**Team Project Sprint #2**

Instructions

KEY:

Sprint 1

Sprint 2

Sprint 3

Please read the instructions carefully. All members of your team should discuss the instructions together to ensure that everyone is on the same page.

**Objectives**

1. Update and complete the user stories and acceptance criteria of the target software that allows a human player to play against either a human or a computer opponent.
2. Implement all the user stories for a human player to play complete Mill games against a human opponent (including all improvements on the previous sprint).
3. Conduct a full retrospective meeting (refer to the lecture notes) and report the meeting minutes.

**Deliverables and Grading Policy**

1. Project Report **(28 points)**

The project report should include the following sections. Please use the attached template.

* 1. Updated complete user stories using the template discussed in class. **(1 points)**

Provide a complete list of user stories and estimated efforts for the target software that allows a human player to play against either a human or a computer opponent.

* 1. Updated complete acceptance criteria using the template discussed in class. **(8 points)**

Provide complete acceptance criteria for all the user stories.

* 1. Implementation tasks **(17 points)**

Describe the production code, automated test code or manual test cases for all the user stories for a human player to play complete Mill games against a human opponent. For each acceptance criterion of every user story, you need to implement at least one test (either test code or manual test case). Some automated tests using xUnit or a similar tool are required.

* 1. Minutes of ALL meetings, including, but not limited to: project/sprint planning meeting, stand-up meeting, backlog grooming, retrospective meeting, and pair programming (or development) session. **(2 points)**
  2. A table of buddy ratings. Individual members may email their buddy ratings to the instructor or teaching assistant.

Each team only needs to submit one report. For an individual member to receive the credit for this part of the project, the team’s project report must include explicit evidence of his/her contribution (e.g., his/her name is listed as a developer).

1. Demonstration **(5 points)**

Submit a 5-minute video, clearly demonstrating that:

1. your project has implemented the working software for a human player to play complete Mill games against a human opponent.
2. For each acceptance criterion of an implemented user story, your project has implemented either an automated test method or performed an acceptance test manually.
3. your project has some unique features or enhancements (optional).

Grading of the demonstration is based on completion of the required functions (**2 points**), and overall presentation (**3 points**) using the following evaluation rubric:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Poor | Fair | Good | Very Good | Excellent |
| Was the demonstration logically organized |  |  |  |  |  |
| Were points made clearly and concisely |  |  |  |  |  |
| Were the grader or instructor’s questions answered satisfactorily |  |  |  |  |  |

3. Source Code

Submit all source code. Make sure your project report is consistent with the source code.

**Team Project Sprint #2**

Report Template

Team Name: Stealth Tech Corp

Team Members: Ami, Omer, Shawn, Thom, Tim

1. **Updated User Stories**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **User Story Name** | **User Story Description** | **Priority** | **Estimated effort (hours)** | **Actual effort (if completed)** | **Status (completed, toDo, inProgress)** | **Developer names** |
| 1 | Start Game | As: a player, when I open the game, on my display,  I want: to start a game with an empty board and 9 pieces to place on the board,  To: be ready to play. | A | BkEnd: 10  GUI: 15 | BkEnd: 8  GUI: 12 | Complete | ALL |
| 2 | Placing Phase | As: a player, after the game has started, on the game board  I want: to be able to place one of my unplaced pieces on an empty spot on the board,  To: play the ‘placing’ phase of the game. | A | BkEnd: 5  GUI: 5 | BkEnd: 4  GUI: 6 | Complete | ALL |
| 3 | Moving Phase | As: a player, after the ‘placing’ phase, on the game board  I want: to be able to move my pieces  To: play the ‘moving’ phase of the game | B | BkEnd: 8  GUI: 8 | BkEnd: 8...  GUI: 4 | Complete | ALL |
| 4 | Have Mill | As: a player, after each placement/move, on the game board  I want: to know when I have completed a mill  To: enable to me to know that I can remove an opponent’s piece | B | BkEnd: 4 | BkEnd: 4 | Complete | Thom, Tim, Omer |
| 5 | Take Piece | As: a player, after forming a mill, on the game board  I want: to know what pieces I can remove  To: reduce how many pieces/moves my opponent has | B | BkEnd: 2  GUI: 2 | BkEnd: 2  GUI: 2 | Complete | ALL |
| 6 | Flying Phase | As a player, at the beginning of each turn, on the game board  I want: to know if I can fly  To: know that I am in the ‘flying’ phase of the game | D | BkEnd: 1 | BkEnd: 1 | Complete | Thom, Omer. Tim |
| 7 | Flying Move | As a player, when I am flying, on the game board  I want: move any one of my pieces to any one spot that is available  To: try not to lose the game | D | BkEnd: 2 | BkEnd: 2 | Complete | Thom, Omer, Tim |
| 8 | Win or Lose | As: a player, after each mill (or turn), on the game board  I want: to know when I win  To: be a BOSS | C | BkEnd: 4  Gui: 2 | BkEnd: 4  GUI: 2 | Complete | ALL |
| 9 | Menu | As: a player, when the application starts, on the screen  I want: to select menu options for the type of game to play  To: select a human/computer opponent | C | Gui: 3 | Gui: 3 | Complete | Ami |
| 10 | Status Box | As: a player, while playing, on the game board  I want: to see a status box that tells me what to do next  To: know the status of the game | C | Gui: 3 | Gui :3 | Complete | Shawn |
| 11 | AI Opponent | As: a Player, when I choose the option to have an AI as my opponent.  I want: The AI knows the rule of the Game.  To: Play a game with an AI same as play against a Human opponent. | A | All:30 | All:N/A | Todo | ALL |
| 12 | AI Behavior-1  (Placing/checking empty spaces) | As: An AI, when I place the piece.  I want: The piece goes in an available space  To: set up the piece on the board. | A | All: 30 | All: N/A | Todo | ALL |
| 13 | AI Behavior-2(Moving phase) | As: An AI, I need to know what to do next if all pieces are placed.  I want: Move my pieces on the board to legal spots (realizing the player’s pieces).  To: Play the game while AI’s Bag is empty. And the board is full. | A | All:30 | All:N/A | Todo | ALL |
| 14 | AI Behavior-3  (Mill recognition) | As: an AI, after each placement/move, on the game board  I want: to know when I have completed a mill  To: enable to me to know that I can remove an opponent’s piece | A | All:30 | All:N/A | Todo | ALL |
| 15 | AI Behavior-4  (Removing player’s pieces legally) | As: an AI, after forming a mill, on the game board  I want: to know what pieces I can remove  To: reduce how many pieces/moves my opponent has | A | All:30 | All:N/A | Todo | ALL |
| 16 | AI Behavior-5  (Auto pilot mode) | As an AI, at the beginning of each turn, on the game board  I want: to know if I can fly  To: know that I am in the ‘flying’ phase of the game | A | All:30 | All:N/A | Todo | ALL |
| 17 | AI Behavior-6  (Fly ) | As an AI, when I am flying, on the game board  I want: move any one of my pieces to any one spot that is available  To: try not to lose the game | A | All:30 | All:N/A | Todo | ALL |
| 18 | AI Behavior-7  (Win condition evaluation) | As: AI, after each mill (or turn), on the game board  I want: to know when I win  To: be a DOMINATOR. | A | All:30 | All:N/A | Todo | ALL |

1. **Updated Acceptance Criteria (AC)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User Story ID and Name** | **AC**  **ID** | **Description of Acceptance Criterion** | **Status (completed, toDo, inProgress)** | **Developer Names** |
| 1 Start Game | 1.1 | Given: a new game,  When: the application starts,  Then: create an empty Nine Men’s Morris board for 2 players, and 9 pieces for each player and, it is player 1’s turn  Note: Game is ready to play | Completed | All members |
|  | 1.2 | Given: a NMM board,  When: a spot is referenced that is not a playable point,  Then: the spot is invalid, and do nothing | Completed | Ami |
|  | 1.3 | Given: a new game  When: a location is clicked on the board that is a playable spot on the board  Then: the spot is valid, the click should be handled, and it should be passed to the back end to decide if a piece should be placed | Completed | Ami |
| 2 Placing Phase | 2.1 | Given: that the game has been initialized OR an ongoing game, and it is player 1’s turn, and there are still pieces to place,  When: player 1 makes a LEGAL placement,  Then: player 1’s piece is placed on the spot and the turn is changed to player 2.  Note: Game is now/still ongoing | Completed | All members |
|  | 2.2 | Given: an ongoing game, and it is player 2’s turn, and there are still pieces to place,  When: player 2 makes a LEGAL placement,  Then: player 2’s piece is placed on the spot, and the turn is changed to player 1.  Note: Game is still ongoing | Completed | All members |
|  | 2.3 | Given: an ongoing game, and it is player 1’s turn, and there are still pieces to place  When: player 1 makes an ILLEGAL placement,  Then: player 1’s piece is not placed, and the turn remains player 1’s (and a message is shown to the user)  Note: Game is still ongoing | Completed | All members |
|  | 2,4 | Given: an ongoing game and it is player 2’s turn, and there are still pieces to place,  When: player 2 makes an ILLEGAL placement,  Then: player 2’s piece is not placed, and the turn remains player 2’s (and a message is shown to the user)  Note: Game is still ongoing | Completed | All members |
| 3 Moving Phase | 3.1 | Given: an ongoing game, and after all pieces have been placed, and it is player 1’s turn,  When: player 1 makes a LEGAL move,  Then: move player 1’s piece, and change the turn to player 2.  Note: Game is still ongoing | Completed | All members |
|  | 3.2 | Given: an ongoing game, and after all pieces have been placed, and it is player 2’s turn,  When: player 2 makes a LEGAL move,  Then: move player 2’s piece, and change the turn to player 1.  Note: Game is still ongoing | Completed | All members |
|  | 3.3 | Given: an ongoing game, and after all pieces have been placed, and it is a player’s turn,  When: the player clicks on the other player’s piece  Then: do not pick up the player’s piece and do not allow the player to move the piece that does not belong to them.  Note: Game is still ongoing | Complete |  |
|  | 3.4 | Given: an ongoing game, and after all pieces have been placed, and it is a player’s turn, and they have clicked on a movable piece  When: the player clicks on an unavailable space (not adjacent or occupied)  Then: do not move the player’s piece, and the turn remains to the player  Note: Game is still ongoing | Complete |  |
|  | 3.5 | Given: an ongoing game, and after all pieces have been placed, and it is a player’s turn,  When: a player clicks on one of their movable pieces,  Then: highlight the locations the piece can move | Complete |  |
|  | 3.6 | Given: an ongoing game, and after all pieces have been placed, and it is a player’s turn, and a player has clicked on one of their movable pieces  When: a player clicks on the same piece again  Then: remove the highlights and cancel the move | Complete |  |
| 4 Have Mill | 4.1 | Given: an ongoing game, and it is player 1’s turn,  When: directly after player 1 has made a legal placement/move,  Then: check if the latest piece placed/moved by player 1 completed any mill(s).  Note: Game is still ongoing. | Complete | All members |
|  | 4.2 | Given: an ongoing game, and it is player 2’s turn,  When: directly after player 2 has made a legal placement/move,  Then: check if the latest piece placed/moved by player 2 completed any mill(s).  Note: Game is still ongoing. | Complete | All members |
| 5 Take Piece | 5.1 | Given: an ongoing game, and player 1 just placed or moved a piece,  When: player 1 makes a mill and if player 2 has any non-mill pieces available,  Then: player 1 can remove any one of those non-mill pieces of player 2. | Complete | All members |
|  | 5.2 | Given: an ongoing game, and player 2 just placed or moved a piece,  When: player 2 makes a mill and if player 1 has any non-mill pieces available,  Then: player 2 can remove any one of those non-mill pieces of player 1. | Complete | All members |
|  | 5.3 | Given: an ongoing game, and player 1 just placed or moved a piece,  When: player 1 makes a mill and if player 2 does not have any non-mill pieces available, then player 1 can remove any of player 2’s pieces from the board. | Complete | All members |
|  | 5.4 | Given: an ongoing game, and player 2 just placed or moved a piece,  When: player 2 makes a mill and if player 1 does not have any non-mill pieces available,  Then: player 2 can remove any of player 1’s pieces from the board. | Complete | All members |
| 6 Flying Phase | 6.1 | Given: an ongoing game, and player 1’s turn,  When: it is the beginning of player 1’s turn,  Then: count player 1’s pieces, and display if player 1 is flying | Completed | Thom/Tim/Omer |
|  | 6.2 | Given: an ongoing game, and player 2’s turn,  When: it is the beginning of player 2’s turn,  Then: count player 2’s pieces, and display if player 2 is flying | Completed | Thom/Tim/Omer |
| 7 Flying Move | 7.1 | Given: an ongoing game, and player 1’s turn, and player 1 is flying  When: player 1 makes a flying move,  Then: move player 1’s piece and change the turn to player 2 | Completed | Thom/Tim/Omer |
|  | 7.2 | Given: an ongoing game, and player 2’s turn, and player 2 is flying  When: player 2 makes a flying move,  Then: move player 2’s piece and change the turn to player 1 | Completed | Thom/Tim/Omer |
|  | 7.3 | Given: an ongoing game and a player that is flying  When: the player clicks on one of their own pieces  Then: Highlight available spaces the piece can be moved | Completed |  |
| 8 Win or Lose | 8.1 | Given: an ongoing game, and after player 1 makes a mill (or finishes their turn),  When: player 2 has available valid moves, and more than 2 pieces,  Then: the game continues, and it is player 2’s turn,  Note: The game is ongoing | Complete | All members |
|  | 8.2 | Given: an ongoing game, and after player 2 makes a mill (or finishes their turn),  When: player 1 has available valid moves, and more than 1 pieces,  Then: the game continues, and it is player 1’s turn,  Note: The game is ongoing | Complete | All members |
|  | 8.3 | Given: an ongoing game, and after player 1 makes a mill (or finishes their turn),  When: player 2 has no available legal moves,  Then: the game is over, and player 1 wins  Note: The game is finished | Complete | All members |
|  | 8.4 | Given: an ongoing game, and after player 2 makes a mill (or finishes their turn),  When: player 1 has no available legal moves,  Then: the game is over, and player 2 wins  Note: The game is finished | Complete | All members |
|  | 8.5 | Given: an ongoing game and after player 1 has made a mill (or finishes their turn),  When: player 2 has only 2 pieces left,  Then: the game is over, and player 1 wins  Note: The game is finished | Complete | All members |
|  | 8.6 | Given: an ongoing game and after player 2 has made a mill (or finishes their turn),  When: player 1 has only 2 pieces left,  Then: the game is over, and player 2 wins | Complete | All members |
|  | 8.7 | Given: an ongoing game, and after each player finishes their turn,  When: the MAX moves has been reached,  Then: the game is over, and the game is a draw  Note: The game is finished | Complete | All members |
| 9 Menu | 9.1 | Given: a menu, after initialization but before beginning play  When: player 1 selects a computer opponent,  Then: the game will set player 2 to AI  Note: The game is now ready to play with human vs AI | Ongoing | All members |
|  | 9.2 | Given: a menu at startup  When: A player clicks “how to play”  Then: Take the player to an instructions page | Ongoing | GUI team |
|  | 9.3 | Given: a user is on the main menu  When: a player clicks “start game”  Then: the game should be started according to the settings on the menu | Complete | GUI team |
|  | 9.4 | Given: a player is on the instructions page  When: a player clicks “start game”  Then: the game should be started according to the settings on the menu | Complete | GUI team |
| 10 Status Text Box | 10.1 | Given: any point during the game  When: anything happens  Then: update the status box to reflect the state of the game | Complete | Shawn |
|  |  | All primitive functions for the below User Stories/AC are finished. We listed them as “Todo” because we have not linked them to our AI. |  |  |
| 11 AI Opponent | 11.1 | Given: a new game before placing pieces.  When: player ready to place the first piece and place it.  Then: Ai will know how to respond to the player’s move.  Note: the game is now ongoing between the player and the AI. | ToDo | ALL |
| 12 AI Behavior-1  (Placing/checking empty spaces) | 12.1 | Given: a player piece is placed but an AI piece is not placed yet.  When: AI will start placing the piece.  Then: The player and AI will go back and forth placing pieces.  Note: Game is now ongoing until all pieces are placed or a mill is formed.  Note: if the mill is formed, the story is now moved on to another one. | Todo | ALL |
| 13 AI Behavior-2(Moving phase) | 13.1 | Given: an ongoing game, and after all pieces have been placed, and it is player 1’s turn,  When: player 1 makes a LEGAL move,  Then: move player 1’s piece, and change the turn to AI.  Note: Game is still ongoing | ToDo | ALL |
|  | 13.2 | Given: an ongoing game, and after all pieces have been placed, and it is AI’s turn,  When: AI makes a LEGAL move,  Then: move AI’s piece, and change the turn to player 1.  Note: Game is still ongoing | ToDo | ALL |
|  | 13.3 | Given: an ongoing game, and after all pieces have been placed, and it is player 1’s turn,  When: player 1 makes an ILLEGAL move,  Then: do not move player 1’s piece, and the turn remains player 1’s (and a message is shown to the user).  Note: Game is still ongoing | Todo | ALL |
|  | 13.4 | Given: an ongoing game, and after all pieces have been placed, and it is player 2’s turn,  When: player 2 makes an ILLEGAL move,  Then: do not move player 2’s piece, and the turn remains player 2’s (and a message is shown to the user).  Note: Game is still ongoing | ToDo | ALL |
|  | 13.5 | Given: an ongoing game, and after all pieces have been placed, and it is a player’s turn,  When: a player hovers over their movable pieces,  Then: highlight the piece to show the user it is movable | ToDo | ALL |
|  | 13.6 | Given: an ongoing game, and after all pieces have been placed, and it is a player’s turn,  When: a player clicks on one of their movable pieces,  Then: highlight the piece and also the locations the piece can move | ToDo | ALL |
| 14 AI Behavior-3  (Mill recognition) | 14.1 | Given: an ongoing game, and it is AI’s turn,  When: directly after AI has made a legal placement/move,  Then: check if the latest piece placed/moved by AI completed any mill(s).  Note: Game is still ongoing. | ToDo | ALL |
| 15 AI Behavior-4  (Removing player’s pieces legally) | 15.1 | Note: This story could lead to a change in the game status  Given: an ongoing game, and player 1 just placed or moved a piece,  When: player 1 makes a mill and if AI has any non-mill pieces available,  Then: player 1 can remove any one of those non-mill pieces of AI. | ToDo | ALL |
|  | 15.2 | Given: an ongoing game, and AI just placed or moved a piece,  When: AI makes a mill and if player 1 has any non-mill pieces available,  Then: AI can remove any one of those non-mill pieces of player 1. | ToDo | ALL |
|  | 15.3 | Given: an ongoing game, and player 1 just placed or moved a piece,  When: player 1 makes a mill and if AI does not have any non-mill pieces available,  then player 1 can remove any of AI’s pieces from the board. | ToDo | ALL |
|  | 15.4 | Given: an ongoing game, and AI just placed or moved a piece,  When: AI makes a mill and if player 1 does not have any non-mill pieces available,  Then: AI can remove any of player 1’s pieces from the board. | ToDo | ALL |
| 16 AI Behavior-5  (Auto pilot mode) | 16.1 | Given: an ongoing game, and AI’s turn,  When: it is the beginning of AI’s turn,  Then: count AI’s pieces, and display if AI is flying | ToDo | ALL |
| 17 AI Behavior-6  (Fly ) | 17.1 | Given: an ongoing game, and AI’s turn, and AI is flying  When: AI makes a flying move,  Then: move AI’s piece and change the turn to player 1 and so on | ToDo | ALL |
| 18 AI Behavior-7  (Win condition evaluation) | 18.1 | Given: an ongoing game, and after player 1 makes a mill (or finishes their turn),  When: AI has available valid moves, and more than 2 pieces,  Then: the game continues, and it is AI’s turn,  Note: The game is ongoing | ToDo | ALL |
|  | 18.2 | Given: an ongoing game, and after AI makes a mill (or finishes their turn),  When: player 1 has available valid moves, and more than 1 pieces,  Then: the game continues, and it is player 1’s turn,  Note: The game is ongoing | ToDo | ALL |
|  | 18.3 | Given: an ongoing game, and after player 1 makes a mill (or finishes their turn),  When: AI has no available legal moves,  Then: the game is over, and player 1 wins  Note: The game is finished | ToDo | ALL |
|  | 18.4 | Given: an ongoing game, and after AI makes a mill (or finishes their turn),  When: player 1 has no available legal moves,  Then: the game is over, and AI wins  Note: The game is finished | ToDo | ALL |
|  | 18.5 | Given: an ongoing game and after player 1 has made a mill (or finishes their turn),  When: AI has only 2 pieces left,  Then: the game is over, and player 1 wins  Note: The game is finished | ToDo | ALL |
|  | 18.6 | Given: an ongoing game and after AI has made a mill (or finishes their turn),  When: player 1 has only 2 pieces left,  Then: the game is over, and AI wins | ToDo | ALL |
|  | 18.7 | Given: an ongoing game, and after each player finishes their turn,  When: the MAX moves has been reached,  Then: the game is over, and the game is a draw  Note: The game is finished | ToDo | ALL |

1. **Updated Implementation Tasks**

Summary of production code

Backend Team: Thom, Tim, Omer

GUI Team: Ami, Shawn

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **User Story ID and Name** | **AC ID** | **Class Name(s)** | **Method Name(s)** | **Developer Name(s)** | **Status** | **Notes (optional)** |
| 1 Start Game | 1.1 | Piece & Player | Pi:pair G/S  Pl: name, pieces G/S  Pl: Init, hasInBag | Backend | Complete |  |
|  | 1.1 | Game & Gameplay | Constructor  InitPlayers | Backend | Complete |  |
|  | 1.1 | Move | linkUp  isOpen  isLegal  getCoords Table  InitCoordTable | Backend | Complete |  |
|  | 1.1 | Gui | start  addVBox  addGridPane  removeVBoxElement | Shawn  Tim | Complete |  |
|  | 1.1 | Cell | Constructor  Cell  convertCoordinates  handleClick  getCoords | Shawn  Shawn  All  All  All | Complete |  |
|  | 1.2 | Cell | checkValidSpace | Ami | Complete |  |
|  | 1.3 | Cell | checkValidSpace | Ami | Complete |  |
| 2 Placing Phase | 2.1 | Cell | handleClick  setPlayer | All  Shawn | Complete |  |
|  | 2.1 | Move | move | Backend | Complete |  |
|  | 2.2 | Cell | handleClick  setPlayer | All  Shawn | Complete |  |
|  | 2.2 | Move | move | Backend | Complete |  |
|  | 2.3 | Cell | handleClick  setPlayer | All  Shawn | Complete |  |
|  | 2.3 | Move | move | Backend | Complete |  |
|  | 2.4 | Cell | handleClick  setPlayer | All  Shawn | Complete |  |
|  | 2.4 | Move | move | Backend | Complete |  |
| 3 Moving Phase | 3.1 | Cell | handleClick | Ami & Thom | Complete |  |
|  | 3.1 | Move | changeLocation = move  findPiece | Backend | Complete |  |
|  | 3.2 | Cell | handleClick | Ami & Thom | Complete |  |
|  | 3.2 | Move | changeLocation  findPiece | Backend | Complete |  |
|  | 3.3 | Cell | handleClick | Ami & Thom | Complete |  |
|  | 3.3 | Move | changeLocation  findPiece | Backend | Complete |  |
|  | 3.4 | Cell | handleClick | Ami & Thom | Complete |  |
|  | 3.4 | Move | changeLocation  findPiece | Backend | Complete |  |
|  | 3.5 | Cell | showAvailableSpaces  highlightAvailableSpace  undoHighlight  handleClick | Ami & Thom | Complete |  |
|  | 3.6 | Cell | handleClick | Thom | Complete |  |
| 4 Have Mill | 4.1 | Game | isInMill  checkMill | Thom | Complete |  |
|  | 4.1 | Move | changeLocation | Thom | Complete |  |
|  | 4.1 | Cell | handleClick | Ami | Complete |  |
|  | 4.2 | Game | isInMill  checkMill | Thom | Complete |  |
|  | 4.2 | Move | changeLocation | Thom | Complete |  |
|  | 4.2 | Cell | handleClick | Ami | Complete |  |
| 5 Take Piece | 5.1 | Game | checkForUnmilledPieces  updateGameState | Thom | Complete |  |
|  | 5.1 | Move | removePiece  findPiece | Thom | Complete |  |
|  | 5.1 | Cell | removeVisualPiece  handleClick | Ami | Complete |  |
|  | 5.2 | Game | checkForUnmilledPieces  updateGameState | Thom | Complete |  |
|  | 5.2 | Move | removePiece  findPiece | Thom | Complete |  |
|  | 5.2 | Cell | removeVisualPiece  handleClick | Ami | Complete |  |
|  | 5.3 | Game | checkForUnmilledPieces  updateGameState | Thom | Complete |  |
|  | 5.3 | Move | removePiece  findPiece | Thom | Complete |  |
|  | 5.3 | Cell | removeVisualPiece  handleClick | Ami | Complete |  |
|  | 5.4 | Game | checkForUnmilledPiece  updateGameStates | Thom | Complete |  |
|  | 5.4 | Move | removePiece  findPiece | Thom | Complete |  |
|  | 5.4 | Cell | removeVisualPiece  handleClick | Ami | Complete |  |
| 6 isFlying | 6.1 | Player | isFlying | Thom | Complete |  |
|  | 6.1 | Game | updateGameState | Thom | Complete |  |
|  | 6.1 | Move | changeLocation | Thom | Complete |  |
|  | 6.1 | Cell | handleClick | Ami & Thom | Complete |  |
|  | 6.2 | Player | isFlying | Thom | Complete |  |
|  | 6.2 | Game | updateGameState | Thom | Complete |  |
|  | 6.2 | Move | changeLocation | Thom | Complete |  |
|  | 6.2 | Cell | handleClick | Ami & Thom | Complete |  |
| 7 Flying Move | 7.1 | Move | changeLocation | Backend | Complete |  |
|  | 7.2 | Move | changeLocation | Backend | Complete |  |
|  | 7.3 | Cell | showAvailableSpaces  highlightAvailableSpace  undoHighlight  handleClick | Ami | Complete |  |
| 8 Win/Lose | 8.1 | Game | updateGameState | Thom | Complete |  |
|  | 8.1 | Cell | handleClick | All | Complete |  |
|  | 8.2 | Game | updateGameState | Thom | Complete |  |
|  | 8.2 | Cell | handleClick | All | Complete |  |
|  | 8.3 | Game | updateGameState  noMove | Thom | Complete |  |
|  | 8.3 | Cell | handleClick | Ami & Thom | Complete |  |
|  | 8.4 | Game | updateGameState  noMove | Thom | Complete |  |
|  | 8.4 | Cell | handleClick | Ami & Thom | Complete |  |
|  | 8.5 | Game | updateGameState  lostByPieceCount | Thom | Complete |  |
|  | 8.5 | Player | hasTwoPieces | Thom | Complete |  |
|  | 8.5 | Cell | handleClick | Ami & Thom | Complete |  |
|  | 8.6 | Game | updateGameState  lostByPieceCount | Thom | Complete |  |
|  | 8.6 | Player | hasTwoPieces | Thom | Complete |  |
|  | 8.6 | Cell | handleClick | Ami & Thom | Complete |  |
|  | 8.7 | Move | getMoveCount  incrementMoveCount | Thom | Complete |  |
|  | 8.7 | Game | updateGameState | Thom | Complete |  |
|  | 8.7 | Cell | handleClick | Ami & Thom | Complete |  |
|  | 9.2 | Gui | initializeInstructionsPage | Ami |  |  |
| 9 Menu | 9.3 | Gui | start  goToMenu  initializeMenu | Ami | Complete |  |
|  | 9.4 | Gui | initializeInstructionsPage | Ami | Complete |  |
| 10 Status Box | 9.1 | Gui | initializeGamePage  changeStatus | Shawn | Complete |  |
|  | 9.1 | Game | updateGuiStatus  updateGameState | Thom | Complete |  |

Summary of automated test code (directly corresponding to some acceptance criteria)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **User Story ID and Name** | **Acceptance Criterion ID** | **Class Name (s) of the Test Code** | **Method Name(s) of the Test Code** | **Description of the Test Case (input & expected output)** | **Status** | **Developer Name(s)** |
| 1 Start Game | 1.1 | GameTest | setUp | set up and instantiate the Game. | Completed | Backend Developers |
|  |  |  | tearDown | tear the Game down | Completed | Backend Developers |
|  | … |  | FirstTest | this was the first UnitTest for the app and was used to test baseline functionality | Completed | Backend Developers |
|  |  |  | ConstructorTest | testing that everything is initialized correctly | Completed | Backend Developers |
| 2 | 2.1 | MoveTest | setUp | set up of a new gameplay | Completed | Backend Developers |
|  | … |  | tearDown | test if a current game exists | Completed | Backend Developers |
|  |  |  | isOpenTest | test if a point is open to make a valid move | Completed | Backend Developers |
|  |  |  | linkUpTest | test if GUI is synced with backend | Completed | Backend Developers |
|  |  |  | isLegalTest | test whether a move is a legal and allowed moved based on constraints of a nine mill game | Completed | Backend Developers |
|  |  |  | moveTestOne | test if a piece can be successfully moved from the player’s bag to the game board point | Completed | Backend Developers |

Summary of manual test cases (directly corresponding to some acceptance criteria)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **User Story ID and Name** | **Acceptance Criterion ID** | **Test Case Input** | **Test Oracle (Expected Output)** | **Status** | **Notes** | **Developer Name(s)** |
| 1Start Game | 1.1 | Run program/ start game | Empty board generated (GUI) | Complete & accepted |  | Ami & Shawn |
|  | 1.2 | Click unplayable place | No visual change and no click handled | Complete & Accepted |  | Ami & Shawn |
|  | 1.3 | Click playable place | Click handled and visual change if approved by back end | Complete & accepted |  | Ami & Shawn |
| 2 Placing Pieces | 2.1 | Click empty playable place on Red’s turn | Red piece is placed on the location and turn changes to Blue | Complete & accepted |  | Ami & Shawn |
|  | 2.2 | Click empty playable space on Blue’s turn | Blue piece is placed on the location and turn changes to Red | Complete & accepted |  | Ami & Shawn |
|  | 2.3 | Click occupied playable space on Red’s turn | No piece is placed, no visual change and turn does not change | Complete & accepted |  | Ami & Shawn |
|  | 2.4 | Click occupied playable space on Blue’s turn | No piece is placed, no visual change and turn does not change | Complete & accepted |  | Ami & Shawn |
| 3 Moving Pieces | 3.1 | Player 1’s turn, Player 1 clicks on their piece and then an open adjacent cell | Piece is removed when clicked, available open cells are highlighted, and then the piece is placed on the new cell. The turn is switched to player 2. | Completed |  | GUI team |
|  | 3.2 | Player 2’s turn, Player 2 clicks on their piece and then an open adjacent cell | Piece is removed when clicked, available open cells are highlighted, and then the piece is placed on the new cell. The turn is switched to player 1. | Completed |  | GUI team |
|  | 3.3 | A player clicks on the other player’s piece. | The piece is not picked up and cannot be moved. The turn is not switched and the click is not handled. | Completed |  | GUI team |
|  | 3.4 | A player has clicked on their own piece and picked it up, then they click on an unavailable space | The unavailable space (not adjacent or might be occupied) is not highlighted. After clicking, the piece is not moved and the turn is not switched. | Completed |  | GUI team |
|  | 3.5 | A player has clicked on their own piece | Any available moves for that piece are highlighted on the board | Completed |  | GUI team |
|  | 3.6 | A player has clicked on their own piece and clicked on it again to cancel their move. | Available moves are un-highlighted and the move is cancelled for that piece. | Completed |  | GUI team |
| 5. Removing Pieces | 5.1 | Player 1 has made a mill and player 2 has pieces that are not in a mill on the board. | Player 1 can remove player 2’s pieces that are not in a mill when clicked. | Completed |  | GUI team |
|  | 5.2 | Player 2 has made a mill and player 1 has pieces that are not in a mill on the board. | Player 2 can remove player 1’s pieces that are not in a mill when clicked. | Completed |  | GUI team |
|  | 5.3 | Player 1 has made a mill and player 2’s pieces are all in mills. | Player 1 can remove any of player 2’s pieces by clicking on one. | Completed |  | GUI team |
|  | 5.4 | Player 2 has made a mill and player 1’s pieces are all in mills. | Player 2 can remove any of player 1’s pieces by clicking on one. | Completed |  | GUI team |
| 7. Flying | 7.1 | Player 1 is flying, they click on their piece and an available space | The piece is moved to the available space and the turn is switched to player 2 | Completed |  | GUI team |
|  | 7.2 | Player 2 is flying, they click on their piece and an available space | The piece is moved to the available space and the turn is switched to player 2 | Completed |  | GUI team |
|  | 7.3 | A player is flying and clicks on one of their pieces | Any available space is highlighted and is in mid-move | Completed |  | GUI team |
| 8. Win/Lose | 8.1 | Game is ongoing with no winner | Game continues | Completed |  | GUI team |
|  | 8.2 | Game is ongoing with no winner | Game is ongoing with no winner | Game continues |  | GUI team |
|  | 8.3 | Player 2 has no legal moves | Player 1 wins | Completed |  | GUI team |
|  | 8.4 | Player 1 has no legal moves | Player 2 wins | Completed |  | GUI team |
|  | 8.5 | Player 2 has only 2 pieces left | Player 1 wins | Completed |  | GUI team |
|  | 8.6 | Player 1 has only 2 pieces left | Player 2 wins | Completed |  | GUI team |
|  | 8.7 | The game’s move count has reached the DRAW limit | The game is a Draw | Completed |  | GUI team |
| 9. Menu | 9.1 | A player selects the option to play against an AI on the menu | The game mode switches to be VS a computer | Ongoing |  |  |
|  | 9.2 | A player clicks “how to play” on the menu | An instruction page is shown to the player. | Ongoing | The page is functioning but needs better instructions | GUI team |
|  | 9.3 | A player clicks “start game” from the main menu | The game starts with the menu settings. | Complete |  | GUI team |
|  | 9.4 | A player clicks “start game” from instructions page | The game starts with the menu settings | Complete |  | GUI team |

Summary of other automated or manual tests (not corresponding to the acceptance criteria)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Number** | **Test Input** | **Expected Result** | **Class Name of the Test Code** | **Method Name of the Test Code** | **Status** | **Developer Name(s)** |
| 1..99 | Various | Various | Various | Various | Ongoing | All |
|  |  |  |  |  |  |  |

1. **Meeting Minutes (only during this sprint)**

Report the minutes of all meetings, including, but not limited to: project/sprint planning meeting, stand-up meeting, backlog grooming, retrospective meeting, and pair programming session.

**(Minutes have been added to the end)**

1. **Buddy Ratings**

If you don’t feel comfortable to include your ratings in this report, you may email your ratings to the instructor or grader.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *Rating giver* | *Rating receiver* | | | | |  |
|  | Tim | Thom | Ami | Shawn | Omer |
| Tim | X | Facebook | Apple | Tesla | Google |
| Thom | K2 | X | Everest | Lhotse | Kangchenjunga |
| Ami | 1 | 1 | X | 1 | 1 |
| Shawn | 1 | 1 | 1 | X | 1 |
|  | Omer | 1 | 1 | 1 | 1 | X |
|  | *Average* |  |  |  |  |  |

**Date:** 10/15/2020

**Attendees**: Tim, Ami, Thom, Omer, Shawn

**Meeting Start**: 10:45 AM CST

**Meeting End**: 11:20 AM CST

**Meeting Purpose & Summary**:

* Retrospective meeting reviewing the first sprint
* Things we liked:
  + We all learned a new language (Java)
  + We learned a lot through the process
  + The user stories made things a lot easier
    - Especially for testing
  + Meetings were good
    - Less procrastinating
    - Less stress
    - Very prepared
* Things we want to change:
  + More pair programming
  + More GitHub task board usage
  + User stories earlier on
* Practice run for the in-class demo
* We ended on team appreciation :)

**Action Items for Next Meeting**:

* In-class demo (Thom and Ami)

**Date:** 10/19/2020

**Attendees‌**:‌ ‌Tim,‌ ‌Ami,‌ ‌Thom,‌ ‌Shawn, Omer

‌

**Meeting‌ ‌Start‌**:‌ ‌6:00 PM ‌CST‌ ‌

**Meeting‌ ‌End‌**:‌ ‌6:20 PM ‌CST‌ ‌

‌

**Meeting‌ ‌Purpose‌ ‌&‌ ‌Summary‌**:‌ ‌

* Discussing the goals for the next deadline - Nov. 10
  + Mills
    - Checking for mills on the back end
    - Front end handles what to do if there is a mill
  + Check the game is over
  + Add flying
  + GUI handle moving
  + Refactor GUI code

**Action Items for Next Meeting**:

* GUI needs to meet and discuss design decisions after seeing demos
  + Highlighting pieces
  + Menu on startup
  + Buttons during game (play again, menu, exit)
  + Add user stories & acceptance criteria where appropriate
* Back end discuss how to check for mills (meeting on Wednesday with Thom, Tim, Omer)
  + Add user stories & acceptance criteria where appropriate

**Date:** 10/21/2020

**Attendees‌**:‌ ‌Ami & Shawn

‌

**Meeting‌ ‌Start‌**:‌ ‌1:30 PM ‌CST‌ ‌

**Meeting‌ ‌End‌**:‌ ‌1:45 PM ‌CST‌ ‌

‌

**Meeting‌ ‌Purpose‌ ‌&‌ ‌Summary‌**:‌ ‌

* Discuss elements of design we want to add
  + Background
  + Add current move label
  + Refactor code
    - Shawn
    - Ami
  + Menu on startup
    - Ami
    - Refer to playlist from shawn
  + 3 buttons during game
    - reset/play again
    - Menu
    - exit/quit
    - Shawn will add the buttons
  + Highlighting pieces & places
    - Ami

**Action Items for Next Meeting**:

Each of us getting 2 tasks written up in user stories or acceptance criteria

Ami:

* Remove piece function
* Highlight piece function
* Highlight place on mouse-over
* Menu on startup

Shawn:

* Refactoring board drawing
* Add buttons
* Add text for who’s move it is
* Background/style

**Date:** 10/22/2020

**Attendees‌**:‌ ‌Thom, Tim, Omer, Ami, Shawn

‌

**Meeting‌ ‌Start‌**:‌ ‌10:45 AM ‌CST‌ ‌

**Meeting‌ ‌End‌**:‌ ‌11:05 AM ‌CST‌ ‌

‌

**Meeting‌ ‌Purpose‌ ‌&‌ ‌Summary‌**:‌ ‌

* Discuss deadline requirements again
* Demonstrate GUI progress
  + Color palette update
  + Refactoring
  + Highlighting cells
* Discussed tasks
  + Menu
    - Input player names

Overall, a quick check-in and update on what’s going on between our team!

**Date:** 10/26/2020

**Attendees‌**:‌ ‌Ami, Shawn, Thom

‌

**Meeting‌ ‌Start‌**:‌ 6:10‌ PM ‌CST‌ ‌

**Meeting‌ ‌End‌**:‌ ‌7:05 ‌CST‌ ‌

‌

**Meeting‌ ‌Purpose‌ ‌&‌ ‌Summary‌**:‌ ‌

* Back end showing code
  + Update methods:
    - isLegal
    - removePiece
    - findPiece
  + quickTable
    - Hashmap storing pieces that have been placed by a player
    - Can be redrawn
  + moveTable
    - Hashmap that returns an array of points that can be moved from given the current placement of a piece
  + Checking for mills
* Some pair programming to check for mills
* Discussed next week tasks
  + Pair programming with back end & front end
  + Remove pieces
  + Moving pieces (gui)

**Action Items for Next Meeting**:

* Homework for 449! We’re putting this assignment on hold until we’re ready

**Date:** 11/2/2020

**Attendees‌**:‌ ‌Ami, Shawn, Thom, Tim, Omer

‌

**Meeting‌ ‌Start‌**:‌ 6:00‌ PM ‌CST‌ ‌

**Meeting‌ ‌End‌**:‌ ‌7:00 ‌PM CST‌ ‌

‌

**Meeting‌ ‌Purpose‌ ‌&‌ ‌Summary‌**:‌ ‌

* Discussed current status
* Fixed package & ide issues
* Major focus on the click handling and how to move pieces - mainly needs to be handed on the GUI end to connect with the backend code

**Action Items for Next Meeting**:

* Ami and Tim code tomorrow working on the click to move pieces
* Anyone who can meet on Wednesday at 5:30 PM to catch up and work on any issues that come up tomorrow

**Date:** 11/4/2020

**Attendees‌**:‌ ‌Ami, Thom, Omer

‌

**Meeting‌ ‌Start‌**:‌ 6:00‌ PM ‌CST‌ ‌

**Meeting‌ ‌End‌**:‌ ‌ 7:00 PM ‌CST‌ ‌

‌

**Meeting‌ ‌Purpose‌ ‌&‌ ‌Summary‌**:‌ ‌

* Pair programming for handleClick() and removePiece()

**Date:** 11/5/2020

**Attendees‌**:‌ ‌Ami, Thom, Tim, Shawn, Omer

‌

**Meeting‌ ‌Start‌**:‌ 10:45 AM ‌CST‌ ‌

**Meeting‌ ‌End‌**:‌ ‌ 11:20 ‌CST‌ ‌

‌

**Meeting‌ ‌Purpose‌ ‌&‌ ‌Summary‌**:‌ ‌

* Discussed refactoring for handleClick and removePiece functions
* Reviewed the progress from pair programming yesterday
* Pushed progress from yesterday onto Git

**Action Items for Next Meeting**:

* Tim: work on AI user stories
* Refactoring
* Coordinate meeting for tomorrow

**Date:** 11/9/2020

**Attendees‌**:‌ ‌Ami, Thom, Tim, Shawn, Omer

‌

**Meeting‌ ‌Start‌**:‌ 5:30 PM ‌CST‌ ‌

**Meeting‌ ‌End‌**:‌ 6:‌15 ‌CST‌ ‌

‌

**Meeting‌ ‌Purpose‌ ‌&‌ ‌Summary‌**:‌ ‌

* Troubleshooting & refactoring methods in pair and cell classes

**Action Items for Next Meeting**:

* Wednesday at 4 or 5 pm to code more - pair programming
* Ami work on menu screen tomorrow
* Shawn work on refactoring GUI drawing code

**Date:** 11/11/2020

**Attendees‌**:‌ ‌Ami, Thom, Tim

‌

**Meeting‌ ‌Start‌**:‌ 5:30 PM ‌CST‌ ‌

**Meeting‌ ‌End‌**:‌ 7:00 PM ‌CST‌ ‌

‌

**Meeting‌ ‌Purpose‌ ‌&‌ ‌Summary‌**:‌ ‌

* Programming together and showing progress

**Action Items for Next Meeting**:

* MENU UPDATE follow these instructions to add javafx control module <https://www.jetbrains.com/help/idea/javafx.html#vm-options>
  + Related stack overflow page: <https://stackoverflow.com/questions/54291958/javafx-11-illegalaccesserror-when-creating-label>
* Ami - work on showing available moves visually when moving a piece
* Ami - handle highlight/unhighlight on hover needs changes

**Date:** 11/12/2020

**Attendees‌**:‌ ‌Ami, Thom, Tim, Omer, Shawn

‌

**Meeting‌ ‌Start‌**:‌ 11:15 AM ‌CST‌ ‌

**Meeting‌ ‌End‌**:‌ 11:25 AM ‌CST‌ ‌

‌

**Meeting‌ ‌Purpose‌ ‌&‌ ‌Summary‌**:‌ ‌

* Discuss submission requirements
  + Project report
* Updates
  + Highlighting in progress (Ami)
    - Show available moves
    - Not unhighlighting available moves on hover
  + Move and mill recognizing is good to go (Thom)
    - Need to prevent removing a piece that is in a mill
  + Tim has been working on testing
  + Menu progress
    - Could use an aesthetic update but is functional

**Action Items for Next Meeting**:

* Shawn work on refactoring
* Ami work on cell highlighting
* Meeting tomorrow to work on highlighting

**Date:** 11/16/2020

**Attendees**: Tim, Ami, Thom, Omer, Shawn

**Meeting Start**: 5:30 PM CST

**Meeting End**: 7:00 PM CST

**Meeting Purpose & Summary**:

* We discussed the requirements left for the report and what will be needed for the demo
  + We cleaned up code some for the demo and ran through a couple practice demos
* We worked on the report as a team until completion
* Demonstrated progress from this weekend
  + Can only remove pieces not in mills
  + Highlighting and click handling
* Discussed submission requirements

**Action Items for Next Meeting**:

* Wait until after presentation, and forumalute where we could improve and review our performance for 2nd iteration as a team.