Software Engineering Cowards – SPRINT 2 DELIVERABLE

Brian Ashworth (Scrum Master)– RedSoxFan

John Johnson – jdj20

Edwin Mellett – WynWinz

Aaron Tamenne – atamenne

https://github.com/RedSoxFan/2174\_CS1530\_SoftwareEngineeringCowards

(15 FEB 2017)

SPRINT 2 ACCOMPLISHMENTS

This sprint a lot of the main functionality of the game was completed. This includes aspects such as clicking to move pieces on the board, the white and black teams alternating turns, and starting a new game. Some additional functionality we included this sprint that wasn’t required was ensuring that only the king can occupy the center square and any of the four corners and the ability to save a game state (but not load yet, we will do that in the next sprint).

A disagreement that arose was that one team member was very for using tabs instead of spaces, but ultimately got outvoted and had to deal with using spaces instead. Other than that we continues to use the style checker to ensure that all of our code is similar in style and easy to read by everyone. We did not really have to resolve any disagreements other than deciding whether to use tabs or spaces for indentation.

We have not changed our development process at all since the first sprint; we continue to communicate via a group text for major decisions. If this method proves to be too much of a burden in the future we will probably switch to something like Slack for team communication. We have not really run into any major challenges with the coding/testing yet because most of the user stories implemented thus far have not been overly complex.

//TODO insert design patterns we used here or anything extra

USER STORIES

As a player

I want to be able to use the mouse to click and move the pieces

So that I can play Hnefatafl

As a player

I want to be able to save and load my Hnefatafl games

So that I can take a break in the middle of a game to resume later

As a player

I want to be able to start a new game or quit the game whenever I choose

So that I have more flexibility in what I do with the game

As a player

I want there to be two sides: a king with his defends and attackers

So that I can play Hnefatafl against someone

As a player

I want there to be 12 defenders and 24 attackers

So that the Copenhagen rules are obeyed

As a player

I want the board to be a grid with 11 rows and 11 columns of squares

So that the Copenhagen rules are obeyed

As a player

I want the initial board setup to be as follows (A:Attacker D:Defender K:King)

A A A A A

A

A D A

A D D D A

A A D D K D D A A

A D D D A

A D A

A

A A A A A

So that the Copenhagen rules are obeyed

As a player

I want the attacker's side to move first

So that the Copenhagen rules are obeyed

As a player

I want the game to progress with alternating moves between the two sides

So that the Copenhagen rules are obeyed

As a player

I want to be able to move a piece on my side any number of vacant squares along a row or column

So that the Copenhagen rules are obeyed

As a player

I want the central square to be the throne and can only be occupied by the king

So that the Copenhagen rules are obeyed

As a player

I want the king to be the only piece that can occupy any of the four corners

So that the Copenhagen rules are obeyed

As a player

I want the king and his defender's side to win if the king occupies any of the four corners

So that the Copenhagen rules are obeyed

As a player

I want any piece to be able to pass through the throne when it is empty

So that the Copenhagen rules are obeyed

As a player

I want the four corners to be hostile to any piece

So that the Copenhagen rules are obeyed

As a player

I want the throne to always be hostile to attackers

So that the Copenhagen rules are obeyed

As a player

I want the throne to be hostile to defenders if and only if it is empty

So that the Copenhagen rules are obeyed

As a player

I want any piece other than the king to be captured if there is an opposing piece and/or hostile square on each side of that piece along a column or row

So that the Copenhagen rules are obeyed

As a player

I want a row of pieces along any edge to be captured, excluding the king, if there is an enemy piece on both ends and an enemy piece adjacent to each piece parallel to the edge

(aka Shield Wall Capture)

So that the Copenhagen rules are obeyed

As a player

I want the king and his defender's side to win if the king is adjacent to an edge, can move, and it is impossible for the attacks to capture him after any number of moves

(aka Edge Fort)

So that the Copenhagen rules are obeyed

As a player

I want the attackers to win if there is an attacker's piece on all four adjacent sides of the king

So that the Copenhagen rules are obeyed

As a player

I want the attackers to win if there is an attacker's piece on three of the adjacent sides of the king and the throne is on the fourth adjacent side of the king

So that the Copenhagen rules are obeyed

As a player

I want the attackers to win if there is a barrier of attacker's pieces surrounding the king and all remaining defenders preventing the king from escaping

So that the Copenhagen rules are obeyed

As a player

I want a side to win if the opposing side moves a piece back and forth three times

So that the Copenhagen rules are obeyed

As a player

I want the attackers to win if the defenders repeat the same defending board position three times without capturing a piece if and only if the king and at least four other defending pieces are remaining

So that the Copenhagen rules are obeyed

As a player

I want a side to win if the opposing side cannot move

So that the Copenhagen rules are obeyed

As a player

I want the game to end in a draw if there is 50 consecutive moves without a capture

So that the Copenhagen rules are obeyed

DESCRIPTION OF DECISIONS

The first main decision we made was simply to use GitHub since we all possessed accounts prior to this assignment. We also researched the rules of Hnefatafl more in depth and practiced playing online/on an app so that we could fully understand the gameplay. Brian found an app for the game that he played on his phone and found a way to beat the AI on the hardest difficulty every time in just a few turns. If we try to implement an AI, it will be interesting to see if ours performs better than the one in the app.

We made a decision that instead of JButtons we would simply paint the JPanel so that the interface would look more professional. So far we have been doing most of our communication via a group text, but we discussed the possibility of using Slack. If future sprints require more in depth communication it is very likely we will find a more efficient and effective system of communication. As far as stand-ups go they were short and just done in the group chat since the level of difficulty for sprint 1 was low. As long as we are making progress, we will continue to hold our stand-ups virtually. If we decide it’s necessary to meet in person in the future for 1 or 2 of the stand-ups per sprint then we will try to find a time to meet that works for all of us.

Our next decision came down to coding style. We talked a bit about our own personal preferences and all agreed that the most important thing is for all of us to use the same coding style and to be consistent. John even added a style checker for us. We ended up adopting a slightly modified version of Google’s Java style. The modification being to allow star includes. The only thing one of our teammates was a little unhappy about was using camel case. He said, “Camel case is the devil” so I think we know how he feels, but he lost the battle on this one.