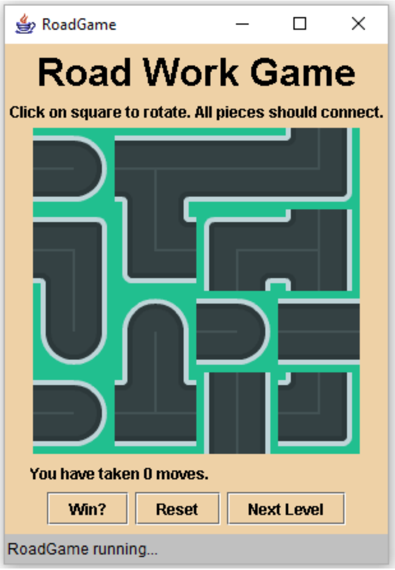
Road Work Game

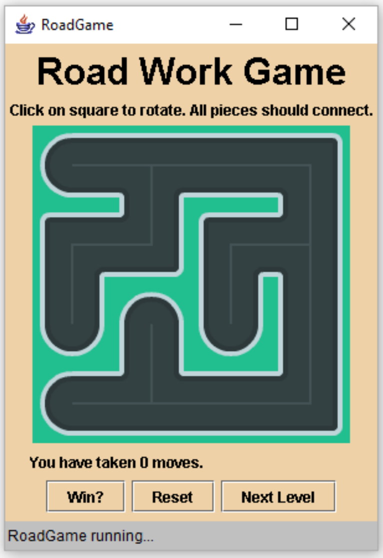


Road Work Game

Click on square to rotate. All pieces should connect.

You have taken 0 moves.

RoadGame running...



Road Work Game

Click on square to rotate. All pieces should connect.

You have taken 0 moves.

RoadGame running...

In this game, you rotate the squares until you can make a road that is connected. Then you win.

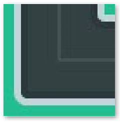
To build the board, you need to record TWO things. The direction of the piece AND which rotation it is in.

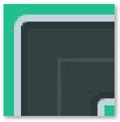
In the dir array, the letters stand for:
C = curved
E = end
S = straight
T = Three joins

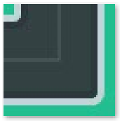
The numbers on the end have to do with the number of rotations.

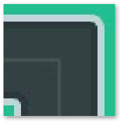
This creates TWO arrays:

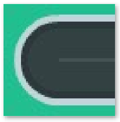
```
        int b[] [] = {{3, 2, 2, 4},
                      {3, 1, 2, 3},
                      {4, 2, 3, 2},
                      {3, 4, 1, 4}};
        char dir[] [] = {{'e', 't', 's', 'c'},
                      {'c', 'c', 'c', 't'},
                      {'e', 'e', 'e', 's'},
                      {'e', 't', 's', 'c'}};
```

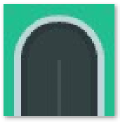

c1.png

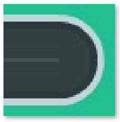

c2.png

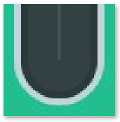

c3.png

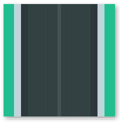

c4.png

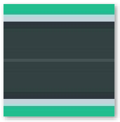

e1.png

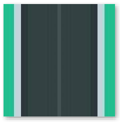

e2.png

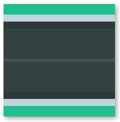

e3.png

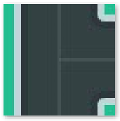

e4.png

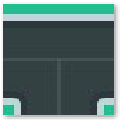

s1.png

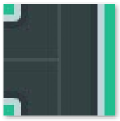

s2.png

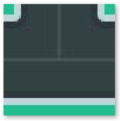

s3.png


s4.png


t1.png


t2.png


t3.png

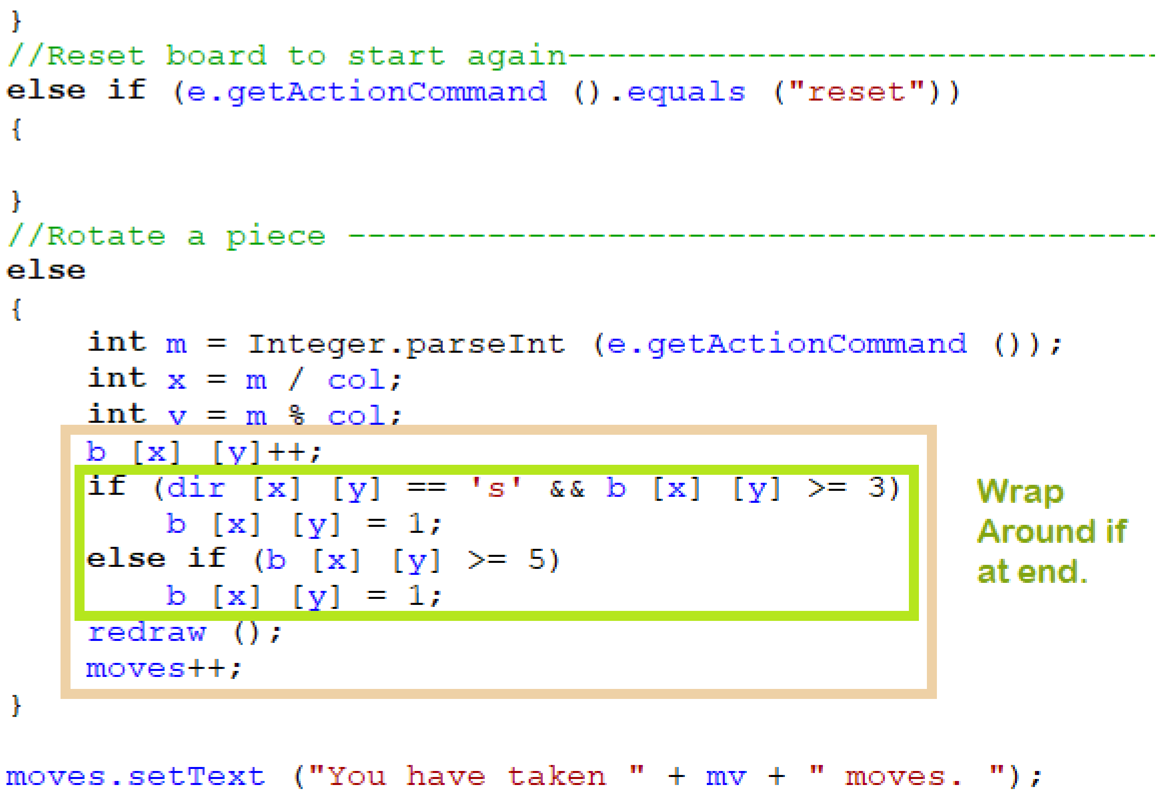

t4.png

These come together to make the picture name:

```
public void redraw ()
{
    int m = 0;
    for (int i = 0 ; i < row ; i++)
    {
        for (int j = 0 ; j < col ; j++)
        {
            pics [m].setIcon (createImageIcon (dir [i] [j] + "" + b [i] [j] + picFileType));
            m++;
        }
    }
}
```

Step 1 Rotate

- To rotate a piece, you simply need to increase the spot in the b array that matches the piece.
- If you have gone over 2 if it is a straight piece, wrap around to 1. If you have gone over 4 for the other pieces, wrap around to 1.
- Then call redraw to update the screen.

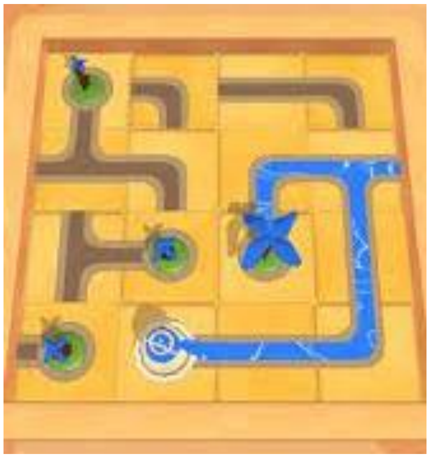


```
}
//Reset board to start again-------------------------------
else if (e.getActionCommand ().equals ("reset"))
{

}
//Rotate a piece -----------------------------------------
else
{
    int m = Integer.parseInt (e.getActionCommand ());
    int x = m / col;
    int y = m % col;
    b [x] [y]++;
    if (dir [x] [y] == 's' && b [x] [y] >= 3)
        b [x] [y] = 1;
    else if (b [x] [y] >= 5)
        b [x] [y] = 1;
    redraw ();
    moves++;
}

moves.setText ("You have taken " + mv + " moves. ");
```

Wrap Around if at end.

Step 2 Levels

- Add one new level.
- Look at the scribble maze for ideas about how to make a new level.
- The Road Work game is adapted from the Water Connect Puzzle. This is a picture of another level that you can add. Alternatively, you can make up your own. I'd recommend paper in that case.



Step 3 Reset

- Allow the user to reset the game back to the initial state.
- This is closely related to the level code.
- http://www.gorskicompsci.ca/ICS3U/Unit6/ResetButton.pdf has some hints but it doesn't perfectly match this program. You will need to think about how to implement it.

Step 4 Win

- Allow the user to see if they have won.
- This has some hints as to how to do this: http://www.gorskicompsci.ca/ICS3U/Unit6/CheckAnswer.pdf. Once again, this doesn't perfectly match this program. You will need to think about how to calculate a win.

Starter Code

```
import javax.swing.*;
import java.applet.*;
import java.awt.event.*;
import java.awt.*;
public class RoadGame extends Applet implements ActionListener
{
    int row = 4;
    int col = 4;
    //to store current settings on screen
    int b[] [] = {{3, 2, 2, 4},
            {3, 1, 2, 3},
            {4, 2, 3, 2},
            {3, 4, 1, 4}};
    char dir[] [] = {{'e', 't', 's', 'c'},
            {'c', 'c', 'c', 't'},
            {'e', 'e', 'e', 's'},
            {'e', 't', 's', 'c'}};
    int win[] [] = {{1, 2, 2, 3},
            {2, 4, 2, 3},
            {4, 2, 4, 1},
            {1, 4, 2, 4}};

    //copy of level 1
    int b1[] [] = {{3, 2, 2, 4},
            {3, 1, 2, 3},
            {4, 2, 3, 2},
            {3, 4, 1, 4}};
    char dir1[] [] = {{'e', 't', 's', 'c'},
            {'c', 'c', 'c', 't'},
            {'e', 'e', 'e', 's'},
            {'e', 't', 's', 'c'}};
    int win1[] [] = {{1, 2, 2, 3},
            {2, 4, 2, 3},
            {4, 2, 4, 1},
            {1, 4, 2, 4}};

    JButton pics[] = new JButton [row * col];
    int sqDimension = 64;
    String picFileType = ".png";

    int mv = 0;
    JLabel moves;
    int lvl = 1;

    public void init ()
    {
        resize (300, 385);
        setBackground (new Color (238, 209, 166));
        JLabel title = new JLabel ("Road Work Game");
        title.setFont (new Font ("Arial", Font.BOLD, 30));
        add (title);
        JLabel ins = new JLabel ("Click on square to rotate. All pieces should connect.");
        add (ins);
        Panel grid = new Panel (new GridLayout (row, col));
```

```java
        int m = 0;
        for (int i = 0 ; i < row ; i++)
        {
            for (int j = 0 ; j < col ; j++)
            {
                pics [m] = new JButton (createImageIcon (dir [i] [j] + "" + b [i] [j] + picFileType));
                pics [m].setPreferredSize (new Dimension (sqDimension, sqDimension));
                pics [m].addActionListener (this);
                pics [m].setBorder (null);
                pics [m].setActionCommand ("" + m);
                grid.add (pics [m]);
                m++;
            }
        }
        add (grid);

        moves = new JLabel ("You have taken " + mv + " moves. ");
        moves.setPreferredSize (new Dimension (260, 20));
        add (moves);
        JButton win = new JButton ("Win?");
        win.setActionCommand ("win");
        win.addActionListener (this);
        win.setBackground (new Color (238, 209, 166));
        add (win);
        JButton reset = new JButton ("Reset");
        reset.setActionCommand ("reset");
        reset.addActionListener (this);
        reset.setBackground (new Color (238, 209, 166));
        add (reset);
        JButton next = new JButton ("Next Level");
        next.setActionCommand ("next");
        next.addActionListener (this);
        next.setBackground (new Color (238, 209, 166));
        add (next);
    }

    public void copyOver (int a[] [], int c[] [])
    {
        for (int i = 0 ; i < row ; i++)
        {
            for (int j = 0 ; j < col ; j++)
            {
                a [i] [j] = c [i] [j];
            }
        }
        mv = 0;
        redraw ();
        showStatus ("New Level begins.");
    }

    public void redraw ()
    {
        int m = 0;
        for (int i = 0 ; i < row ; i++)
        {
            for (int j = 0 ; j < col ; j++)
            {
                pics [m].setIcon (createImageIcon (dir [i] [j] + "" + b [i] [j] + picFileType));
                m++;
            }
        }
    }

    public void actionPerformed (ActionEvent e)
    { //Moves to next level -----------------------------------------
        if (e.getActionCommand ().equals ("next"))
        {

        }
        //Checks if user wins ----------------------------------------
        else if (e.getActionCommand ().equals ("win"))
        {

        }
        //Reset board to start again----------------------------------
        else if (e.getActionCommand ().equals ("reset"))
        {

        }
        //Rotate a piece ---------------------------------------------
        else
        {
            int m = Integer.parseInt (e.getActionCommand ());
            int x = m / col;
            int y = m % col;


        }
        moves.setText ("You have taken " + mv + " moves. ");
    }


    protected static ImageIcon createImageIcon (String path)
    {
        java.net.URL imgURL = RoadGame.class.getResource (path);
        if (imgURL != null)
            return new ImageIcon (imgURL);
        else
            return null;
    }
}
```