**Team Project Sprint #2**

Instructions

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: Overwork and Understaffed

Team Members: Joshua Koni, Elizabeth Nastoff, Ruby Rios, Marley Symmonds

1. **Updated User Stories**

Highlighted user stories were added during this sprint.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **User Story Name** | **User Story Description** | **Priority** | **Estimated effort (hours)** | **Actual effort (if completed)** | **Status (completed, toDo, inProgress)** | **Developer names** |
| 1 | Board Visualization | As a user, I want to be able to visualize the board so I can see the game. | Very High – Sprint 1 | 3 | 4 | Completed | Elizabeth Nastoff |
| 2 | Piece Visualization | As a user, I want to be able to place my pieces so I can see the game. | High – Sprint 1 | 3 | 5 | Completed | Joshua Koni |
| 3 | Turn-based Gameplay | As a user, I want the system to track whose turn it is so I can only play on my turn. | High –  Sprint 1 | 2 | 3 | Completed | Joshua Koni, Elizabeth Nastoff,  Marley Symmonds |
| 4 | Piece Differentiation | As a user, I want to know which color I am so I can know which pieces are mine. | High – Sprint 1 | 1 | 2 | Completed | Joshua Koni, Elizabeth Nastoff,  Marley Symmonds |
| 5 | Start Screen Functionality | As a user, I want to be able to select whether I want to play against a computer or another player, so I can choose who I play against. | High– Sprint 2 | 3 | 3 | Completed | Joshua Koni, Elizabeth Nastoff, Ruby Rios |
| 6 | End Screen Functionality - Close | As a user, I want to be able to close the game so I can end the application when I am done. | High– Sprint 2 | 1 | 1 | Completes | Joshua Koni, Elizabeth Nastoff, Ruby Rios |
| 7 | End Screen Functionality - Restart | As a user, I want to be able to reset the game so I can start a new game. | High – Sprint 2 | 1 | 2 | Completed | Joshua Koni, Elizabeth Nastoff, Ruby Rios |
| 8 | Player vs Player Functionality | As a user, I want to be able to play against another player on the same device so I can play against my friends when we are using the same computer. | High– Sprint 2 | 4 | 6 | Completed | Elizabeth Nastoff, Marley Symmonds |
| 9 | “Placing Pieces” Phase Production | As a user, I want to be able to place my pieces anywhere on the board where there is not a piece currently during the Placing Pieces phase of the game, so that I can play the game. | High – Sprint 2 | 4 | 3 | Completed | Joshua Koni,  Elizabeth Nastoff, Marley Symmonds |
| 10 | “Moving Pieces” Phase Production | As a user, I want to be able to move my pieces one space on the board where there is not a piece currently during the Moving Pieces phase of the game, so that I can play the game. | High– Sprint 2 | 4 | 4 | Completed | Joshua Koni, Elizabeth Nastoff, Marley Symmonds |
| 11 | “Flying” Phase Production | As a user, when I have only three pieces left, I want to be able to move my pieces anywhere on the board where there is not a piece currently during the Flying phase of the game, so that I can play the game. | High – Sprint 2 | 4 | 3 | Completed | Elizabeth Nastoff, Joshua Koni, Marley Symmonds |
| 12 | Mill Functionality | As a user, when I have moved three pieces into a “mill” (all of the pieces are in consecutive spaces in a row or column), I want to be able to remove one of my opponents’ pieces, so that I can play the game. | High – Sprint 2 | 2 | 4 | Completed | Joshua Koni, Elizabeth Nastoff, Marley Symmonds |
| 13 | End Screen – Win/Loss | As a user, I want to know if I have won or lost so I can know if the game is done. | Medium – Sprint 2 | 1 | 2 | Completed for Player vs Player, In Progress for Player vs AI | Joshua Koni, Elizabeth Nastoff, Ruby Rios |
| 14 | Computer Selection | As a user, I want to be able to play against the computer so I can play solo. | Medium - Sprint 3 | 1 | 0.5 | In Progress | Ruby Rios |
| 15 | Computer Functionality- “Placing Pieces” Phase Production | As a user, I want the computer to place and move pieces so I can play against it. | Medium – Sprint 3 | 4 |  | To Do |  |
| 16 | Computer Functionality- “Moving Pieces” Phase Production | As a user, I want the computer to place and move pieces so I can play against it. | Medium – Sprint 3 | 4 |  | To Do |  |
| 17 | Computer Functionality- “Flying” Phase Production | As a user, I want the computer to place and move pieces so I can play against it. | Medium – Sprint 3 | 4 |  | To Do |  |

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

Highlighted acceptance criteria were added during this sprint.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User Story ID and Name** | **AC**  **ID** | **Description of Acceptance Criterion** | **Status (completed, toDo, inPprogress)** | **Developer Names** |
| 1 Board Visualization | 1.1 | When an option is selected from the start screen for player opponent, then 16 lines and 24 circles will appear on the screen in the formation of the Nine Men’s Morris board. | Completed | Elizabeth Nastoff |
| 2 Piece Visualization | 2.1 | Given a board is displayed on the screen, when the screen is clicked on a circle where there are not any existing pieces and it is the “Placing Pieces” phase, then a piece is placed on the screen at that location. | Completed | Joshua Koni, Elizabeth Nastoff |
|  | 2.2 | Given a board is displayed on the screen, when the screen is clicked on a circle where there is an existing piece and it is the “Placing Pieces” phase, then a piece will not be placed on the screen at that location. | Completed | Joshua Koni, Elizabeth Nastoff |
|  | 2.3 | Given a board is displayed on the screen, when the screen is clicked in a location where a piece cannot go and it is the “Placing Pieces” phase, then a piece will not be placed on the screen at that location. | Completed | Joshua Koni, Elizabeth Nastoff |
| 3 Turn-based Gameplay | 3.1 | Given a game is started, when it is the user’s turn, then they will be able to make an action according to what phase the game is in. | Completed | Joshua Koni, Elizabeth Nastoff, Marley Symmonds |
|  | 3.2 | Given a game is started, when it is the opponent’s turn, then they will be able to make an action according to what phase the game is in. | Completed | Joshua Koni, Elizabeth Nastoff, Marley Symmonds |
| 4 Piece Differentiation | 4.1 | Given a board is displayed on the screen and the board is clicked in a place where there are not any existing pieces, when it is the user’s turn and it is Phase 1, then the piece placed is in the color of the user. | Completed | Joshua Koni, Elizabeth Nastoff, Marley Symmonds |
|  | 4.2 | Given a board is displayed on the screen and the board is clicked in a place where there are not any existing pieces, when it is the opponent’s turn and it is Phase 1, then the piece placed is in the color of the opponent. | Completed | Joshua Koni, Elizabeth Nastoff, Marley Symmonds |
| 5 Start Screen Functionality | 5.1 | When the application has been opened, then a screen will be displayed with buttons for choosing an opponent and for viewing the rules. | Completed | Joshua Koni, Elizabeth Nastoff, Ruby Rios |
|  | 5.2 | Given the application has been opened, when the button for viewing the rules is clicked on, then a list of rules for the game will be displayed, as well as a button to return to the start screen. | Completed | Joshua Koni, Elizabeth Nastoff, Ruby Rios |
| 6 End Screen Functionality – Close | 6.1 | Given that a game has ended and an end screen appears, when the user clicks the “close” button, the application will close. | Completed | Joshua Koni, Elizabeth Nastoff, Ruby Rios |
| 7 End Screen Functionality – Restart | 7.1 | Given that a game has ended and an end screen appears, when the user clicks the “restart” button, the application will reset (the board will appear on the screen with no pieces on it, a new coin toss will be performed, the game will start in phase 1)  . | Completed | Joshua Koni, Elizabeth Nastoff, Ruby Rios |
| 8 Player vs Player Functionality | 8.1 | Given that a game has started and the option to play against another player has been selected on the start screen, when the user is playing the game, then they will be able to play against another player on the same computer with full game functionality. | Completed | Elizabeth Nastoff, Marley Symmonds |
| 9 “Placing Pieces” Phase Production | 9.1 | Given that a game has started, when it is the user’s turn, less than nine pieces of the players have been placed on the board, and the place where the user clicks does not have any existing pieces on it, then the user will be able to play down a piece in this space. | Completed | Joshua Koni,  Marley Symmonds,  Elizabeth Nastoff |
|  | 9.2 | Given that a game has started, when it is the opponent’s turn, less than nine pieces of the opponents have been placed on the board, and the place where the opponent clicks does not have any existing pieces on it, then the opponent will be able to play down a piece in this space. | Completed | Joshua Koni,  Marley Symmonds,  Elizabeth Nastoff |
|  | 9.3 | Given that a game has started, when it is the user’s turn, less than nine pieces of the players have been placed on the board, and the place where the user clicks does have an existing piece on it, then the game will do nothing. | Completed | Joshua Koni,  Marley Symmonds,  Elizabeth Nastoff |
|  | 9.4 | Given that a game has started, when it is the opponent’s turn, less than nine pieces of the opponents have been placed on the board, and the place where the opponent clicks does have an existing piece on it, then the game will do nothing. | Completed | Joshua Koni,  Marley Symmonds,  Elizabeth Nastoff |
| 10 “Moving Pieces” Phase Production | 10.1 | Given that both players have placed all nine of their pieces, when the user clicks on a piece that is theirs and then click on a spot on the board that does not have any existing pieces on it, then the user will be able to move a piece to this space (the piece will disappear from its original position and reappear where the user has clicked). | Completed | Joshua Koni, Elizabeth Nastoff, Marley Symmonds |
|  | 10.2 | Given that both players have placed all nine of their pieces, when the opponent clicks on a piece that is theirs and then click on a spot on the board that does not have any existing pieces on it, then the opponent will be able to move a piece to this space (the piece will disappear from its original position and reappear where the user has clicked). | Completed | Joshua Koni, Elizabeth Nastoff, Marley Symmonds |
|  | 10.3 | Given that both players have placed all nine of their pieces, when the user clicks on a piece that is theirs and then click on a spot on the board that has an existing piece on it, then the game will do nothing. | Completed | Joshua Koni, Elizabeth Nastoff, Marley Symmonds |
|  | 10.4 | Given that both players have placed all nine of their pieces, when the opponent clicks on a piece that is theirs and then click on a spot on the board that has an existing piece on it, then the game will do nothing. | Completed | Joshua Koni, Elizabeth Nastoff, Marley Symmonds |
|  | 10.5 | Given that both players have placed all nine of their pieces, when the user clicks on a piece that is not theirs or a spot on the board that does not have a piece on it and then click on any spot on the board then the game will do nothing. | Completed | Joshua Koni, Elizabeth Nastoff, Marley Symmonds |
|  | 10.6 | Given that both players have placed all nine of their pieces, when the opponent clicks on a piece that is not theirs or a spot on the board that does not have a piece on it and then click on any spot on the board then the game will do nothing. | Completed | Joshua Koni, Elizabeth Nastoff, Marley Symmonds |
| 11 “Flying” Phase Production | 11.1 | Given that the user has only 3 pieces remaining on the board, when it is the user’s turn, the user clicks on one of their own pieces, and the place where the user clicks does not have any existing pieces on it, then the user will be able to move a piece to this space (the piece will disappear from its original position and reappear where the user has clicked). | Completed | Joshua Koni, Elizabeth Nastoff, Marley Symmonds |
|  | 11.2 | Given that the opponent has only 3 pieces remaining on the board, when it is the opponents turn, the opponent clicks on one of their own pieces, and the place where the opponent clicks does not have any existing pieces on it, then the opponent will be able to move a piece to this space (the piece will disappear from its original position and reappear where the user has clicked). | Completed | Joshua Koni, Elizabeth Nastoff, Marley Symmonds |
|  | 11.3 | Given that the user has only 3 pieces remaining on the board, when it is the user’s turn, the user clicks on one of their own pieces, and the place where the user clicks has an existing piece on it, then the game will do nothing. | Completed | Joshua Koni, Elizabeth Nastoff, Marley Symmonds |
|  | 11.4 | Given that the opponent has only 3 pieces remaining on the board, when it is the opponent’s turn, the opponent clicks on one of their own pieces, and the place where the opponent clicks has an existing piece on it, then the game will do nothing. | Completed | Joshua Koni, Elizabeth Nastoff, Marley Symmonds |
|  | 11.5 | Given that the user has only 3 pieces remaining on the board, when it is the user’s turn, the user clicks on a piece that is not their own or a space on the board that does not have any pieces and any other spot on the board, then the game will do nothing. | Completed | Joshua Koni, Elizabeth Nastoff, Marley Symmonds |
|  | 11.6 | Given that the opponent has only 3 pieces remaining on the board, when it is the opponent’s turn, the opponent clicks on a piece that is not their own or a space on the board that does not have any pieces and any other spot on the board, then the game will do nothing. | Completed | Joshua Koni, Elizabeth Nastoff, Marley Symmonds |
| 12 Mill Functionality | 12.1 | Given that a game is going, when 3 pieces of the users are aligned consecutively vertically or horizontally, then the player has formed and mill and gets to remove one of their opponent’s pieces from the board. | Completed | Joshua Koni, Elizabeth Nastoff, Marley Symmonds |
|  | 12.2 | Given that a game is going, when 3 pieces of the opponents are aligned consecutively vertically or horizontally, then the opponent has formed and mill and gets to remove one of the user’s pieces from the board. | Completed | Joshua Koni, Elizabeth Nastoff, Marley Symmonds |
| 13 End Screen – Win/Loss | 13.1 | Given that a game has ended, the user has won, and the game was player vs player, when the end screen is displayed, then it will say that the user has won. | Completed | Joshua Koni, Elizabeth Nastoff, Ruby Rios |
|  | 13.2 | Given that a game has ended, the opponent has won, and the game was player vs player, when the end screen is displayed, then it will say that the opponent has won. | Completed | Joshua Koni, Elizabeth Nastoff, Ruby Rios |
|  | 13.3 | Given that a game has ended, the user has won, and the game was player vs computer, when the end screen is displayed, then it will say that the user has won. | In Progress | Joshua Koni, Elizabeth Nastoff, Ruby Rios |
|  | 13.4 | Given that a game has ended, the opponent has won, and the game was player vs computer, when the end screen is displayed, then it will say that the user has lost. | In Progress | Joshua Koni, Elizabeth Nastoff, Ruby Rios |
| 14 Computer Selection | 14.1 | Given that a game has started and the start screen is displayed, when the user selects to play against a computer, then the user will begin a game against a computer. | In Progress | Ruby Rios |
| 15 Computer Functionality- “Placing Pieces” Phase Production | 15.1 | Given that a game has started and the option to play against a computer has been selected on the start screen, when the computer is given the option to place their first piece, if none of the user’s pieces have been played, the computer will place their piece on a random place towards the center of the board. | To Do |  |
|  | 15.2 | Given that a game has started and the option to play against a computer has been selected on the start screen, when the computer is given the option to place their first piece, if the user has placed a piece, the computer will place a piece adjacent to the user’s piece. | To Do |  |
|  | 15.3 | Given that a game has started and the option to play against a computer has been selected on the start screen, when the computer is still in the placing pieces phase of gameplay, when given an option to place a piece, if the ability to place a piece in a mill is available, the computer will complete a mill. | To Do |  |
|  | 15.4 | Given that a game has started and the option to play against a computer has been selected on the start screen, when the computer is still in the placing pieces phase of gameplay, when given an option to place a piece, if the ability for the computer to complete a mill is not available and the user has 2 adjacent pieces next to each other, the computer will block the user from completing a mill. | To Do |  |
|  | 15.5 | Given that a game has started and the option to play against a computer has been selected on the start screen, when the computer is still in the placing pieces phase of gameplay, when given an option to place a piece, if the ability for the computer to complete a mill is not available for either player, if there is a place where the computer can play a piece adjacent to where it’s previously placed a piece and where the additional space in the mill is open, the computer will place the piece there. | To Do |  |
|  | 15.6 | Given that a game has started and the option to play against a computer has been selected on the start screen, when the computer is still in the placing pieces phase of gameplay, when given an option to place a piece, if the ability for the computer to complete a mill is not available for either player and there is no open spots on the board where the computer can work towards a mill, the computer will place a piece where there is an opportunity to form a new mill. | To Do |  |
|  | 15.7 | Given that a game has started and the option to play against a computer has been selected on the start screen, when the computer is still in the placing pieces phase of gameplay, when given an option to place a piece, if the ability for the computer to complete a mill or form a new mill is not available for either player, the computer will place a piece in one of the remaining open spots on the board. | To Do |  |
| 16 Computer Functionality- “Moving Pieces” Phase Production | 16.1 | Given that a game has started and the option to play against a computer has been selected on the start screen, when the computer is in the moving phase, when given an option to move a piece, if the ability to move a piece into a mill is available, the computer will complete a mill. | To Do |  |
|  | 16.2 | Given that a game has started and the option to play against a computer has been selected on the start screen, when the computer is in the moving phase, when given an option to move a piece, if the ability to move a piece in mill is not available, the user has 2 pieces adjacent to each other, and the computer can move a piece in a place that will prevent a mill from being formed, the computer will block a mill. | To Do |  |
|  | 16.3 | Given that a game has started and the option to play against a computer has been selected on the start screen, when the computer is in the moving phase, when given an option to move a piece, if the ability to move a piece in mill or to block a user’s mill is not available, if the computer has pieces already in a mill, it will move one of those pieces out of the mill to create the opportunity to form another mill in another turn. | To Do |  |
|  | 16.4 | Given that a game has started and the option to play against a computer has been selected on the start screen, when the computer is in the moving phase, when given an option to move a piece, if the ability to move a piece in mill, block a user’s mill, or to move piece out of an existing mill is not available, if the ability is open for the computer to move a piece adjacent to another piece that it already has, it will do so. | To Do |  |
|  | 16.5 | Given that a game has started and the option to play against a computer has been selected on the start screen, when the computer is in the moving phase, when given an option to move a piece, f the ability to move a piece in mill, block a user’s mill, to move piece out of an existing mill, or to move a piece adjacent to an existing piece it already has is not available, the computer will move a piece at random. | To Do |  |
| 17 Computer Functionality- “Flying” Phase Production | 17.1 | Given that a game has started and the option to play against a computer has been selected on the start screen, when the computer is in the flying phase, when given an option to move a piece, if the ability to move a piece into a mill is available, the computer will complete a mill. | To Do |  |
|  | 17.2 | Given that a game has started and the option to play against a computer has been selected on the start screen, when the computer is in the flying phase, when given an option to move a piece, if the ability to move a piece in mill is not available, and the user has 2 pieces adjacent to each other, the computer will block a mill. | To Do |  |
|  | 17.3 | Given that a game has started and the option to play against a computer has been selected on the start screen, when the computer is in the flying phase, when given an option to move a piece, if the ability to move a piece in mill or to block a user’s mill is not available, if the computer has pieces already in a mill, it will move one of those pieces out of the mill to create the opportunity to form another mill in another turn. | To Do |  |
|  | 17.4 | Given that a game has started and the option to play against a computer has been selected on the start screen, when the computer is in the flying phase, when given an option to move a piece, if the ability to move a piece in mill, block a user’s mill, or to move piece out of an existing mill is not available, if the ability is open for the computer to move a piece adjacent to another piece that it already has, it will do so. | To Do |  |
|  | 17.5 | Given that a game has started and the option to play against a computer has been selected on the start screen, when the computer is in the flying phase, when given an option to move a piece, f the ability to move a piece in mill, block a user’s mill, to move piece out of an existing mill, or to move a piece adjacent to an existing piece it already has is not available, the computer will move a piece at random. | To Do |  |

1. **Updated Implementation Tasks**

Include the tasks from the previous report and highlight the new tasks with a different color.

Summary of production code

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **User Story ID and Name** | **AC ID** | **Class Name(s)** | **Method Name(s)** | **Developer Name(s)** | **Status** | **Notes (optional)** |
| 1 Board Visualization | 1.1 | Board, Main | drawBoard() | Elizabeth Nastoff | Completed | Based on design criteria presented in class, new classes were added. |
| 2 Piece Visualization | 2.1 | Board | drawBoard(), handleMouseClick(s,t) | Joshua Koni, Elizabeth Nastoff | Completed | Based on design criteria presented in class, new classes were added. |
|  | 2.2 | Board | drawBoard(), handleMouseClick(s,t) | Joshua Koni, Elizabeth Nastoff | Completed | Based on design criteria presented in class, new classes were added. |
|  | 2.3 | Board | drawBoard(), handleMouseClick(s,t) | Joshua Koni, Elizabeth Nastoff | Completed | Based on design criteria presented in class, new classes were added. |
| 3 Turn-based Gameplay | 3.1 | Board | drawBoard(), handleMouseClick(s,t),  turnCheck() | Joshua Koni, Elizabeth Nastoff, Marley Symmonds | Completed | Based on design criteria presented in class, new classes were added. |
|  | 3.2 | Board | drawBoard(), handleMouseClick(s,t),  turnCheck() | Joshua Koni, Elizabeth Nastoff, Marley Symmonds | Completed | Based on design criteria presented in class, new classes were added. |
| 4 Piece Differentiation | 4.1 | Board | drawBoard(), handleMouseClick(s,t),  turnCheck() | Joshua Koni, Elizabeth Nastoff, Marley Symmonds | Completed | Based on design criteria presented in class, new classes were added. |
|  | 4.2 | Board | drawBoard(), handleMouseClick(s,t),  turnCheck() | Joshua Koni, Elizabeth Nastoff, Marley Symmonds | Completed | Based on design criteria presented in class, new classes were added. |
| 5 Start Screen Functionality | 5.1 | Main, Menu | drawMenu(screen) | Joshua Koni, Elizabeth Nastoff, Ruby Rios | Completed |  |
|  | 5.2 | Main, Menu | drawMenu(screen),  drawInstruction(instruction, height, screen) | Joshua Koni, Elizabeth Nastoff, Ruby Rios | Completed |  |
| 6 End Screen Functionality – Close | 6.1 | Main, Menu | drawMenu(screen),  handleMenuClick(screen, mousePosition) | Joshua Koni, Elizabeth Nastoff, Ruby Rios | Completed |  |
| 7 End Screen Functionality – Restart | 7.1 | Main, Menu, Board | drawMenu(screen),  handleMenuClick(screen, mousePosition),  clearBoard() | Joshua Koni, Elizabeth Nastoff, Ruby Rios | Completed |  |
| 8 Player vs Player Functionality | 8.1 | Board, Menu | drawBoard(),  handleMenuClick(screen, mousePosition),  handleMouseClick(s,t),  turnCheck() | Elizabeth Nastoff, Marley Symmonds | Completed |  |
| 9 “Placing Pieces” Phase Production | 9.1 | Board | drawBoard(), handleMouseClick(s,t),  turnCheck(),  phaseOne(location),  isNotTaken(num) | Joshua Koni,  Marley Symmonds,  Elizabeth Nastoff | Completed |  |
|  | 9.2 | Board | drawBoard(), handleMouseClick(s,t),  turnCheck(),  phaseOne(location),  isNotTaken(num) | Joshua Koni,  Marley Symmonds,  Elizabeth Nastoff | Completed |  |
|  | 9.3 | Board | drawBoard(), handleMouseClick(s,t),  turnCheck(),  phaseOne(location),  isNotTaken(num) | Joshua Koni,  Marley Symmonds,  Elizabeth Nastoff | Completed |  |
|  | 9.4 | Board | drawBoard(), handleMouseClick(s,t),  turnCheck(),  phaseOne(location),  isNotTaken(num) | Joshua Koni,  Marley Symmonds,  Elizabeth Nastoff | Completed |  |
| 10 “Moving Pieces” Phase Production | 10.1 | Board | drawBoard(), handleMouseClick(s,t),  turnCheck(),  isAdj(current\_spot, potential\_spot),  isNotTaken(num),  clickOne(pieceLocation),  clickTwo(pieceLocation) | Joshua Koni, Elizabeth Nastoff, Marley Symmonds | Completed |  |
|  | 10.2 | Board | drawBoard(), handleMouseClick(s,t),  turnCheck(),  isAdj(current\_spot, potential\_spot),  isNotTaken(num),  clickOne(pieceLocation),  clickTwo(pieceLocation) | Joshua Koni, Elizabeth Nastoff, Marley Symmonds | Completed |  |
|  | 10.3 | Board | drawBoard(), handleMouseClick(s,t),  turnCheck(),  isAdj(current\_spot, potential\_spot),  isNotTaken(num),  clickOne(pieceLocation),  clickTwo(pieceLocation) | Joshua Koni, Elizabeth Nastoff, Marley Symmonds | Completed |  |
|  | 10.4 | Board | drawBoard(), handleMouseClick(s,t),  turnCheck(),  isAdj(current\_spot, potential\_spot),  isNotTaken(num),  clickOne(pieceLocation),  clickTwo(pieceLocation) | Joshua Koni, Elizabeth Nastoff, Marley Symmonds | Completed |  |
|  | 10.5 | Board | drawBoard(), handleMouseClick(s,t),  turnCheck(),  isAdj(current\_spot, potential\_spot),  isNotTaken(num),  clickOne(pieceLocation),  clickTwo(pieceLocation) | Joshua Koni, Elizabeth Nastoff, Marley Symmonds | Completed |  |
|  | 10.6 | Board | drawBoard(), handleMouseClick(s,t),  turnCheck(),  isAdj(current\_spot, potential\_spot),  isNotTaken(num),  clickOne(pieceLocation),  clickTwo(pieceLocation) | Joshua Koni, Elizabeth Nastoff, Marley Symmonds | Completed |  |
| 11 “Flying” Phase Production | 11.1 | Board | drawBoard(), handleMouseClick(s,t),  turnCheck(),  isNotTaken(num),  clickOne(pieceLocation),  clickTwo(pieceLocation) | Joshua Koni, Elizabeth Nastoff, Marley Symmonds | Completed |  |
|  | 11.2 | Board | drawBoard(), handleMouseClick(s,t),  turnCheck(),  isNotTaken(num),  clickOne(pieceLocation),  clickTwo(pieceLocation) | Joshua Koni, Elizabeth Nastoff, Marley Symmonds | Completed |  |
|  | 11.3 | Board | drawBoard(), handleMouseClick(s,t),  turnCheck(),  isNotTaken(num),  clickOne(pieceLocation),  clickTwo(pieceLocation) | Joshua Koni, Elizabeth Nastoff, Marley Symmonds | Completed |  |
|  | 11.4 | Board | drawBoard(), handleMouseClick(s,t),  turnCheck(),  isNotTaken(num),  clickOne(pieceLocation),  clickTwo(pieceLocation) | Joshua Koni, Elizabeth Nastoff, Marley Symmonds | Completed |  |
|  | 11.5 | Board | drawBoard(), handleMouseClick(s,t),  turnCheck(),  isNotTaken(num),  clickOne(pieceLocation),  clickTwo(pieceLocation) | Joshua Koni, Elizabeth Nastoff, Marley Symmonds | Completed |  |
|  | 11.6 | Board | drawBoard(), handleMouseClick(s,t),  turnCheck(),  isNotTaken(num),  clickOne(pieceLocation),  clickTwo(pieceLocation) | Joshua Koni, Elizabeth Nastoff, Marley Symmonds | Completed |  |
| 12 Mill Functionality | 12.1 | Board | drawBoard(), handleMouseClick(s,t),  turnCheck(),  isMill(placed\_pieces, new\_piece) | Joshua Koni, Elizabeth Nastoff, Marley Symmonds | Completed |  |
|  | 12.2 | Game\_Class | drawBoard(), handleMouseClick(s,t),  turnCheck(),  isMill(placed\_pieces, new\_piece) | Joshua Koni, Elizabeth Nastoff, Marley Symmonds | Completed |  |
| 13 End Screen – Win/Loss | 13.1 | Main, Menu, Board | handleMouseClick(s,t),  hasWon(),  drawMenu(screen) | Joshua Koni, Elizabeth Nastoff, Ruby Rios | Completed |  |
|  | 13.2 | Main, Menu, Board | handleMouseClick(s,t),  hasWon(),  drawMenu(screen) | Joshua Koni, Elizabeth Nastoff, Ruby Rios | Completed |  |
|  | 13.3 | Main, Menu, Board | handleMouseClick(s,t),  hasWon(),  drawMenu(screen) | Joshua Koni, Elizabeth Nastoff, Ruby Rios | In Progress |  |
|  | 13.4 | Main, Menu, Board | handleMouseClick(s,t),  hasWon(),  drawMenu(screen) | Joshua Koni, Elizabeth Nastoff, Ruby Rios | In Progress |  |

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)** |
| 2 Piece Visualization | 2.1 |  | Piece\_Location(x, y. spots\_Dict),  Piece\_Location\_Test(spots\_Dict) | Piece\_Location() returns true is the x and y values input are in the range of an item in spots\_Dict. Piece\_Location\_Test() checks that this works for every coordinate on the board  The expected output is null. | Working | Joshua Koni, Ruby Rios |
|  | 2.3 |  | Piece\_Location(x, y. spots\_Dict),  Piece\_Location\_Test(spots\_Dict) | Piece\_Location() returns true is the x and y values input are in the range of an item in spots\_Dict. Piece\_Location\_Test() checks that this works for every coordinate on the board  The expected output is null. | Working | Joshua Koni, Ruby Rios |
| 3 Turn-based Gameplay | 3.1 |  | def CheckTurnTest (turn) | def CheckTurnTest (turn) gets which turn the game is at, and checks that the color of the piece and the player putting down that piece are correct.  The expected output is null. | Working | Joshua Koni, Ruby Rios |
|  | 3.2 |  | def CheckTurnTest (turn) | def CheckTurnTest (turn) gets which turn the game is at, and checks that the color of the piece and the player putting down that piece are correct.  The expected output is null. | Working | Joshua Koni, Ruby Rios |
| 4 Piece Differentiation | 4.1 |  | def CheckTurnTest (turn) | def CheckTurnTest (turn) gets which turn the game is at, and checks that the color of the piece and the player putting down that piece are correct.  The expected output is null. | Working | Joshua Koni, Ruby Rios |
| 8 Player vs Player Functionality | 8.1 |  | def CheckTurnTest (turn) | def CheckTurnTest (turn) gets which turn the game is at, and checks that the color of the piece and the player putting down that piece are correct.  The expected output is null. | Working | Joshua Koni, Ruby Rios |
| 9 “Placing Pieces” Phase Production | 9.1 |  | Piece\_Location(x, y. spots\_Dict),  Piece\_Location\_Test(spots\_Dict) | Piece\_Location() returns true is the x and y values input are in the range of an item in spots\_Dict. Piece\_Location\_Test() checks that this works for every coordinate on the board  The expected output is null. | Working | Joshua Koni, Ruby Rios |
|  | 9.2 |  | Piece\_Location(x, y. spots\_Dict),  Piece\_Location\_Test(spots\_Dict) | Piece\_Location() returns true is the x and y values input are in the range of an item in spots\_Dict. Piece\_Location\_Test() checks that this works for every coordinate on the board  The expected output is null. | Working | Joshua Koni, Ruby Rios |
|  | 9.3 |  | Piece\_Location(x, y. spots\_Dict),  Piece\_Location\_Test(spots\_Dict) | Piece\_Location() returns true is the x and y values input are in the range of an item in spots\_Dict. Piece\_Location\_Test() checks that this works for every coordinate on the board  The expected output is null. | Working | Joshua Koni, Ruby Rios |
|  | 9.4 |  | Piece\_Location(x, y. spots\_Dict),  Piece\_Location\_Test(spots\_Dict) | Piece\_Location() returns true is the x and y values input are in the range of an item in spots\_Dict. Piece\_Location\_Test() checks that this works for every coordinate on the board  The expected output is null. | Working | Joshua Koni, Ruby Rios |
| 12 Mill Functionality | 12.1 |  | def CheckMillTest() | CheckMillTest() returns true if the the pieces provided produces a mill and false if the pieces provided do not form a mill. It compares these to the isMill function.  The expected output is null. | Working | Marley Symmonds |
|  | 12.2 |  | def CheckMillTest() | CheckMillTest() returns true if the the pieces provided produces a mill and false if the pieces provided do not form a mill. It compares these to the isMill function.  The expected output is null. | Working | Marley Symmonds |

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)** |
| 1 Board Visualization | 1.1 | print(“Board has been drawn”)  return True | “Board has been drawn” prints whenever a change to the board is made. | Working | More tests done on board functionality in other tests. | Joshua Koni, Ruby Rios |
| 2 Piece Visualization | 2.2 | print("Piece has already been placed!")  Input cases for all 24 circles checked | For each circle, on first click:  x, y, i[0], i[1]  Upon repeat click:  x, y, i[0], i[1]  Piece has already been placed! | Working | Test worked when first implemented. Currently, with the beginnings of a piece counter, the test does not work, as the counter only allows 9 pieces to be placed down by each player. | Ruby Rios |
| 5 Start Screen Functionality | 5.1 | print(“Button has been pressed. Start Game.”) | “Button has been pressed. Start Game.”, before the board is drawn | Working |  | Ruby Rios |
|  | 5.2 | print("Button has been pressed. Instructions Menu Printed.") | “Button has been pressed. Instructions Menu Printed.", before the Instructions Menu is drawn | Working |  | Ruby Rios |
| 6 End Screen Functionality – Close | 6.1 | print(“Button has been pressed. Close Program.”) | “Button has been pressed. Close Program.”, before the program is closed. | Working |  | Ruby Rios |
| 7 End Screen Functionality – Restart | 7.1 | print(“Button has been pressed. Restart Game.”) | “Button has been pressed. Restart Game.”, before the start screen is redrawn. | Working |  | Ruby Rios |
| 9 “Placing Pieces” Phase Production | 9.1 | print("Phase 1 in progress, Valid piece placed.") | “Phase 1 in progress, Valid piece placed.", after a piece is placed at a valid location on the board | Working |  | Ruby Rios |
|  | 9.2 | print("Phase 1 in progress, Valid piece placed.") | “Phase 1 in progress, Valid piece placed.", after a piece is placed at a valid location on the board | Working |  | Ruby Rios |
|  | 9.3 | print("Phase 1 in progress, Invalid piece not placed.") | "Phase 21in progress, Invalid piece not placed." after a piece is not placed at an invalid location on the board | Working |  | Ruby Rios |
|  | 9.4 | print("Phase 1 in progress, Invalid piece not placed.") | "Phase 1 in progress, Invalid piece not placed." after a piece is not placed at an invalid location on the board | Working |  | Ruby Rios |
| 10 “Moving Pieces” Phase Production | 10.1 | print("Phase 2 in progress, Valid piece placed.") | “Phase 2 in progress, Valid piece placed.", after a piece is moved to a valid location on the board | Working |  | Ruby Rios |
|  | 10.2 | print("Phase 2 in progress, Valid piece placed.") | “Phase 2 in progress, Valid piece placed.", after a piece is moved to a valid location on the board | Working |  | Ruby Rios |
|  | 10.3 | print("Phase 2 in progress, Invalid piece not placed.") | "Phase 2 in progress, Invalid piece not placed." after a piece is not moved to an invalid location on the board | Working |  | Ruby Rios |
|  | 10.4 | print("Phase 2 in progress, Invalid piece not placed.") | "Phase 2 in progress, Invalid piece not placed." after a piece is not moved to an invalid location on the board | Working |  | Ruby Rios |
|  | 10.5 | print("Phase 2 in progress, Invalid piece not placed.") | "Phase 2 in progress, Invalid piece not placed." after a piece is not moved to an invalid location on the board | Working |  | Ruby Rios |
|  | 10.6 | print("Phase 2 in progress, Invalid piece not placed.") | "Phase 2 in progress, Invalid piece not placed." after a piece is not moved to an invalid location on the board | Working |  | Ruby Rios |
| 11 “Flying” Phase Production | 11.1 | print("Phase 3 in progress, Valid white piece placed.") | "Phase 3 in progress, Valid white piece placed." after a piece is moved to a valid user location on the board | Working |  | Ruby Rios |
|  | 11.2 | print("Phase 3 in progress, Valid black piece placed.") | "Phase 3 in progress, Valid white piece placed." after a piece is moved to a valid opponent location on the board | Working |  | Ruby Rios |
|  | 11.3 | print("Phase 3 in progress, Invalid white piece not placed.") | "Phase 3 in progress, Invalid white piece not placed." after a piece is not moved to an invalid user location on the board | Working |  | Ruby Rios |
|  | 11.4 | print("Phase 3 in progress, Invalid black piece not placed.") | "Phase 3 in progress, Invalid black piece not placed." after a piece is not moved to an invalid opponent location on the board | Working |  | Ruby Rios |
|  | 11.5 | print("Phase 3 in progress, Invalid white piece not placed.") | "Phase 3 in progress, Invalid white piece not placed." after a piece is not moved to an invalid user location on the board | Working |  | Ruby Rios |
|  | 11.6 | print("Phase 3 in progress, Invalid black piece not placed.") | "Phase 3 in progress, Invalid black piece not placed." after a piece is not moved to an invalid opponent location on the board | Working |  | Ruby Rios |
| 13 End Screen – Win/Loss | 13.1 | print("White Win Menu printed") | “White Win Menu printed", before the win screen for the white player is drawn. | Working |  | Ruby Rios |
|  | 13.2 | print("Black Win Menu printed") | “Black Win Menu printed", before the win screen for the black player is drawn. | Working |  | Ruby Rios |

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. **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.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Date** | **Time and Duration** | **Place** | **Participant Names** | **Purpose of the Meeting** | **Specific Action Items** |
| October 15, 2020 | 1:00 PM – 3:00 PM  2 hours | Microsoft Teams | Joshua Koni, Ruby Rios, Marley Symmonds | Retrospective Meeting to discuss Sprint 1, backlog grooming, and planning for Sprint 2 | Due to the nature of our schedules, our team has recognized that we do not have the same amount of time for meetings as we did for Sprint 1. The importance of communication for this sprint has been emphasized, and work will be done on implementation of better text-based communication systems, so then the functionality of stand-up meetings can be performed by text-based communication.  Tasks for the completion of this sprint have been assigned. |
| November 6, 2020 | 10:00 PM -10:15 PM  15 minutes | Microsoft Teams | Joshua Koni, Ruby Rios | Stand-Up Meeting to discuss current standing of existing projects | No new tasks were assigned at this meeting. |
| November 8, 2020 | 12:00 PM - 4:00 PM  4 hours | Microsoft Teams | Joshua Koni, Elizabeth Nastoff, Ruby Rios, Marley Symmonds | Pair-programming session to combine together our portions of the code | New tasks have been assigned to each member of the team. |
| November 10, 2020 | 1:00 PM -1:45 PM  45 minutes | Microsoft Teams | Joshua Koni, Elizabeth Nastoff, Ruby Rios, Marley Symmonds | Stand-Up Meeting to discuss current standing of existing projects | New tasks have been assigned to members of the team who had completed their tasks. |
| November 12, 2020 | 10:30 PM -12:00 AM  1 hour 30 minutes | Microsoft Teams | Joshua Koni, Elizabeth Nastoff, Ruby Rios | Stand-Up Meeting to discuss current standing of existing projects with Pair Programming to work through a bug in the code | Tasks had not been completed, so a timeline was better established for the completion of tasks, and members have been encouraged to continue working on their tasks. |
| November 15, 2020 | 10:45 PM -11:45 PM  1 hour | Discord | Elizabeth Nastoff, Ruby Rios | Pair Programming to work through a bug in the code | No new tasks were assigned at this meeting. |
| November 16, 2020 | 2:45 PM -3:30 PM  45 minutes | Microsoft Teams | Joshua Koni, Elizabeth Nastoff, Ruby Rios, Marley Symmonds | Stand-Up Meeting to discuss current standing of existing projects | New tasks have been assigned to members of the team who had completed their tasks. |
| November 16, 2020 | 8:45 PM -9:15 PM  30 minutes | Microsoft Teams | Joshua Koni, Elizabeth Nastoff, Ruby Rios | Pair Programming session to work through a bug in the code | No new tasks were assigned at this meeting. |
| November 17, 2020 | 12:00AM -12:20 AM  20 minutes | Microsoft Teams | Joshua Koni, Elizabeth Nastoff, Ruby Rios, Marley Symmonds | Stand-Up Meeting to discuss current standing of existing projects | No new tasks were assigned at this meeting, but a more concrete schedule for completion of tasks was established. |
| November 18, 2020 | 10:00PM -10:30 PM  30 minutes | Microsoft Teams | Joshua Koni, Elizabeth Nastoff, Ruby Rios, Marley Symmonds | Stand-Up Meeting to discuss current standing of existing projects | Test code was assigned to each member of the team. A time was scheduled to record the presentation video. |
| November 19, 2020 | 9:30PM - | Microsoft Teams | Joshua Koni, Elizabeth Nastoff, Ruby Rios, Marley Symmonds | Recording and turn in of the Sprint Report | No new tasks were assigned at this meeting. A retrospective for this sprint and backlog grooming session have been planned. |

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* | | | | |
|  | Joshua Koni | Elizabeth Nastoff | Ruby Rios | Marley Symmonds |
| Joshua Koni | X | 1.0 | 1.0 | 1.0 |
| Elizabeth Nastoff | 1.0 | X | 1.0 | 1.0 |
| Ruby Rios | 1.0 | 1.0 | X | 1.0 |
| Marley Symmonds | 1.0 | 1.0 | 1.0 | X |
|  | *Average* | 1.0 | 1.0 | 1.0 | 1.0 |