Nine Men's Morris

Midterm Report

Jesse Rupe & Shelby Speegle

Section I - Team Organization and Buddy Rating

1. Task Division
    a. **Jesse** - High-level Class Hierarchy, Theme Design, UI Design, Object Interactivity, Overall Game Logic
    b. **Shelby** - Raphael Implementation, Game Phases, GameView Design, Board Logic and Interaction, GitHub Setup and Site Deployment

2. Buddy Ratings

Person1 rating Person2	**Rating**
Jesse rating Shelby	1.0
Shelby rating Jesse	1.0

Section II - Product Backlog

User story ID	User story description	Priority (1 - 5)	Status	Estimated effort (hrs)
1	As a user, I want to start a game by placing all of my pieces on a board.	5	✓	10-15
2	Players will take turns placing pieces.	5	✓	2-5
3	Once all pieces have been placed, I want to begin moving my pieces around the board.	5	✓	3-5
4	If a player forms a line of three pieces, that player can remove one of the other player's pieces from the board.	3	✓	2-5
5	Once a player has a total of three pieces left in play, their pieces may be moved to any empty spaces on the board.	3	✓	0.5-1.0
6	Once a game is finished, I want to restart and play a new game.	2	✓	0.5-1.0
7	A game should have elements to see that current status of the players (e.g. piece count, lost pieces, etc.).	5	TODO	2-4
8	Once a game has been won the winning player should be displayed.	3	✓	0.5-1.0
9	As a user, I'd like to play against either a human or AI opponent.	4	TODO	15-20
10	The AI should be easily beatable.	3	TODO	10-15

User story ID	Acceptance criteria	Status
1	Both player's sets of pieces can be placed on the board.	✓
2	Players can take turns in game control.	✓
3	After the first phase of gameplay, users will be able to move their pieces one spot away from their piece's current	✓

	location.	
4	A line of three pieces can be formed, resulting in one of the opponent’s pieces being removed.	✓
5	Players can “fly” (move available pieces to any empty spaces on the board) once they are down to 3 remaining pieces.	✓
6	Once a game has finished, players have the ability to start a new game with a fresh board.	✓
7	Various elements displaying relative information to the player(s).	TODO
8	Winning player is displayed upon beating opponent.	✓
9	Option to play against AI is presented to player.	TODO
10	The AI is beatable.	TODO

User story ID	**Task description**	**Owner**	**Estimated effort (hrs)**	**Status**
1	Allow user to start a game by placing all of their pieces on the board.	Shelby	10-15	✓
2	Allow players to take turns in game control.	Shelby	2-5	✓
3	Allow the game to advance to a second phase,once all pieces have been placed. This second phase will limit player movement to one spot away from the pieces current spot.	Shelby	3-5	✓
4	Allow a line to be formed with three pieces. This should allow the user that formed the line to remove one of their opponent’s pieces from gameplay.	Jesse	2-5	✓
5	Players have ability to “fly” once their piece count	Jesse/Shelby	0.5-1.0	✓

	reaches 3.			
6	Game can be restarted/reset once completed.	Jesse	0.5-1.0	✓
7	UI elements displaying useful info to the players.	Jesse	2-4	TODO
8	Winning player is displayed.	Jesse	0.5-1.0	✓
9	Option to play against AI.	Jesse/Shelby	15-20	TODO
10	AI is beatable.	Jesse/Shelby	10-15	TODO

Meet #	**Time/place**	**Topics and decisions**
1	10:15 / Library	Project overview and setup of development environment.
2	10:15 / SUB	Created Game, Piece, Board, Point, and Player classes.
3	10:15 / SUB	Created basic visuals and introduced GameView class.
4	10:15 / SUB	Applied Raphael.js library for drawing UI elements.
5	10:15 / SUB	Started Line class conceptualization and implementation.
6	10:15 / SUB	Completed Line class/object implementation.
7	6:00 / Jesse’s	Created player piece ‘Graveyards’ where dead pieces go.
8	10:15 / SUB	Began working on game Phase 2 logic and implementation.
9	5:00 / Shelby’s	Wrapped up Phase 2 gameplay.
10	10:15 / SUB	Implemented game over and concluded with a playable, functional game for two human players.

Section III - Summary of Test-Driven Development and Refactoring

*Note: Tests were conducted by hand as much of the product was difficult to test strictly with code-based tests (most of which would have involved hard-coded inputs to create a situation we could simply do my hand anyway).

Test #	**Description of test case (input & oracle)**	**User story # and task #**	**Developer(s)**
1	Visual confirmation that a player's piece has been placed on the board during the piece placement phase.	1	Shelby
2	Visual confirmation that during a player's turn, they cannot move the opponent's pieces. Player control is swapped at the conclusion of each turn.	2	Shelby
3	Once all pieces are placed, player's piece movement is limited to adjacent spots only.	3	Shelby
4	If a player forms a line of three adjacent pieces, then at the end of their turn they are able to remove one of their opponent's pieces from the board.	4	Jesse/Shelby
5	When a player has three remaining pieces, their piece movement is no longer limited to adjacent places. Pieces may be placed on any unoccupied spot.	5	Jesse/Shelby
6	A new game can be started after one has been completed.	6	Jesse
7	Test to see if a player is stuck due to the opponent placing pieces in such a way that the player is blocked. In this case the game is over, and the blocked player loses.	3	Jesse/Shelby

8	When a player is reduced to two pieces in phase one, the game is considered game over.	1	Jesse

Refactor #	**Description of the refactoring (problem & solution)**	**Developer(s)**
1	We refactored the way we detected Mill formations. Our initial approach was to set up relations between Points, hoping that lines could be determined based on these relationships. The new approach introduced a Line class that keeps track of which Points are part of a line.	Jesse and Shelby
2	We switched from drawing UI elements in html to generating them on the fly by utilizing the Raphael.js library.	Jesse and Shelby

Section IV - Summary of Pair Development

	Date/time/duration	**Place**	**Developer names**	**Tasks**
1	9/16 - 10:30AM - 1.5hrs	SUB	Jesse and Shelby	Project Setup
2	9/18 - 10:30AM - 1.5hrs	SUB	Jesse and Shelby	Web Restructure
3	9/23 - 10:30AM - 1.5hrs	SUB	Jesse and Shelby	Class Creation
4	9/27 - 10:30AM - 5.0hrs	Jesse Home	Jesse and Shelby	Simple Visuals
5	9/30 - 10:30AM - 2.0hrs	SUB	Jesse and Shelby	Piece Class
6	10/2 - 10:30AM - 1.5hrs	SUB	Jesse and Shelby	Neighbor Logic
7	10/7 - 10:30AM - 2.5hrs	SUB	Jesse and Shelby	Line Refactor
8	10/9 - 10:30AM - 1.5hrs	SUB	Jesse and Shelby	Line Details
9	10/11 - 10:30AM - 1.0hrs	SUB	Jesse and Shelby	Graveyards
10	10/14 - 10:30AM - 1.5hrs	SUB	Jesse and Shelby	Phase 2 - start
11	10/14 - 10:30AM - 2.0hrs	Java Cafe	Jesse and Shelby	Phase 2 - 99%
12	10/16 - 10:30AM - 1.5hrs	SUB	Jesse and Shelby	Game Over
13	10/21 - 10:30AM - 1.5hrs	SUB	Jesse and Shelby	Basics Done

Section V - Lessons Learned

Shelby Section

1. This project has helped me grow as a developer working with another team member. I tend to enjoy working alone on projects and this has allowed me to see another developer's feedback and perspective throughout the software construction process.
2. Our program I feel is fairly well rounded in terms of functionality. One thing I would like to make better is to improve the usability of the game. I feel that the pieces are not too terribly fun to place. Perhaps being able to drag a piece onto its destination place would make things more enjoyable.
3. I believe our team is very capable of putting together great software. We did a fine job with this project, but I feel like we would have been better off had we drawn more things out, and really figured more things out on say a whiteboard before implementing functionality with software.

Jesse Section

1. What did you personally gain from the project?
   a. Shelby and I are longtime friends and have worked together on projects in the past. This particular project I got to learn a lot from him regarding how object-orientation works for web-based languages, specifically javascript. I had played around with it myself, but after doing this I feel comfortable implementing it in my own environments (mainly work). I also enjoyed learning about how all of the game's' components are required to work together in order for the game to work as a whole. I plan to pursue game development and design in my future, so even though this isn't near the scale of what I dream of it's still fun to play with.

2. What does your program do well, and what could your program do better?
   a. The first thing that comes to mind is we have a lot of hard-coded parts that work great but feel hacky. For example, each point on the board has a hard-coded array containing that point's neighboring points. An idea we had early on (that got quickly shut down due to it being a lot of work) was to build a system that could walk around the board given starting points and such. Every time we would need to check neighbors or successful lines we could just call that system with a starting point and what to look for and it would return respectively important data. This might be something we tackle later on is we have extra time after

building the other necessities, but we'll cross that bridge when/if we get there.

3. How could you improve your development process if you develop a similar game from scratch?
    a. Honestly it totally depends on our time schedule. Given the same amount of time I suppose we would mainly focus on avoiding some specific instances that slowed us down. For example, our neighbor system that I just explained above was redesigned and rewritten at least three separate times as the rest of our game took shape. With another chance I'd love to either immediately implement the system we have now or something even better.