Software Engineering Cowards – SPRINT 6 DELIVERABLE

Brian Ashworth – RedSoxFan

John Johnson (Scrum Master) – jdj20

Edwin Mellett – WynWinz

Aaron Tamenne – atamenne

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

(17 APR 2017)

SPRINT 6 ACCOMPLISHMENTS

This sprint we completely finished implementing all of the Hnefatafl rules, and added our own additional functionality. We included an AI so that the user can play against the computer, and they can also choose which side they want the AI to be (attacking or defending). We continued to communicate just by meeting in class and using a group text. No meetings with the customer occurred during this sprint, as we were pretty sure we had a good grasp on what specific tasks we needed to accomplish and how to accomplish them.

We implemented AI as our signature functionality. The AI may potentially take a long time to calculate moves depending on the CPU of the user. We did not have enough hardware diversity to thoroughly test performance ranges. More often than not, the AI executed its moves in less than ten seconds.

Some problems arose when integrating the AI because it conflicted with calculating draw forts, but those problems were fixed with some inspection. We also had a bug from a previous sprint where we were not properly resetting the counter used to determine if fifty moves without a capture had occurred.

Other backlog tasks accomplished included adding tests that we lost points on for not including on previous sprints, and fixing the bug caused by loading corrupt or empty text files. We also completed a lot of refactoring of the code in order to make it more efficient and readable, and try to eliminate redundancies between classes in order to increase code quality.

USER STORIES COMPLETED

**Story Points: 16**

As a player

I want to be able to play against an AI as a defender or attacker

So that I can play by myself

**Story Points: 8**

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 Exit Fort)

So that the Copenhagen rules are obeyed

**Story Points: 8**

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

**Story Points: 8**

As a player

I want the attackers to win if the defenders repeat the defending board position three times while no piece is captured

So that the Copenhagen rules are obeyed

**Total Velocity: 40**

DEFECTS FOUND

**ID:** 1

**Title:** Crash on corrupted file load

**Description:** We were not properly sanitizing file loads.

**Reproduction Steps:**

1. Open the Hnefatafl program.
2. Click the Load Game button.
3. Load any invalid save file.

**Expected Results:** The user receives a message indicating a corrupt file.

**Observed Results:** The program crashes with a null pointer exception.

**Resolution:** A pull request was merged with the changes needed to sanitize inputs.

**ID:** 2

**Title:** Unexpected win after 50 moves

**Description:** There was a bug where the counter for the 50 moves without capture rule never got reset on a capture.

**Reproduction Steps:**

1. Open the Hnefatafl program.
2. Play a game that that consists of 50 moves (between the sides) that has at least one capture.
3. Make the 51st move (that is not a winning move according to the rules).

**Expected Results:** The move executes per normal game rules.

**Observed Results:** The game ends in a draw regardless of the move.

**Resolution:** A pull request was merged with the changes needed to update the counter appropriately.

**ID:** 3

**Title:** Win after first move in Human/AI mode

**Description:** There was an integration bug where the defenders would win after the first move due to a HashMap not being cloned properly.

**Reproduction Steps:**

1. Open the Hnefatafl program.
2. Choose Human/AI mode.
3. Make your first move as an attacker.

**Expected Results:** The move executes per normal game rules.

**Observed Results:** The game ends as a win for the defender.

**Resolution:** A pull request was merged with the changes needed to deep copy the HashMap.