CS 351 Project 2 Part 1 HER

Due September 18th

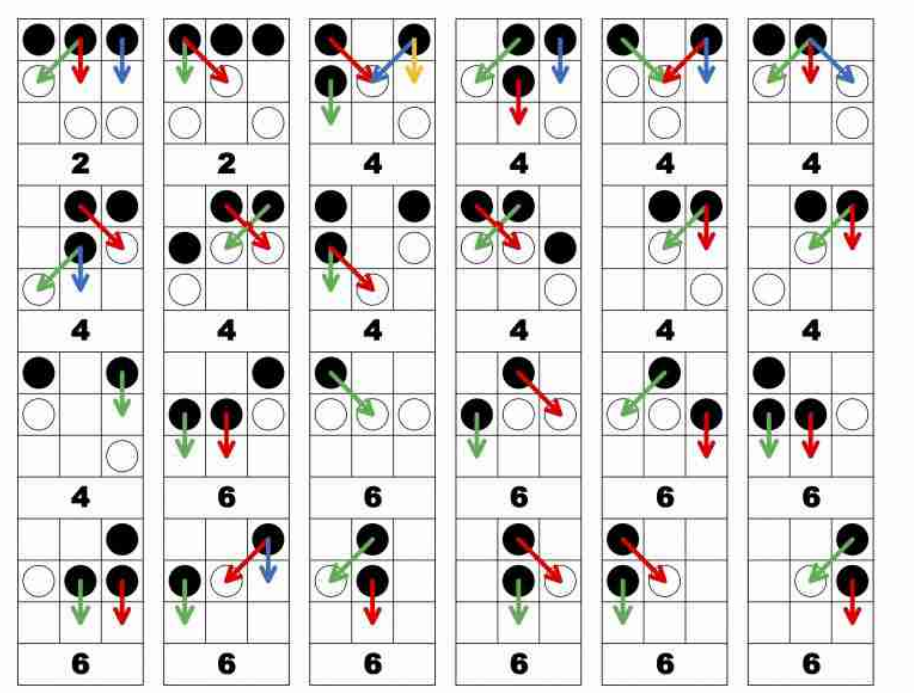

Figure 1: Hexapawn

Project Description

The mathematician Martin Gardner came up with a deceivingly simple game in 1958 that he called Hexapawn. His original paper can be found here. VSauce2 made a video which inspired me to make this project. I encourage you to both watch the video and read the paper before starting on this project.

In this project you will be tasked with a simulation similar to what VSauce2 did only using Java and JavaFX rather than doing it by hand. You will create a computer player which can eventually learn to play Hexapawn perfectly. The computer player will henceforth be known as the Hexapawn Educable Robot (HER).

HER is represented by matchboxes full of beads. On the cover of each matchbox is the board configuration it corresponds to. Each turn that HER takes consists of finding the corresponding matchbox then randomly choosing a bead from within that matchbox. Each bead represents one of the possible moves given the current board state. Each matchbook starts with one bead for each possible move. Each player starts with 3 pawns on their respective starting row. The human player always goes first. Players alternating moving one pawn at a time. Each pawn can either move forward (respective to their starting row) if their is no pawn currently occupying that space. Or it can move diagonally left or right (also only forward) but only if a pawn from the opposing team is currently occupying that space. Thus, you can generate all possible valid moves for any board state which is what is shown on the front of each matchbox. As the game progresses each matchbox and the bead that was chosen should be saved. A game is concluded when:

1. A player advances their pawn to the opponents starting row. That player wins.
2. A team has no pawns left. That team loses.
3. A player leaves the opposing player with no moves. The player that made the last move wins.

If the human player wins then HER must be punished for losing, this means removing the bead from HER's last move and putting all the other beads back into their matchboxes. However, if HER wins then you simply put all the beads back. If you encounter an empty matchbox then HER automatically loses and you remove the bead from HER's second to last move and put all the other beads back.

Your game must have 3 different modes:

1. Slow mode: In slow mode when it is the computers turn the matchbox (the visualization attached to the matchbox) that corresponds to the current board is shown on the screen. In addition the current probability of each arrow is shown. The user can then choose one of arrows to force the computer to make that move or allow the computer to randomly choose.

2. Fast mode: In fast mode the computer simply randomly picks a move for the current board state. Thus, the matchbox should not be shown.
3. Auto mode: In auto mode HER plays against a copy of itself for a given number of rounds. When the desired number of rounds has been played you should display how many rounds each HER won and the ratio of those wins which happened when they went first.

The desired mode should be configurable from within your GUI.

Project Requirements

- Displays board
- Human player can successfully move their pawns on their turn
- Invalid moves are rejected
- Detects when the game has concluded
- Properly punishes HER
- Slow mode works properly
- Fast mode works properly
- Auto mode works properly
- HER makes a move in no more than a quarter second