

Squeeze-It Game

Project Description

Zhang Wei and Ahmad Aseeri

1 Introduction

The game is designed using Java language aiming to be more interactive and enjoyable to play with. The winner is the first one that achieves eliminating (squeezing) all 8 marbles of its opponent. Figure 1 illustrates the initial board configuration.

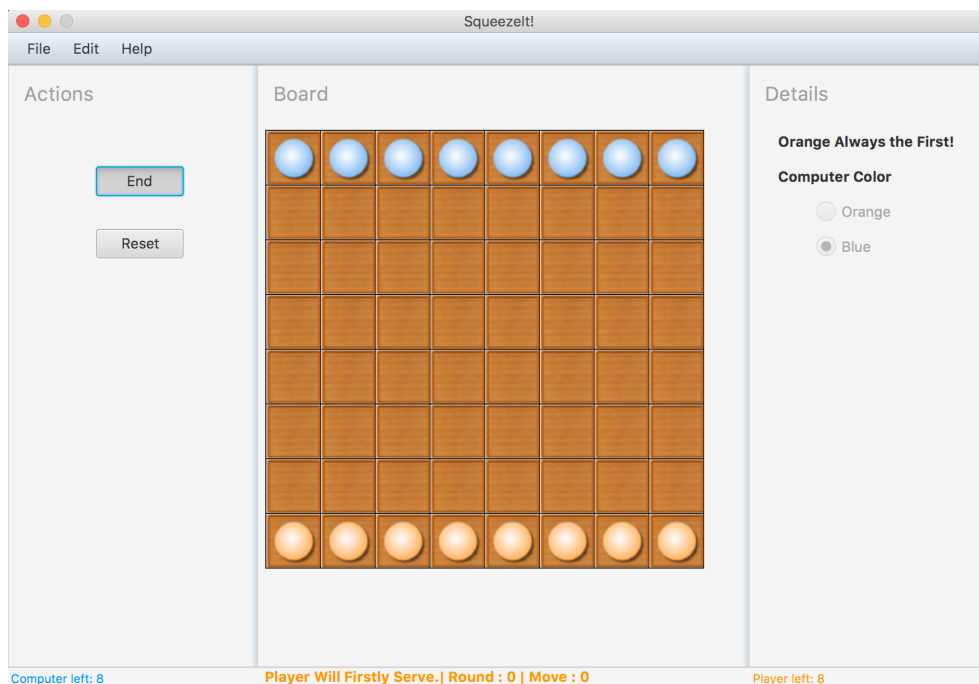


Figure 1: Squeeze it Interface!

2 How to play

There are two types of marbles: blue and orange. We make the orange marble to be the starting one and thus the user has the choice. Every player has 50 moves so it is important to manage every movement wisely. The program highlights the player's

possible movement whether vertically and horizontally. The reset of the game rounds should follow easily as the program will control any violated movement according to the game's rules. The figures 2 to 5 illustrate an example of the first round of this game.

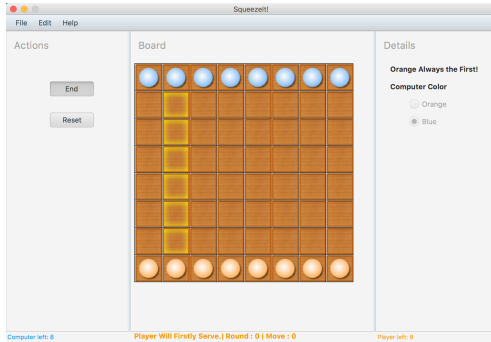


Figure 2: (a)

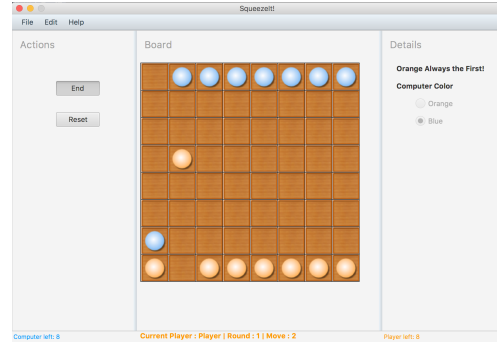


Figure 3: (b)

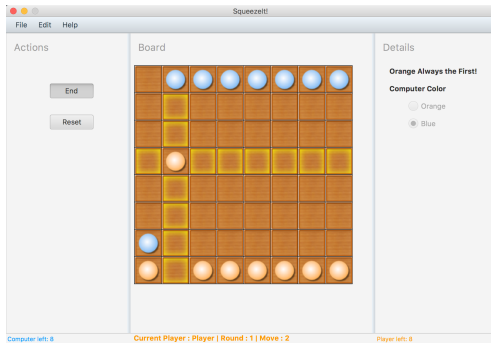


Figure 4: (c)

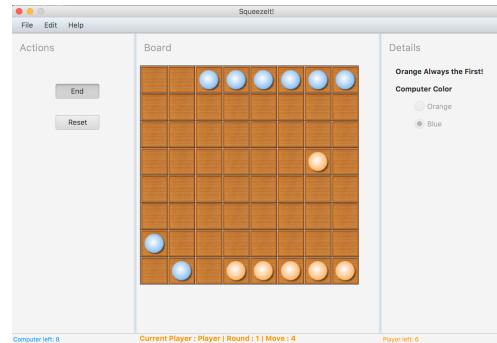


Figure 5: (d)

2.1 Pattern Types

The player's goal is to maximize his residual marbles and minimize its opponent's marbles on the current board configuration. The most valuable pattern is similar to what is illustrated in figure 6 below.

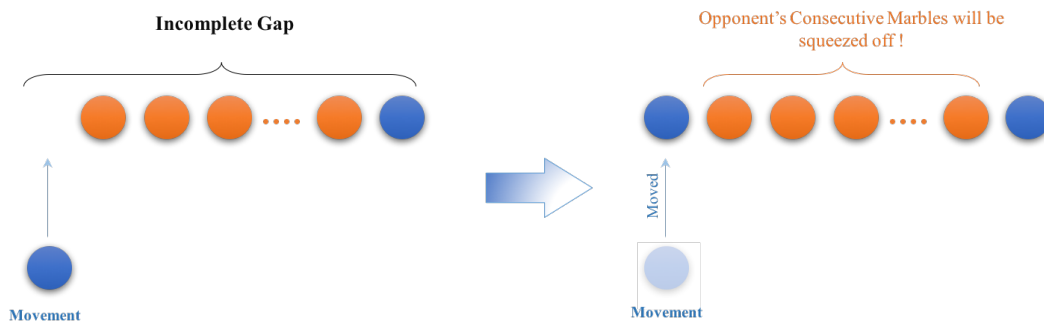


Figure 6: The pattern with the largest consecutive marbles is desired

The other possible pattern is when the player completes his consecutive marbles which then causes eliminating two marbles only, as illustrated in figure 7 below.

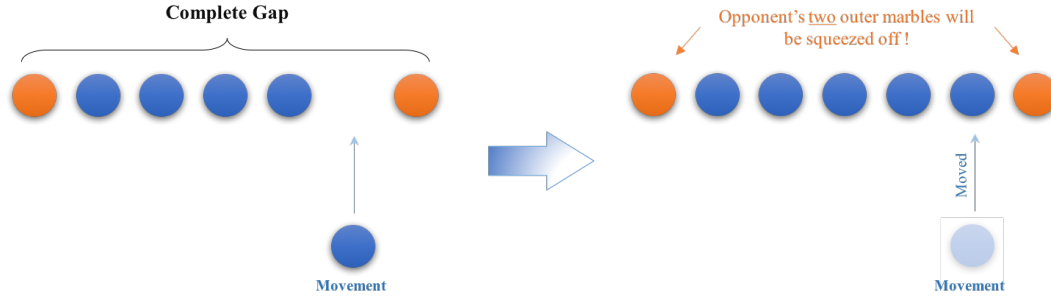


Figure 7: This type of pattern causes to squeeze off two marbles only.

2.2 Game Rules

The rules of this game can be summarized in the following three points:

- The basic rule is that each player can move either horizontally or vertically by any number of steps along to the highlighted path (possible moves).
- Whenever a pattern is formed, the player who most recently forms this pattern will eliminate the opponent's marbles appeared in this pattern (vertically or horizontally).
- Within the specified maximum number of moves, the player who has the most remaining marbles wins.

3 Game Heuristic

Based on the status of current board configuration, we introduce two types of heuristics:

1. Pattern-based Movement Generating Heuristic:

- (a) Start from each pattern on the current board configuration.
- (b) Find all possible movements which could fulfill any pattern on the current board.
- (c) Measure each movement based on the number of eliminated marbles that this movement will result in, and rank them based on the number of eliminated marbles that they can achieve. We call the movement with highest elimination score *the most aggressive movement*.
- (d) Evaluate each movement with Minimax $\alpha - \beta$ Pruning technique which will give a score to each movement based on **the difference between number of residual marbles of each player on the board**. So, the one which has higher Minimax score will be a better movement, since it will have

higher chance to maximize the number remaining marbles. Thus, we call the movement with maximum "minimax" evaluating score *the most defensive movement*.

For instance, the pattern "BOOOEB"¹ will be fulfilled by Orange player with only one possible movement, and the number of elimination would be 2 Blue marbles. Another pattern "OBBBBE" will be fulfilled by Orange player such that 4 Blue marbles will be eliminated. In this case, for player Orange the movement which fulfill the pattern "OBBBBE" will be much more valuable than the one which fulfill the pattern "BOOOEB", and so on. However, if the movement which will fulfill the pattern "OBBBBE" has a lower Minimax score, the value of this movement will be diminished.

2. Global Movement Generating Heuristic:

- (i) Find out every possible movement on the current board configuration.
- (ii) Collect those movements that create a pattern in any type (complete gap or incomplete gap).
- (iii) Sort them according to their pattern score (The function for calculating the score will be discussed in 3.1)
- (iv) Therefore, find the movements that give us the pattern with the maximum eliminating score (it will be the best movements).
- (v) Among all best movements, we carry out the minimax $\alpha - \beta$ pruning on each of them to see which gives a higher value, i.e. *defensive*.

3.1 Elimination Rating

During the *Global Movement Generating* process, in order to rank each movement, we need to follow certain criteria. Here is the criteria we use to rate each movement, based on the value of the patterns that those movements can create.

1. **Valid Removal Count R_v :** Express the number of opponent's marbles in the pattern. For example, in a pattern "OBBBBEO", if the opponent's color is Orange then $R_v = 4$, but for player Blue, this value would be 2.
2. **Length L :** The length of the pattern. Based on the pattern above which is "OBBBBEO", we have $L = 7$, for pattern "BOOE", $L = 4$.
3. **Threatening Score $T_c = \frac{R_v}{L}$**
 - Gap (such as "OBBBBEO"): For player Orange, it's $\frac{4}{7}$, but for player Blue, it's $\frac{2}{7}$
 - Incomplete Gap (such as "BOOOE"): For player Orange, it's $\frac{1}{5}$, but for player Blue, it's $\frac{3}{5}$
 - FulFilled Gap (such as "BOOOEB"): For player Orange, it's $\frac{2}{5}$, but for player Blue, it's $\frac{3}{6}$

¹B implies Blue, O implies Orange, and E implies Empty cell.

4. Global-Eliminating Score G_c :

This score is evaluated by Minimax $\alpha - \beta$ Pruning. The default search depth is 3 and default search width is 12. Whenever the number of the human player's residual marbles becomes less than 4, the search depth will slightly increase and the search width will increase by 2.

4 Source Code

You can visit our source code repository by clicking [here](#).