Flume
Programação em Lógica

Carolina Moreira - up201303494
Maria Teresa Ferreira - up201603811

October 21, 2018

1 Game Description

1.1 History

In 2010, Mark Steere designed Flume.

1.2 Rules

This is a two-player game, in which each player has different color pieces.
The board is a square with odd number positions, meaning there can not be ties.
The players take turns placing pieces of their own color on the empty board places.
If, when a player makes a move, the piece placed is adjacent to three or four other pieces (regardless of color), this player can play again until this does not happen.
The game ends when the board is full and the winner is the one who has more of their own pieces placed.

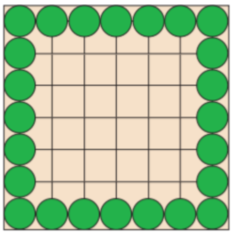

Figure 1: Image Representation of Initial Board

2 Game State Internal Representation

2.1 Initial Board

```
initial_board(Board) :-
        Board = [
                [green, green, green, green, green, green, green],
                [green, blank, blank, blank, blank, blank, green],
                [green, blank, blank, blank, blank, blank, green],
                [green, blank, blank, blank, blank, blank, green],
                [green, blank, blank, blank, blank, blank, green],
                [green, blank, blank, blank, blank, blank, green],
                [green, green, green, green, green, green, green]].
```

Listing 1: Source Code for the Internal Representation of the Initial Board

2.1.1 Intermediate Board

```
example_board1(Board) :-
        Board = [
                [green, green, green, green, green, green, green],
                [green, blank, blank, blank, blank, blank, green],
                [green, blank, blank, blank, blank, blank, green],
                [green, blank, blank, blank, blue, red, green],
                [green, blank, blank, red, blank, blank, green],
                [green, blank, blank, blue, blue, red, green],
                [green, green, green, green, green, green, green]].
```

Listing 2: Source Code for the Internal Representation of an Example Intermediate Board

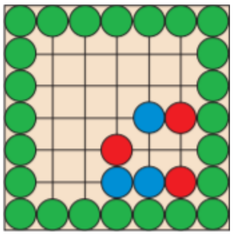

Figure 2: Image Representation of Example Intermediate Board

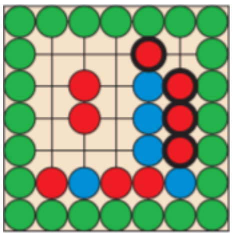

Figure 3: Image Representation of Example Winning Condition Board

2.2 Winning Condition Board

```
example_board2(Board) :-
        Board = [
                [green, green, green, green, green, green, green],
                [green, blank, blank, blank, red, blank, green],
                [green, blank, red, blank, blue, red, green],
                [green, blank, red, blank, blue, red, green],
                [green, blank, blank, blank, blue, red, green],
                [green, red, blue, red, red, blue, green],
                [green, green, green, green, green, green, green]].
```

Listing 3: Source Code for the Internal Representation of an Example Winning Condition Board

3 Text Visualization of Board

```
print_board([]) :-
        print_separator.
print_board(Board) :-
        Board = [H|T],
        print_separator,
        print_line(H),
        print_board(T).

print_separator :-
                print('*-*-*-*-*-*-*-*\n').

print_line([]) :-
        print('|\n').

print_line(Line) :-
        Line = [H|T],
        print('|'),
        print_piece(H),
        print_line(T).

print_piece(Piece) :-
        piece_text(Piece, Text),
        print(Text).

piece_text(green, 'G').
piece_text(red, 'R').
piece_text(blue, 'B').
piece_text(blank, ' ').
```

Listing 4: Source Code for the Text Representation

```
*-*-*-*-*-*-*-*
|G|G|G|G|G|G|G|
*-*-*-*-*-*-*-*
|G| | | | | |G|
*-*-*-*-*-*-*-*
|G| | | | | |G|
*-*-*-*-*-*-*-*
|G| | | | | |G|
*-*-*-*-*-*-*-*
|G| | | | | |G|
*-*-*-*-*-*-*-*
|G| | | | | |G|
*-*-*-*-*-*-*-*
|G|G|G|G|G|G|G|
*-*-*-*-*-*-*-*
```

Listing 5: Text Representation of the Initial Board