

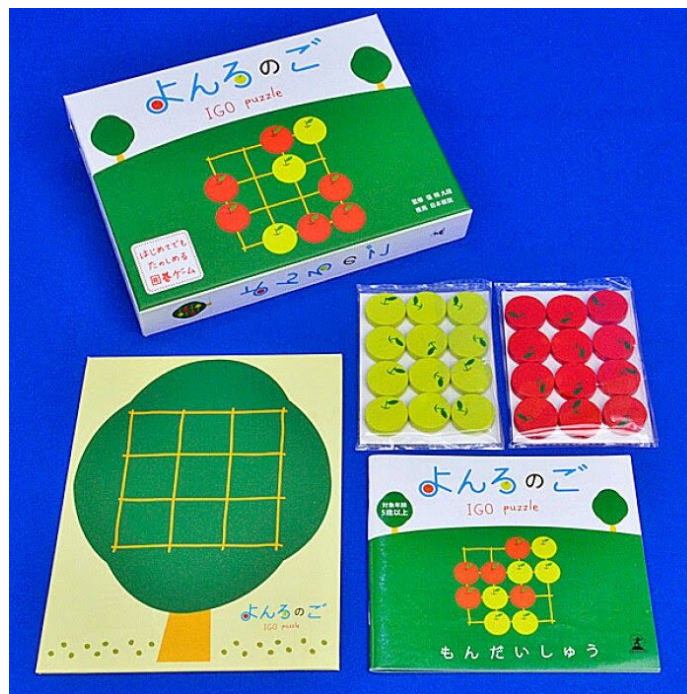
Fall 19 Project Summary: Stone Capture Go

This semester's project was based on the game Yonronogo (よんろのご), a Go-like game invented by pro Go player named Chang-Hsu to teach children the different rules of Go on a small 4x4 board game. The physical game contains a sizable booklet of puzzles that require understanding the rules of Go to solve. The aim of the puzzles is to gradually teach children how to make certain moves in Go that would help play the actual game on a 19x19 board. A regular game of Go can often be divided up into smaller, territorial disputes. The advantage of teaching children using a smaller board is that one, it is easier for them to comprehend, and two, it gives them a taste of how to deal with these smaller territorial disputes.

When the child learned enough, they could use the board to play a variant of Go called Stone Capture. The best way to describe this variant is that it is 50% normal Go, 20% Atari Go, and about 30% something else. Stone Capture can be played in two variants. One of which is just Atari Go, with sudden death capture. Another of which includes captures, but the win condition is that one player has dominated the board with a certain number of their own pieces. This number is agreed upon by the players before the game is played.

Game Contents

The game comes with a board, 16 green pieces, 16 red pieces, and a puzzle booklet. To be more child friendly, the game board is skinned to be a tree, and the pieces are skinned to look like apples.



Programming

A few problems arose when I was programming this game. The first of which deals with the concept of claim vs control. In normal Go, any game state can technically be a primitive game state. This is because the game only ends when both players pass on their turn. A player passes when they do not think they can invade any of the other player's controlled territory. The very nature of this end condition seems somewhat subjective, as it is hard to define vs control. A player can claim territory on a board, but it does not mean they control it as the other player might still be able to invade.

Another problem was that the code that was written was not efficient enough to run solve the game in a reasonable time.

Future Steps

The first step for the future is rebuilding the solver in a more efficient language such as C. Currently there is no data available for analysis. Using the results of solving this small 4x4 board, I hope to gain insight on the complexity, competitiveness, and overall measure of fun of this game scaled to bigger board sizes