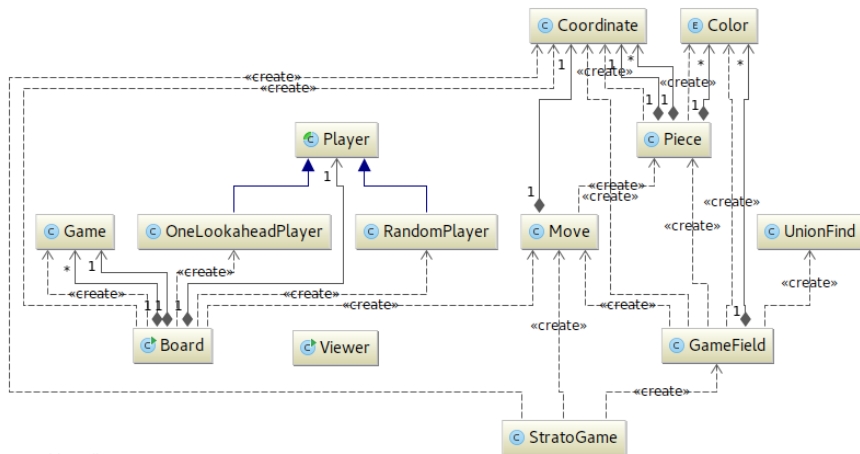


A Java Implementation of Stratopolis

Yuxi Liu

19 October, 2016

UML (1 minute)



GameField class (2 minute)

The GameField class represents the game board. It can decide whether a piece can be played on the board, and it can score the board.

A Gamefield has three matrices, representing color, height, and index.

$$\begin{bmatrix} B & B & B & B & B & B \\ B & B & R & B & B & B \\ B & B & G & B & B & B \\ B & B & B & B & B & B \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 0 & -1 & -1 & -1 \\ -1 & -1 & 0 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix}$$



GameField class (2 minute)

The GameField class represents the game board. It can decide whether a piece can be played on the board, and it can score the board.

A Gamefield has three matrices, representing color, height, and index.

$$\begin{bmatrix} B & B & G & B & B & B \\ B & B & R & G & B & B \\ B & B & G & B & B & B \\ B & B & B & B & B & B \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} -1 & -1 & 1 & 1 & -1 & -1 \\ -1 & -1 & 0 & 1 & -1 & -1 \\ -1 & -1 & 0 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix}$$



GameField class (2 minute)

The GameField class represents the game board. It can decide whether a piece can be played on the board, and it can score the board.

A Gamefield has three matrices, representing color, height, and index.

$$\begin{bmatrix} B & B & G & B & B & B \\ B & R & R & G & B & B \\ B & B & G & B & B & B \\ B & B & B & B & B & B \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} -1 & -1 & 1 & 1 & -1 & -1 \\ -1 & 2 & 0 & 1 & -1 & -1 \\ 2 & 2 & 0 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix}$$



GameField class (2 minute)

The GameField class represents the game board. It can decide whether a piece can be played on the board, and it can score the board.

A Gamefield has three matrices, representing color, height, and index.

$$\begin{bmatrix} B & B & G & B & B & B \\ B & R & R & G & R & B \\ B & B & G & G & G & B \\ B & B & B & B & B & B \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} -1 & -1 & 1 & 1 & -1 & -1 \\ -1 & 2 & 0 & 1 & 3 & -1 \\ 2 & 2 & 0 & 3 & 3 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix}$$



GameField class (2 minute)

The GameField class represents the game board. It can decide whether a piece can be played on the board, and it can score the board.

A Gamefield has three matrices, representing color, height, and index.

$$\begin{bmatrix} R & B & G & B & B & B \\ R & R & R & G & R & B \\ B & B & G & G & G & B \\ B & B & B & B & B & B \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 4 & 4 & 1 & 1 & -1 & -1 \\ 4 & 2 & 0 & 1 & 3 & -1 \\ 2 & 2 & 0 & 3 & 3 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix}$$



GameField class (2 minute)

The GameField class represents the game board. It can decide whether a piece can be played on the board, and it can score the board.

A Gamefield has three matrices, representing color, height, and index.

$$\begin{bmatrix} R & B & G & B & B & B \\ R & R & R & G & R & B \\ B & B & G & G & G & G \\ B & B & B & B & G & G \end{bmatrix}
 \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}
 \begin{bmatrix} 4 & 4 & 1 & 1 & -1 & -1 \\ 4 & 2 & 0 & 1 & 3 & -1 \\ 2 & 2 & 0 & 3 & 3 & 5 \\ -1 & -1 & -1 & -1 & 5 & 5 \end{bmatrix}$$



GameField class (2 minute)

The GameField class represents the game board. It can decide whether a piece can be played on the board, and it can score the board.

A Gamefield has three matrices, representing color, height, and index.

$$\begin{bmatrix} R & R & B & B & B & B \\ R & R & R & G & R & B \\ B & B & G & G & G & G \\ B & B & B & B & G & G \end{bmatrix} \begin{bmatrix} 1 & 2 & 2 & 1 & 0 & 0 \\ 1 & 1 & 2 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 4 & 6 & 6 & 1 & -1 & -1 \\ 4 & 2 & 6 & 1 & 3 & -1 \\ 2 & 2 & 0 & 3 & 3 & 5 \\ -1 & -1 & -1 & -1 & 5 & 5 \end{bmatrix}$$



GameField class (2 minute)

The GameField class represents the game board. It can decide whether a piece can be played on the board, and it can score the board.

A Gamefield has three matrices, representing color, height, and index.

$$\begin{bmatrix} R & R & B & B & B & B \\ R & R & R & G & R & G \\ B & B & G & G & G & G \\ B & B & B & B & G & G \end{bmatrix} \begin{bmatrix} 1 & 2 & 2 & 1 & 1 & 1 \\ 1 & 1 & 2 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 4 & 6 & 6 & 1 & 7 & 7 \\ 4 & 2 & 6 & 1 & 3 & 7 \\ 2 & 2 & 0 & 3 & 3 & 5 \\ -1 & -1 & -1 & -1 & 5 & 5 \end{bmatrix}$$



GameField class (2 minute)

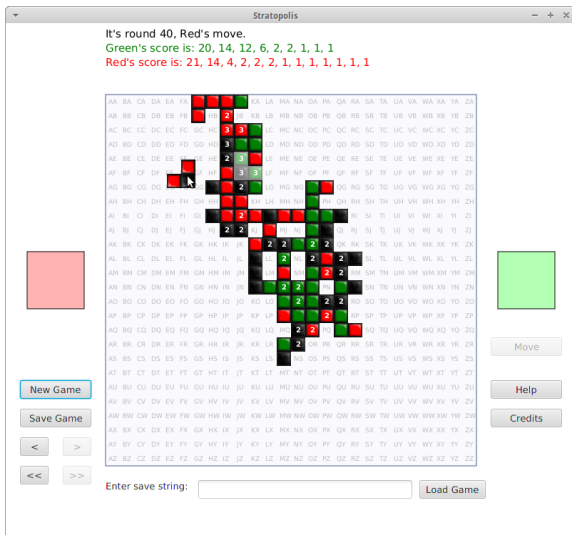
The GameField class represents the game board. It can decide whether a piece can be played on the board, and it can score the board.

A Gamefield has three matrices, representing color, height, and index.

$$\begin{bmatrix} R & R & B & B & B & B \\ R & R & R & G & R & G \\ R & R & G & G & G & G \\ B & B & B & B & G & G \end{bmatrix} \begin{bmatrix} 1 & 2 & 2 & 1 & 1 & 1 \\ 2 & 1 & 2 & 1 & 1 & 1 \\ 2 & 2 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 4 & 6 & 6 & 1 & 7 & 7 \\ 8 & 2 & 6 & 1 & 3 & 7 \\ 8 & 8 & 0 & 3 & 3 & 5 \\ -1 & -1 & -1 & -1 & 5 & 5 \end{bmatrix}$$

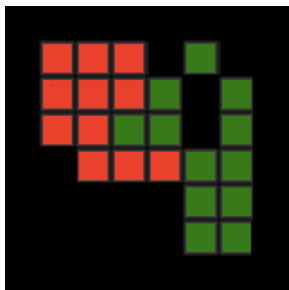


GUI screenshot (10 seconds)

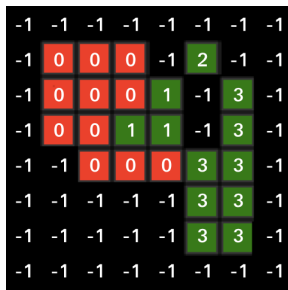


Connected Component Labelling (1 minutes)

To score the board, we must solve the algorithmically interesting problem: how to find out the connected color regions of a board?



(a) Input: the color matrix



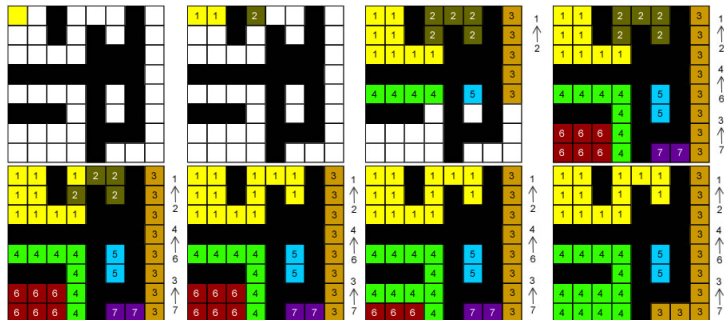
(b) Output: the component label matrix.

Connected Component Labelling. The background gets label -1.

2 Pass Union-find CCL Algorithm (20 seconds)

We used a two pass algorithm that uses an implementation of the set ADT, called “union-find”.

Solves the problem in linear space and nearly in linear time.



Pictures copied from <http://aishack.in/tutorials/labelling-connected-components-example/>

Search “union find connected components labelling” for details.

Base64 encoding (20 seconds)

We implemented game saving and loading by converting the game state into a single savestring, which can later be loaded to recreate the game state.

<u>Game history</u>	<u>Red's stack of pieces</u>	<u>Green's stack of pieces</u>
MMUANNPAOMHDMKRAKAA,	EIABFCIDBDGJGHCJFE,	TLMQNNRSKOSKPLQTMO

Base64 encoding

TU1VQU50UEFPtUhetUtSQutLQUESRUlBQkZDSURCRedKR0hDSkZFLFRMTVFOTlJTS09TS1BMUVRNTw==

The savestring is encoded in Base64 to discourage cheating.

Demonstration (4 minutes)

Green is human, red is OneLookaheadPlayer.

Demonstrate error warning, move highlighting, forward and backward stepping, savestring, help screen, and credit screen.

Q&A (1 minute)

Any questions?