

# CS449 Sprint 1 Report

Michael Cu, Elias Julian Marko Garcia, Samuel Lim

October 6, 2019

## Contents

<b>1</b>	<b>Micro Charter</b>	<b>2</b>
1.1	N Men Morris . . . . .	2
<b>2</b>	<b>User Stories</b>	<b>3</b>
2.1	Mills Board . . . . .	4
2.2	User Input and Selection . . . . .	4
2.3	Starting a Game . . . . .	4
2.4	Assigning Players . . . . .	5
2.5	Piece Placement . . . . .	5
2.6	Piece Movement . . . . .	5
2.7	Mill Formation . . . . .	5
2.8	Piece Elimination . . . . .	5
2.9	Flying Pieces . . . . .	6
2.10	Defining End Game . . . . .	6
2.11	Restarting and Replaying a Game . . . . .	6
<b>3</b>	<b>Acceptance Criteria</b>	<b>6</b>
3.1	Criterion 1 . . . . .	7
<b>4</b>	<b>Implementation Tasks</b>	<b>7</b>
4.1	Summary of Production Code . . . . .	7
4.1.1	Class QUX . . . . .	7
4.2	Automated Test Code . . . . .	8
4.3	Manual Test Code . . . . .	8
4.4	Other Manual Test Code . . . . .	8
<b>5</b>	<b>Meeting Minutes</b>	<b>9</b>

# 1 Micro Charter

## 1.1 N Men Morris

### **Vision Statement**

Create a extensible framework for board game web apps with scalability and performance.

### **Mission Statement**

To play Nine Men's Morris on the web browser using a composable web technology stack that allows for future modularity while not foregoing performance.

### **Elevator Pitch and Business Value**

We are creating Nine Men's Morris on a board game framework using Express.js and Neon for Rust. This allows for a data and type safe application that is capable of composability, scalability, extensibility, and performance.

### **Customers and Users**

- Customers: Entrepreneurs and ventures that want to deploy board games on the web with low overhead, latency, and maintenance.
- Users: Individuals who are passionate about board games and want a new online experience that they can take and play wherever they go with their friends.

### **Metrics**

By benchmarking N Men Morris, we can compare our solution to other products on the market on:

1. latency
2. binary size
3. up-time

## **Milestones**

1. First MVP
2. First Offline N Men's Morris
3. Player versus Player (Offline)
4. Player versus Player (Online)

## **Risks**

1. Inherent complexity of technology stack.
2. Inability to cooperate with teammates.
3. Plausibility of orphaning project due to development team size.

## **Authors**

## **2 User Stories**

Below you will find a table that makes up our "User Story Board", with some simplifications taken with respect to the total contents of the board. With respect to the final formal documentation, i.e. this paper, we only keep the basic qualitative and quantitative values for each story in the table while giving each user story proper its own section. This makes documenting each story less unruly while also easier to read. Each Story I.D. (SID) value is internally linked to its respective story, which also helps with navigating this section.

SID	Story Name	Priority	Time Est. (hr)	Actual (hr)	Status	Developer(s)
S1	Mills Board	high	10hr			
S2	User Input and Selection	high	10hr			
S3	Starting a Game					
S4	Assigning Players					
S5	Piece Placement					
S6	Piece Movement					
S7	Mill Formation					
S8	Piece Elimination					
S9	Flying Pieces					
S10	Defining End Game					
S11	Restarting/Replaying Game					

## 2.1 Mills Board

### Description

As a user, I need an empty board consisting of 4 expanded squares with 8 equidistant positions each to play a game of Nine Men's Morris.

## 2.2 User Input and Selection

### Description

As a user, I need to be able to select and choose input from the web GUI of the application to be able to play and take turns at Nine Men's Morris.

## 2.3 Starting a Game

### Description

As a user, I need a GUI to prompt me with the options to start a game with either another human or against the computer for Nine Men's Morris in order to play the game.

## **2.4 Assigning Players**

### **Description**

As a user, I need to be assigned the role as either the first or second player, whether against another human or the computer, in order to know my player turn (either first or second) in the game.

## **2.5 Piece Placement**

### **Description**

As a user, I need to place nine pieces on unoccupied positions in turn with another player to start off a game of Nine Men's Morris.

## **2.6 Piece Movement**

### **Description**

As a user, I need to be able to move my pieces into adjacent positions that are not occupied by the other player or adjacent to their mill in order to take a turn.

## **2.7 Mill Formation**

### **Description**

As a user, I need the game to recognize that I have formed a mill upon moving three of my own pieces into adjacent positions so that I may gain the future ability to attack and defend my mill pieces from being eliminated.

## **2.8 Piece Elimination**

### **Description**

As a user, after forming a mill, I need the ability to remove an opponent's piece of my choosing so long as either it is not in a mill or any piece given all available pieces are in a mill, so that I may appropriately attack my opponent.

## 2.9 Flying Pieces

### Description

As a user, upon reaching three remaining pieces, I need the ability to fly (jump) my pieces across the board to any un-occupied point in order to play Nine Men's Morris according to the rules. Whether the position is guarded is a variant of the game, implementation decision TBD.

## 2.10 Defining End Game

### Description

As a user, when either myself or the opponent reaches less than three pieces, i.e. two pieces, I need the game and to declare the respective winner in order to successfully finish a game of Nine Men's Morris.

## 2.11 Restarting and Replaying a Game

### Description

As a user, after having completed a game of Nine Men's Morris, I need the GUI to prompt me to either play again or to end the game software so that I can accordingly choose whether to keep playing or to end my game session.

### [Template User Story]

#### Description

#### Priority

#### Estimate

#### Actual

#### Status

#### Developer

## 3 Acceptance Criteria

The following section covers the acceptance criteria enumerated in response to the User Stories discovered and documented in §2. In a similar fashion to §2, the table documenting these acceptance criteria is in a simplified form. Every Acceptance Criterion has an Acceptance Criterion ID (ACID), which

is associated in the table below with its respective **SID**, development status, and the developers responsible for implementing it. Each **ACID** is linked to its respective subsection below for viewing the description of each criterion.

SID & Name	ACID	Status	Developer(s)
1	1	Qux	Bizz
2			

### 3.1 Criterion 1

ACID	Description
1	
1.0	

**Further Notes**

[**TEMPLATE, Remove UNNUMBERED prop**] Criterion N

ACID	Description
1	
1.0	

**Further Notes**

## 4 Implementation Tasks

This section summarizes the details of implementation tasks for the project. You will find in each subsection a table similar to those found in §2 and §3.

### 4.1 Summary of Production Code

SID & Name	ACID	Class Name(s)	Developer(s)	Status	Notes
1	2	Qux	Daz	Qud	Foo

#### 4.1.1 Class QUX

Class summary goes here.

Method	Notes
Bizz	blah blah blah

#### [TEMPLATE] Class FOOBAR

Class summary goes here.

Method	Notes
Qud	blah blah blah

### 4.2 Automated Test Code

SID & Name	ACID	Class Name(s)	Method Name(s)	Description	Status	Developer
1	2	Foo	Bar	Fizz	Buzz	Quz

### 4.3 Manual Test Code

SID & Name	ACID	Test Input	Test Oracle	Status	Notes	Developer(s)
1	2	Fizz	Fuzz	Quz	Bar	Qud

### 4.4 Other Manual Test Code

ID	Test Input	Expected Result	Class Name	Method Name of Test	Status	Developer
1	Foo	Bar	Fuzz	Quz	Fizz	Bazz



## 5 Meeting Minutes