | Milson  Jimmy  Sung-Ju | Add Drag and Drop functionality  Add the board points where Men’s can be dropped |
| 11 | 10/10  13:00 - 13:30  MEC410 | | Milson  Jimmy  Sung-Ju | 1st draft for the document  Discuss and review refactoring |
| 12 | 10/15  13:00 - 13:30  MEC410 | | Milson  Jimmy  Sung-Ju | Try to integrate the back end design and the board to apply the game rules |
| 13 | 10/17  13:00 - 13:30  MEC410 | | Milson  Jimmy  Sung-Ju | Final draft for project midterm document  Produce the presentation slides |

Section III. Summary of Test-Driven Development and Refactoring (12%)

(1) Descriptions of the Tests

|  |  |  |  |
| --- | --- | --- | --- |
| Number | Description of test case (test input and oracle) | User story # and Task # | Developers |
| 1 | **testRightNeighbor**  input: a valid point  output: the right neighbor (if exists) | #2  #6, 7 | Milson |
| 2 | **testLeftNeighbor**  input: a valid point  output: the left neighbor (if exists) | #2  #6,7 | Milson |
| 3 | **testUpNeighbor**  input: a valid point  output: the Up neighbor (if exists) | #2  #6,7 | Milson |
| 4 | **testDownNeighbor**  input: a valid point  output: the down neighbor (if exists) | #2  #6, 7 | Milson |
| 5 | **showAllAdjacentPoints**  input: a valid point  output: a set of adjacent points | #4  #13 | Sung-Ju, Milson |
| 6 | **testShowPlayer**  input: N/A  output: Display the status of a player. | #2  #6 | Milson |
| 7 | **testPlaceAMan**  input: a vacant point, player  output: place a point on the board, player’s status updated. | #7  #18 | Jimmy |
| 8 | **testMoveAMan**  Input: a men on the board, a vacant valid point, player  output: move a man on the board, player’s status updated. | #7  #17 | Jimmy |
| 9 | **testRemoveAMan**  input: a man on the board, player  Output: remove a man from the board, player’s status updated. | #7  #17 | Jimmy |
| 10 | **testAllowToFly**  input: player  output: Boolean(if the player is allowed to fly.) | #7  #20 | Jimmy |
| 11 | **testHasMills**  input: player  output: Boolean(if the player has mills) | #4  #13 | Sung-Ju |
| 12 | **GameLogicTestSuite** (console)  Input: points or set of points (for different actions performed in the game)  Output: updated game status and helpful information for debugging and playing. | #3  #11, 12, 13 | Sung-Ju, Milson |

(2) Descriptions of Refactoring

|  |  |  |
| --- | --- | --- |
| Refactoring # | Description of the refactoring (problem and solution) | Developers |
| 1 | Remove redundant code  → less code is good | Milson  Jimmy  Sung-Ju |
| 2 | Object oriented approach With Classes and subclasses | Milson  Jimmy |
| 3 | Coverage testing and remove uncovered code | Milson  Jimmy  Sung-Ju |
| 4 | Renaming the method names, variable names, class names, fields  → to correct better method name without side effects | Milson  Jimmy |
| 5 | Split Variable Declaration  → when you initialize a variable at the declaration but later on find out that the initialization needs to be in a try- or if-block | Milson  Jimmy  Sung-Ju |
| 6 | Extract Method  → create a method out of some code without any side effects | Milson  Jimmy |
| 7 | Change Method Signature  → change the parameters, visibility, and return type of a method including default values for new parameters | Milson  Jimmy  Sung-Ju |
| 8 | Extract Interface/Super-class  → Extract an interface or a super-class out of the Main class to a separate java file | Milson  Jimmy  Sung-Ju |
| 9 | Correct Indentation  → to keep the code readable | Milson  Jimmy  Sung-Ju |
| 10 | Pull Up/Push Down Pull methods and variables to a generic interface or super-class or push it down to a subclass | Milson  Jimmy  Sung-Ju |

Section IV. Summary of Pair Development (8%)

(1) Pair Development

|  |  |  |  |
| --- | --- | --- | --- |
| Session # | Time duration/place | Participants | Tasks |
| 1 | 25 hours/ MEC410 | Jimmy, Milson,  Sung-Ju | To implement the valid and legal moves of the Men and Testing and bug fixing all possible cases |
| 2 | 20 hour/ MEC410 | Jimmy, Sung-Ju | Possible ways to form a mill  And remove the opponent player’s Men after making mill |
| 3 | 15 hours/ MEC410 | Milson, Jimmy | Moving and flying a Men on board  Drag and drop of Men to the allowed nodes on board |
| 4 | 10 hours/ MEC410 | Sung-Ju, Milson | Testing the winning and losing condition of any player |

Section V. Lessons Learned

Member: Milson Munakami

(1) What did you personally gain from the project?

I learned about the agile methodology to do any project. I tried to keep up with the developer and keep track of the requirements and execution. I tried to implement Object Oriented Programming System (OOPS) to do this project. I tried to implement the Test Driven Development (TDD) and it is easier to code once we understand the possible pass and fail boundary conditions.

(2) What does your program do well, and what could your program do better?

I have tried to implement drag and drop function to the game using Java Swing Framework. I have included an excellent GUI which is user friendly and easier to use and learn to play. But I wish to make it more animation and use more graphics so it can be more responsive and user can interact with the game easily. Some of the Testing frameworks and other IDE tools are new to me so I learn them with in this process.

(3) How could you improve your development process if you develop a similar game from scratch?

I would use the TDD methodology and will follow the software life cycle models to execute the task using more Agile methodology. I will use the regular standup meeting to discuss with team members regarding the achievements and trouble they are facing daily.

I will track the sprints and check the initial user stories to check if we are going on right track to achieve the final goal. The development process can be increased and improved if we can plan more detail specifications in advance. To assure the quality and less bugs in the program, I will allocated significant amount of time for testing i.e. integration and functional of the application to achieve more performance and good quality.

Member: Jimmy Wang

(1) What did you personally gain from the project?

(2) What does your program do well, and what could your program do better?

(3) How could you improve your development process if you develop a similar game from scratch?

Member: Sung-Ju Fan Chiang

(1) What did you personally gain from the project?

I have learned to start a project as the course materials which I have never done before, for instance, to do it by user story, organize the project like scrum method, try to do it by pair programming, and Test-Driven Development (TFD + refactoring). It is a great opportunity to apply what we have learned for software engineering and have hands-on experience. When I was a programmer many years ago, we divided project by different blocks. The system analysis manager had organized some blocks, and programmers just did their parts. The old way was all controlled by manager and may not work well compare to what I have learned from this project.

(2) What does your program do well, and what could your program do better?

The project can take advantage by the specialty of individual team member. The program coding could be done efficiently by the strong member if he/she spend more time on coding. The person have more experience about organization could coordinate each other and share the program development experience to others. Each team member may focus on what his/her strength and to benefit the project.

The program could do better if it followed the schedule and the loading wouldn't be pushed in the end. It is natural and very common to delay schedule even the project could be done in time. However, the project quality probably could be better if it have been done by ordinary schedule.

(3) How could you improve your development process if you develop a similar game from scratch?

I will overlook all project stuffs to have an idea about the requirement, members, and schedule. The project schedule and organization still can be modified during the program development if it is needed. Besides, there are some resource from internet, the team may discover and to share all possible and useful information in order to deal with potential problem.