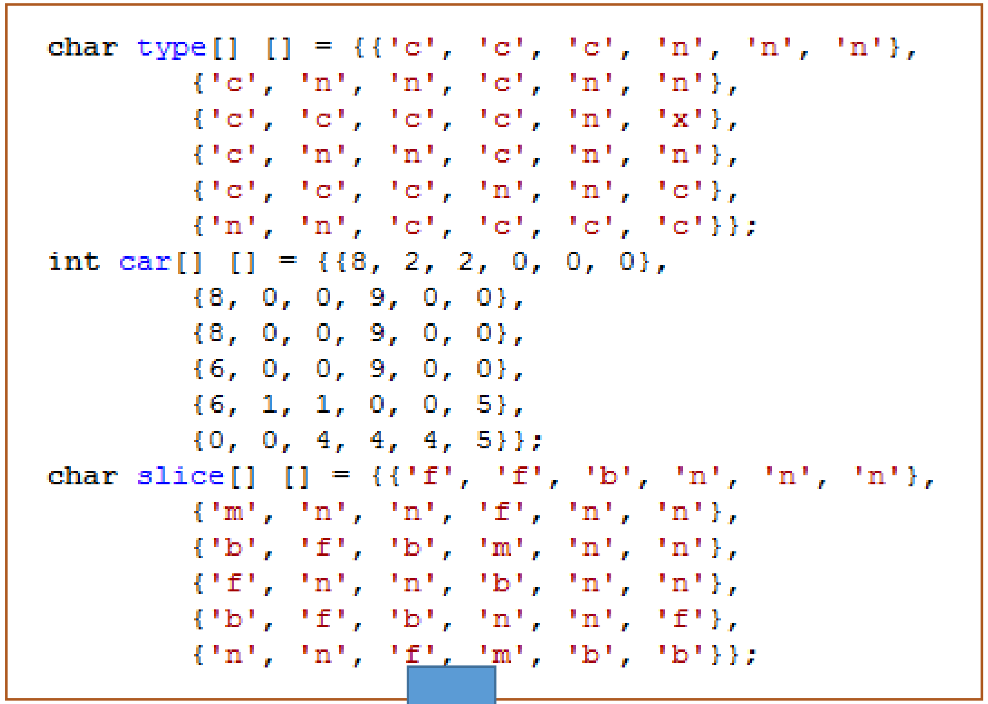
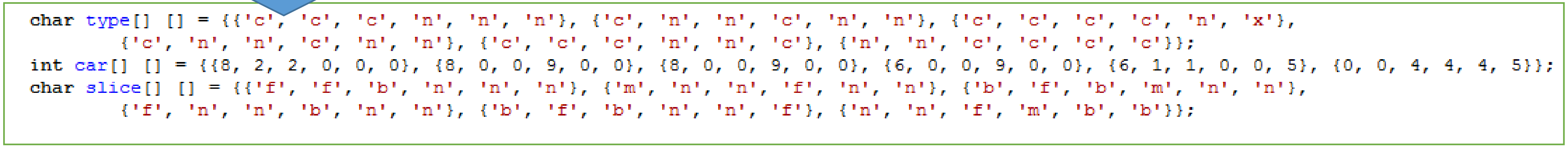
Rush Hour

Adding Levels & Reset



```
char type[] [] = {{'c', 'c', 'c', 'n', 'n', 'n'},
        {'c', 'n', 'n', 'c', 'n', 'n'},
        {'c', 'c', 'c', 'c', 'n', 'x'},
        {'c', 'n', 'n', 'c', 'n', 'n'},
        {'c', 'c', 'c', 'n', 'n', 'c'},
        {'n', 'n', 'c', 'c', 'c', 'c'}};
int car[] [] = {{8, 2, 2, 0, 0, 0},
        {8, 0, 0, 9, 0, 0},
        {8, 0, 0, 9, 0, 0},
        {6, 0, 0, 9, 0, 0},
        {6, 1, 1, 0, 0, 5},
        {0, 0, 4, 4, 4, 5}};
char slice[] [] = {{'f', 'f', 'b', 'n', 'n', 'n'},
        {'m', 'n', 'n', 'f', 'n', 'n'},
        {'b', 'f', 'b', 'm', 'n', 'n'},
        {'f', 'n', 'n', 'b', 'n', 'n'},
        {'b', 'f', 'b', 'n', 'n', 'f'},
        {'n', 'n', 'f', 'm', 'b', 'b'}};
```

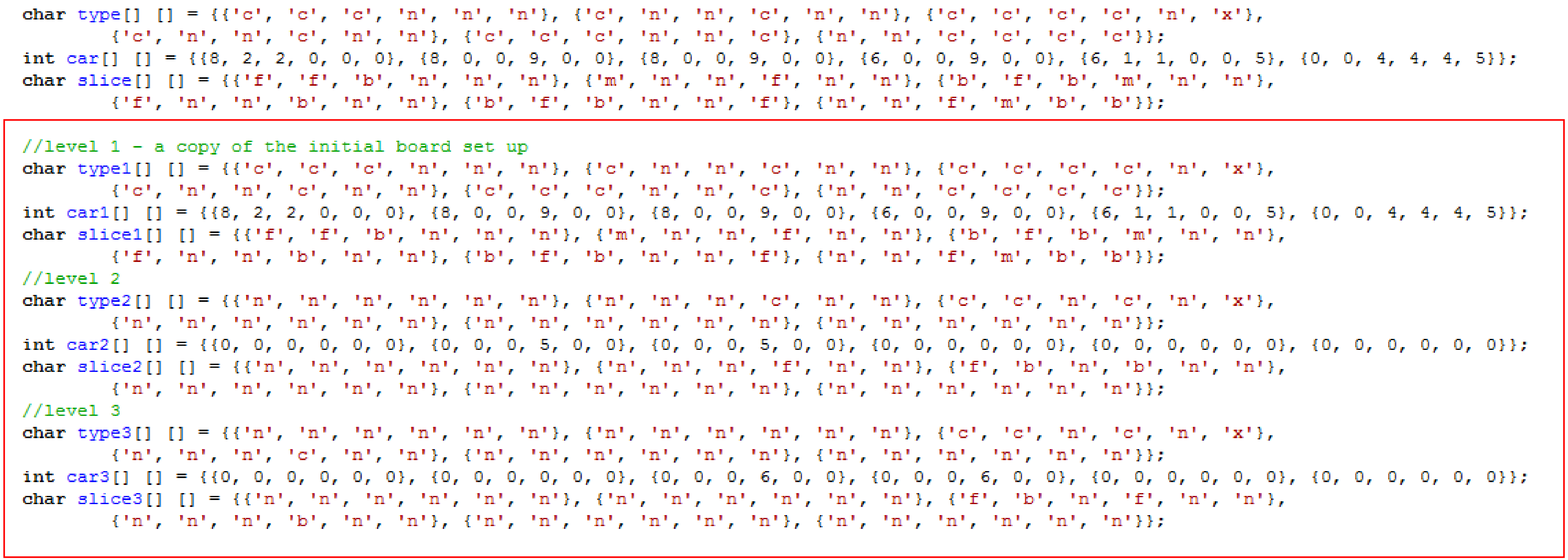
After I have created and tested the level, I convert it to a more compact form.



```
char type[] [] = {{'c', 'c', 'c', 'n', 'n', 'n'}, {'c', 'n', 'n', 'c', 'n', 'n'}, {'c', 'c', 'c', 'c', 'n', 'x'},
        {'c', 'n', 'n', 'c', 'n', 'n'}, {'c', 'c', 'c', 'n', 'n', 'c'}, {'n', 'n', 'c', 'c', 'c', 'c'}};
int car[] [] = {{8, 2, 2, 0, 0, 0}, {8, 0, 0, 9, 0, 0}, {8, 0, 0, 9, 0, 0}, {6, 0, 0, 9, 0, 0}, {6, 1, 1, 0, 0, 5}, {0, 0, 4, 4, 4, 5}};
char slice[] [] = {{'f', 'f', 'b', 'n', 'n', 'n'}, {'m', 'n', 'n', 'f', 'n', 'n'}, {'b', 'f', 'b', 'm', 'n', 'n'},
        {'f', 'n', 'n', 'b', 'n', 'n'}, {'b', 'f', 'b', 'n', 'n', 'f'}, {'n', 'n', 'f', 'm', 'b', 'b'}};
```

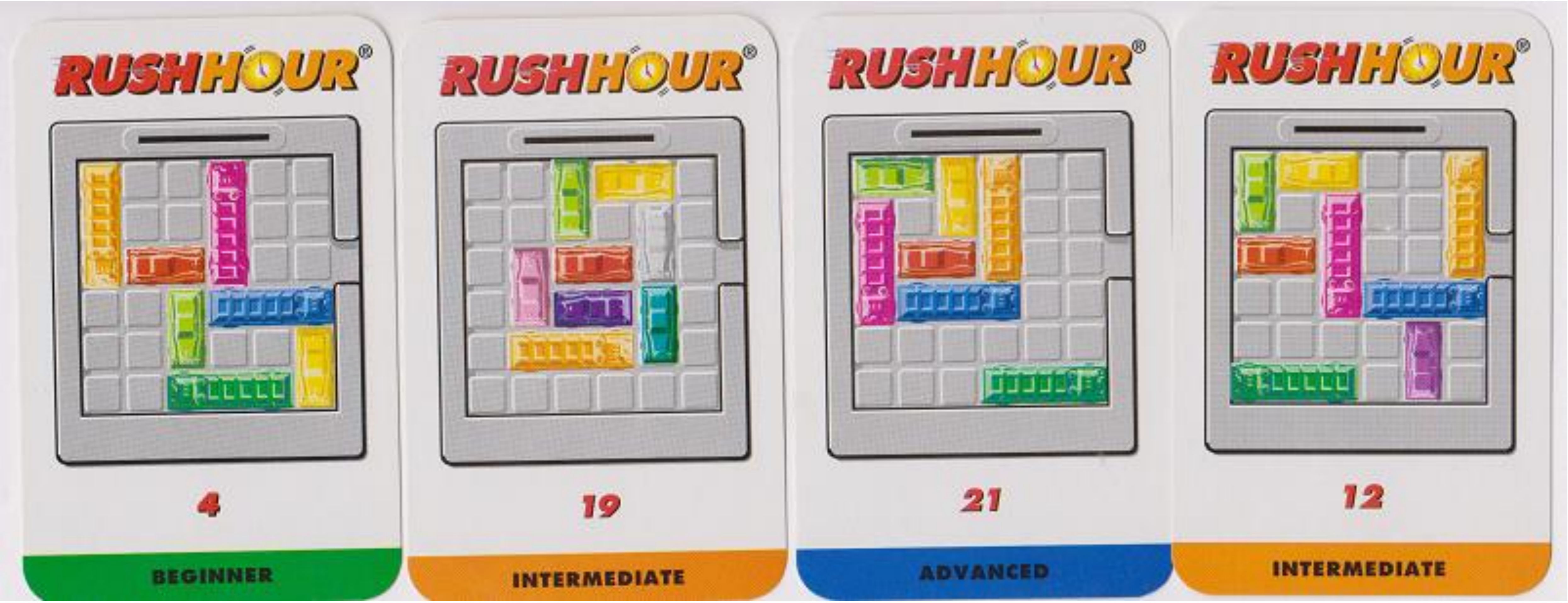
I created levels and stored them in a series of arrays

Notice that level 1 and the initial board are the same!!!

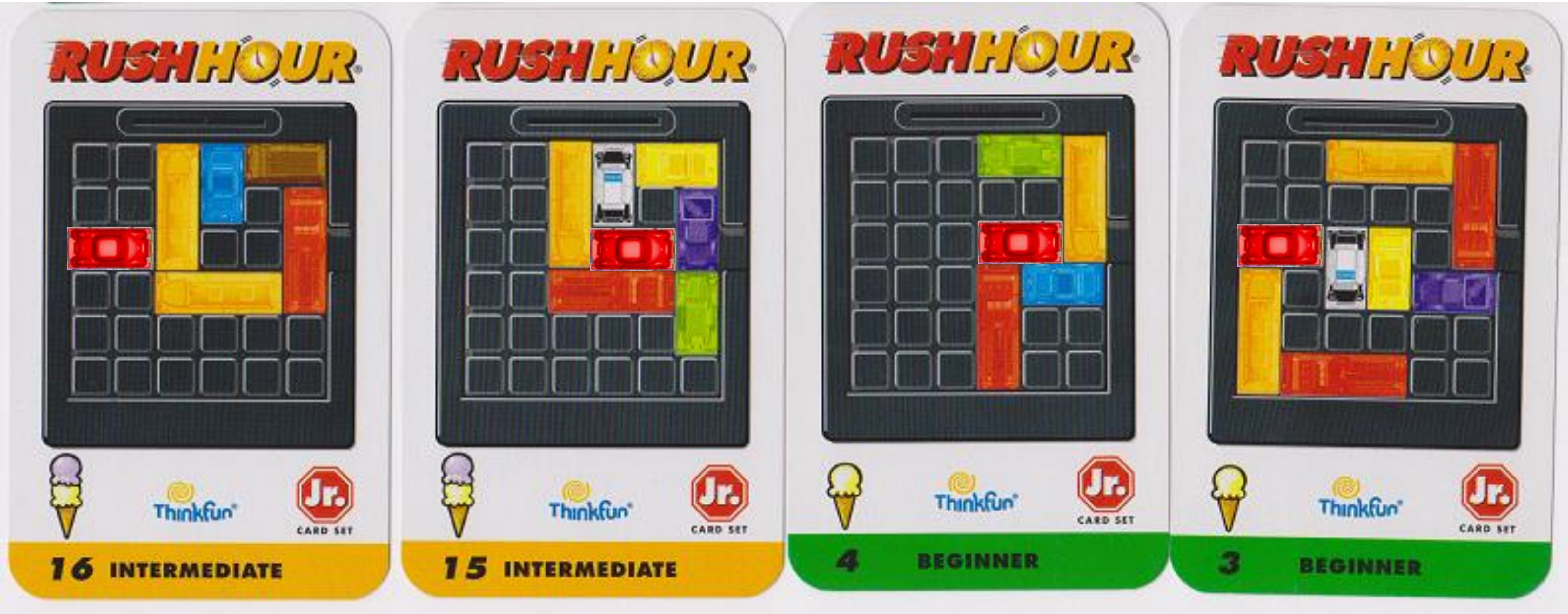


```
char type[] [] = {{'c', 'c', 'c', 'n', 'n', 'n'}, {'c', 'n', 'n', 'c', 'n', 'n'}, {'c', 'c', 'c', 'c', 'n', 'x'},
        {'c', 'n', 'n', 'c', 'n', 'n'}, {'c', 'c', 'c', 'n', 'n', 'c'}, {'n', 'n', 'c', 'c', 'c', 'c'}};
int car[] [] = {{8, 2, 2, 0, 0, 0}, {8, 0, 0, 9, 0, 0}, {8, 0, 0, 9, 0, 0}, {6, 0, 0, 9, 0, 0}, {6, 1, 1, 0, 0, 5}, {0, 0, 4, 4, 4, 5}};
char slice[] [] = {{'f', 'f', 'b', 'n', 'n', 'n'}, {'m', 'n', 'n', 'f', 'n', 'n'}, {'b', 'f', 'b', 'm', 'n', 'n'},
        {'f', 'n', 'n', 'b', 'n', 'n'}, {'b', 'f', 'b', 'n', 'n', 'f'}, {'n', 'n', 'f', 'm', 'b', 'b'}};

//level 1 - a copy of the initial board set up
char type1[] [] = {{'c', 'c', 'c', 'n', 'n', 'n'}, {'c', 'n', 'n', 'c', 'n', 'n'}, {'c', 'c', 'c', 'c', 'n', 'x'},
        {'c', 'n', 'n', 'c', 'n', 'n'}, {'c', 'c', 'c', 'n', 'n', 'c'}, {'n', 'n', 'c', 'c', 'c', 'c'}};
int car1[] [] = {{8, 2, 2, 0, 0, 0}, {8, 0, 0, 9, 0, 0}, {8, 0, 0, 9, 0, 0}, {6, 0, 0, 9, 0, 0}, {6, 1, 1, 0, 0, 5}, {0, 0, 4, 4, 4, 5}};
char slice1[] [] = {{'f', 'f', 'b', 'n', 'n', 'n'}, {'m', 'n', 'n', 'f', 'n', 'n'}, {'b', 'f', 'b', 'm', 'n', 'n'},
        {'f', 'n', 'n', 'b', 'n', 'n'}, {'b', 'f', 'b', 'n', 'n', 'f'}, {'n', 'n', 'f', 'm', 'b', 'b'}};
//level 2
char type2[] [] = {{'n', 'n', 'n', 'n', 'n', 'n'}, {'n', 'n', 'n', 'c', 'n', 'n'}, {'c', 'c', 'n', 'c', 'n', 'x'},
        {'n', 'n', 'n', 'n', 'n', 'n'}, {'n', 'n', 'n', 'n', 'n', 'n'}, {'n', 'n', 'n', 'n', 'n', 'n'}};
int car2[] [] = {{0, 0, 0, 0, 0, 0}, {0, 0, 0, 5, 0, 0}, {0, 0, 0, 5, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}};
char slice2[] [] = {{'n', 'n', 'n', 'n', 'n', 'n'}, {'n', 'n', 'n', 'f', 'n', 'n'}, {'f', 'b', 'n', 'b', 'n', 'n'},
        {'n', 'n', 'n', 'n', 'n', 'n'}, {'n', 'n', 'n', 'n', 'n', 'n'}, {'n', 'n', 'n', 'n', 'n', 'n'}};
//level 3
char type3[] [] = {{'n', 'n', 'n', 'n', 'n', 'n'}, {'n', 'n', 'n', 'n', 'n', 'n'}, {'c', 'c', 'n', 'c', 'n', 'x'},
        {'n', 'n', 'n', 'c', 'n', 'n'}, {'n', 'n', 'n', 'n', 'n', 'n'}, {'n', 'n', 'n', 'n', 'n', 'n'}};
int car3[] [] = {{0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 6, 0, 0}, {0, 0, 0, 6, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}};
char slice3[] [] = {{'n', 'n', 'n', 'n', 'n', 'n'}, {'n', 'n', 'n', 'n', 'n', 'n'}, {'f', 'b', 'n', 'f', 'n', 'n'},
        {'n', 'n', 'n', 'b', 'n', 'n'}, {'n', 'n', 'n', 'n', 'n', 'n'}, {'n', 'n', 'n', 'n', 'n', 'n'}};
```

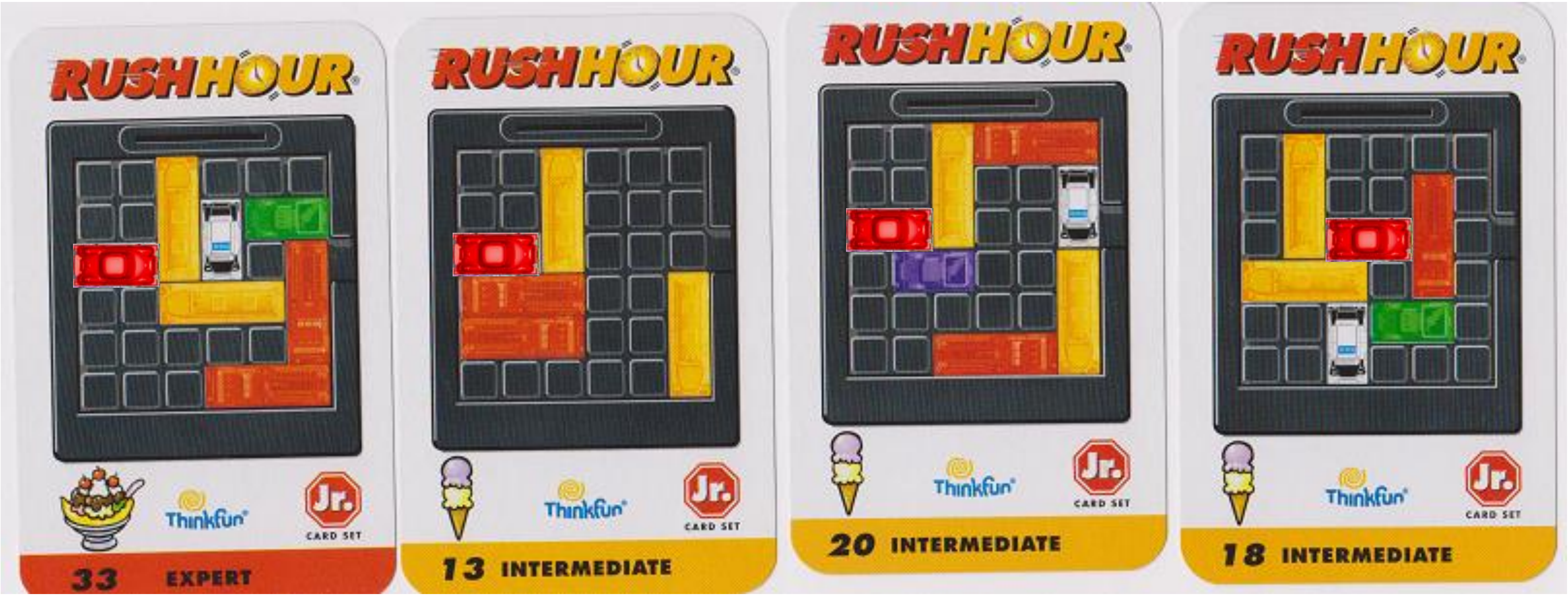
Some example levels:



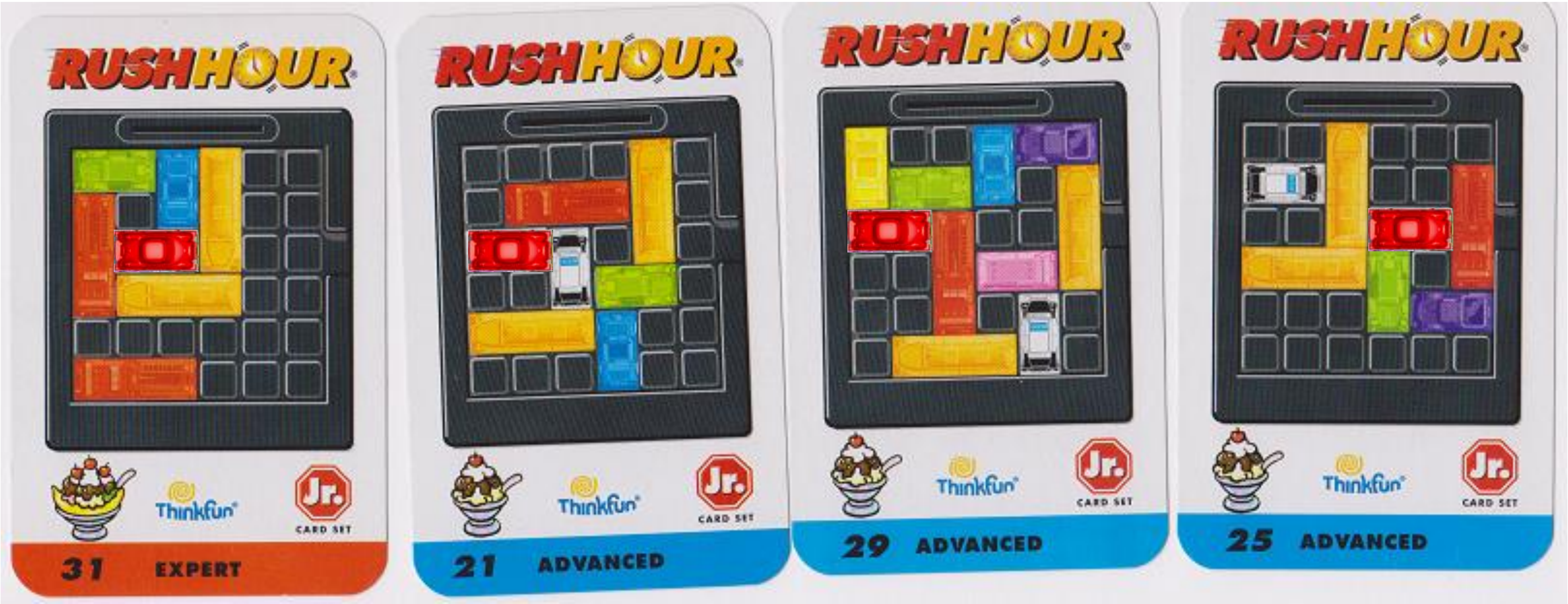
More example levels:



Even more example levels:

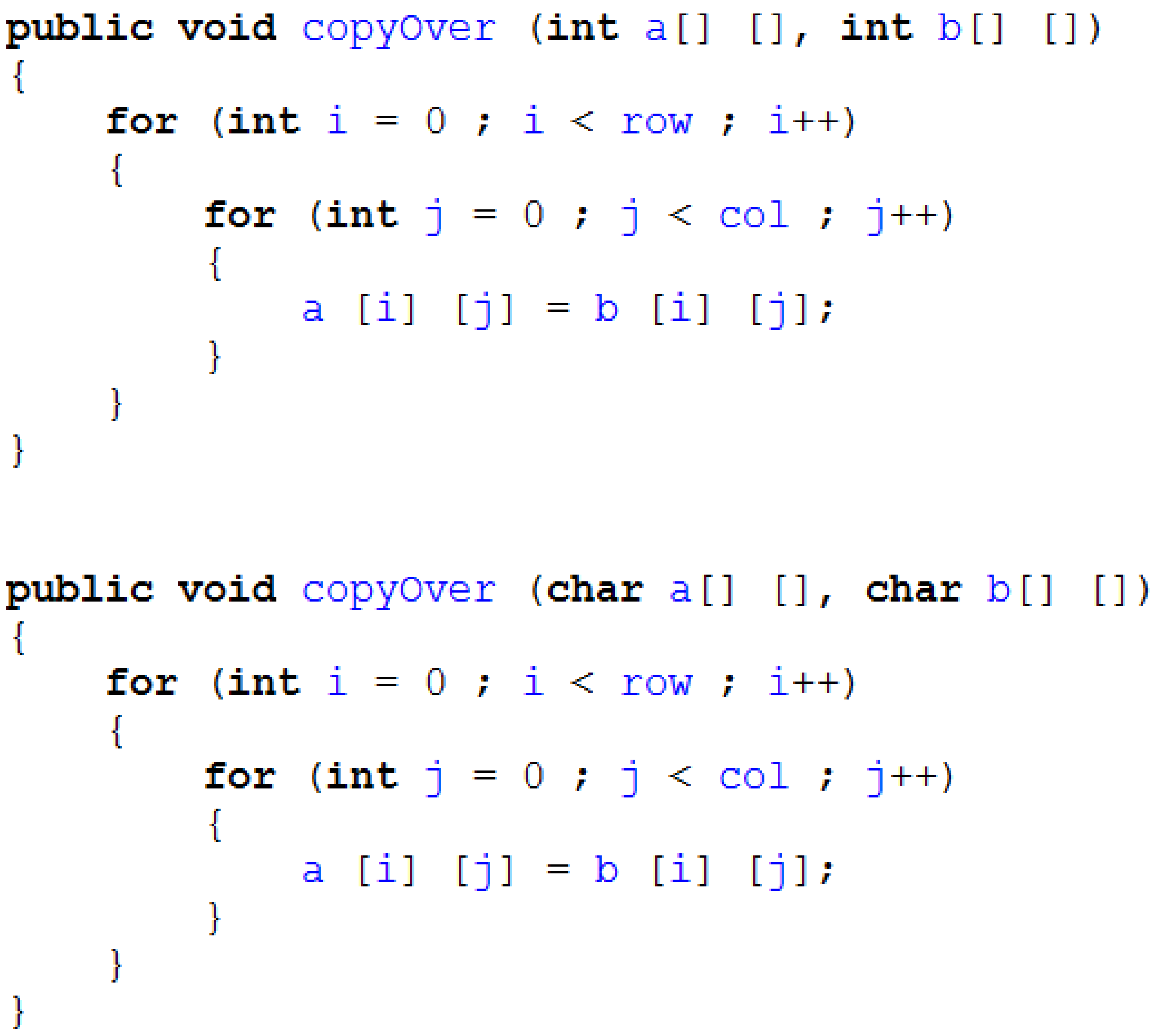


Still more example levels:



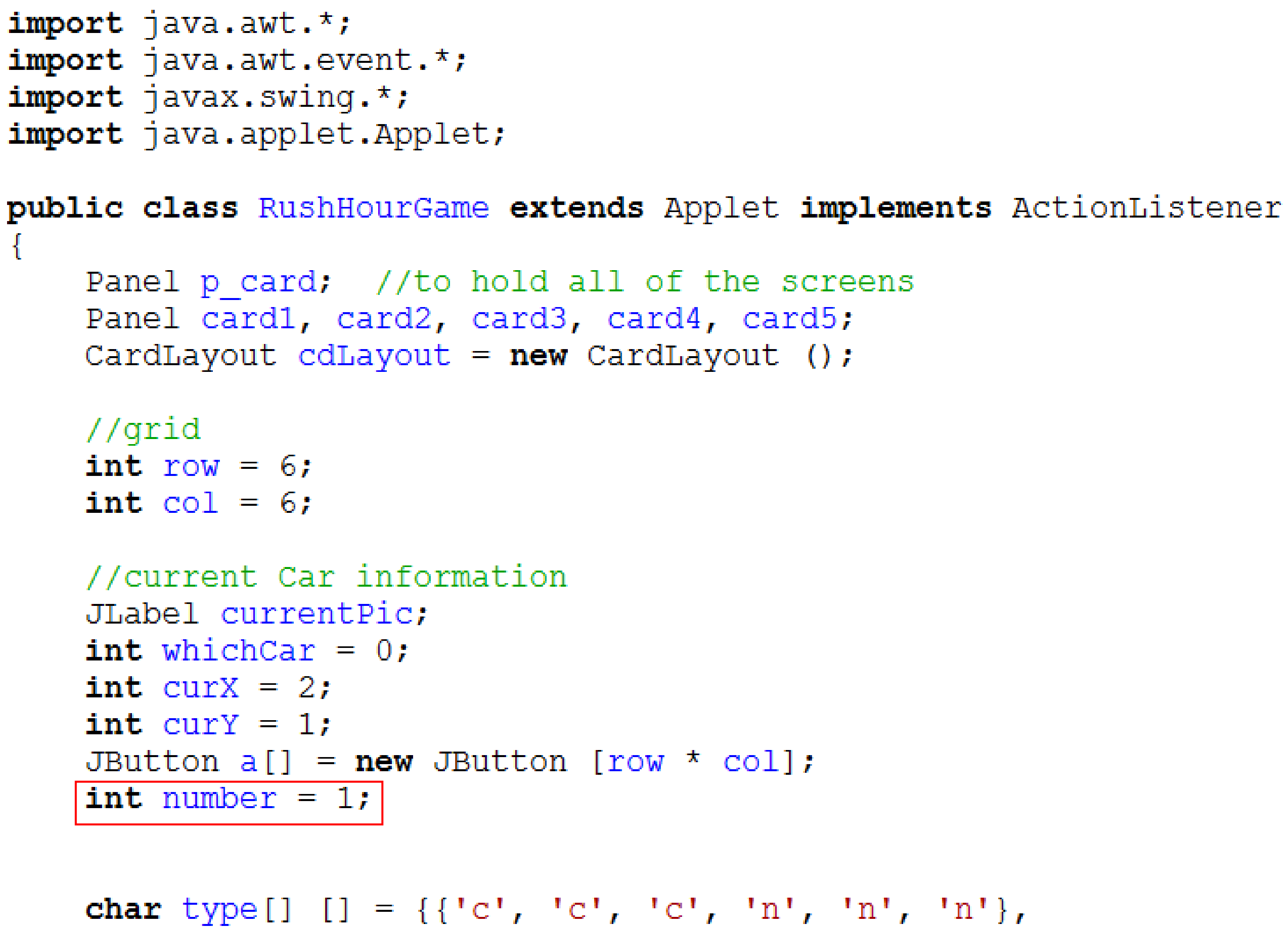
TEST your levels completely before moving on.

Add these two methods to your program.



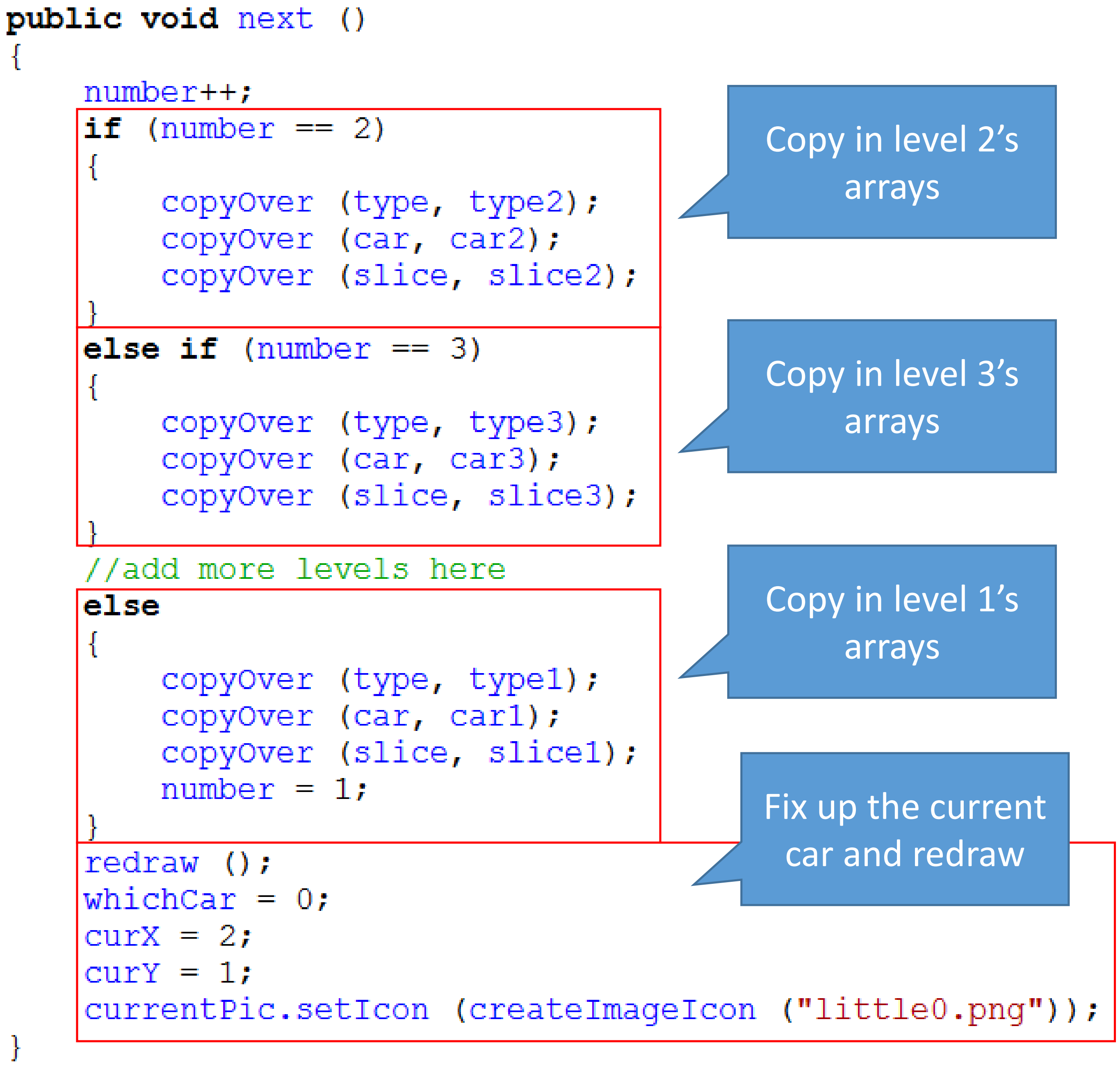
```
public void copyOver (int a[] [], int b[] [])
{
    for (int i = 0 ; i < row ; i++)
    {
        for (int j = 0 ; j < col ; j++)
        {
            a [i] [j] = b [i] [j];
        }
    }
}


public void copyOver (char a[] [], char b[] [])
{
    for (int i = 0 ; i < row ; i++)
    {
        for (int j = 0 ; j < col ; j++)
        {
            a [i] [j] = b [i] [j];
        }
    }
}
```

Add a global variable number to the top of your code



```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.applet.Applet;

public class RushHourGame extends Applet implements ActionListener
{
    Panel p_card;  //to hold all of the screens
    Panel card1, card2, card3, card4, card5;
    CardLayout cdLayout = new CardLayout ();

    //grid
    int row = 6;
    int col = 6;

    //current Car information
    JLabel currentPic;
    int whichCar = 0;
    int curX = 2;
    int curY = 1;
    JButton a[] = new JButton [row * col];
    int number = 1;


    char type[] [] = {{'c', 'c', 'c', 'n', 'n', 'n'},
```

The method to move between levels



```
public void next ()
{
    number++;
    if (number == 2)
    {
        copyOver (type, type2);
        copyOver (car, car2);
        copyOver (slice, slice2);
    }
    else if (number == 3)
    {
        copyOver (type, type3);
        copyOver (car, car3);
        copyOver (slice, slice3);
    }
    //add more levels here
    else
    {
        copyOver (type, type1);
        copyOver (car, car1);
        copyOver (slice, slice1);
        number = 1;
    }
    redraw ();
    whichCar = 0;
    curX = 2;
    curY = 1;
    currentPic.setIcon (createImageIcon ("little0.png"));
}
```

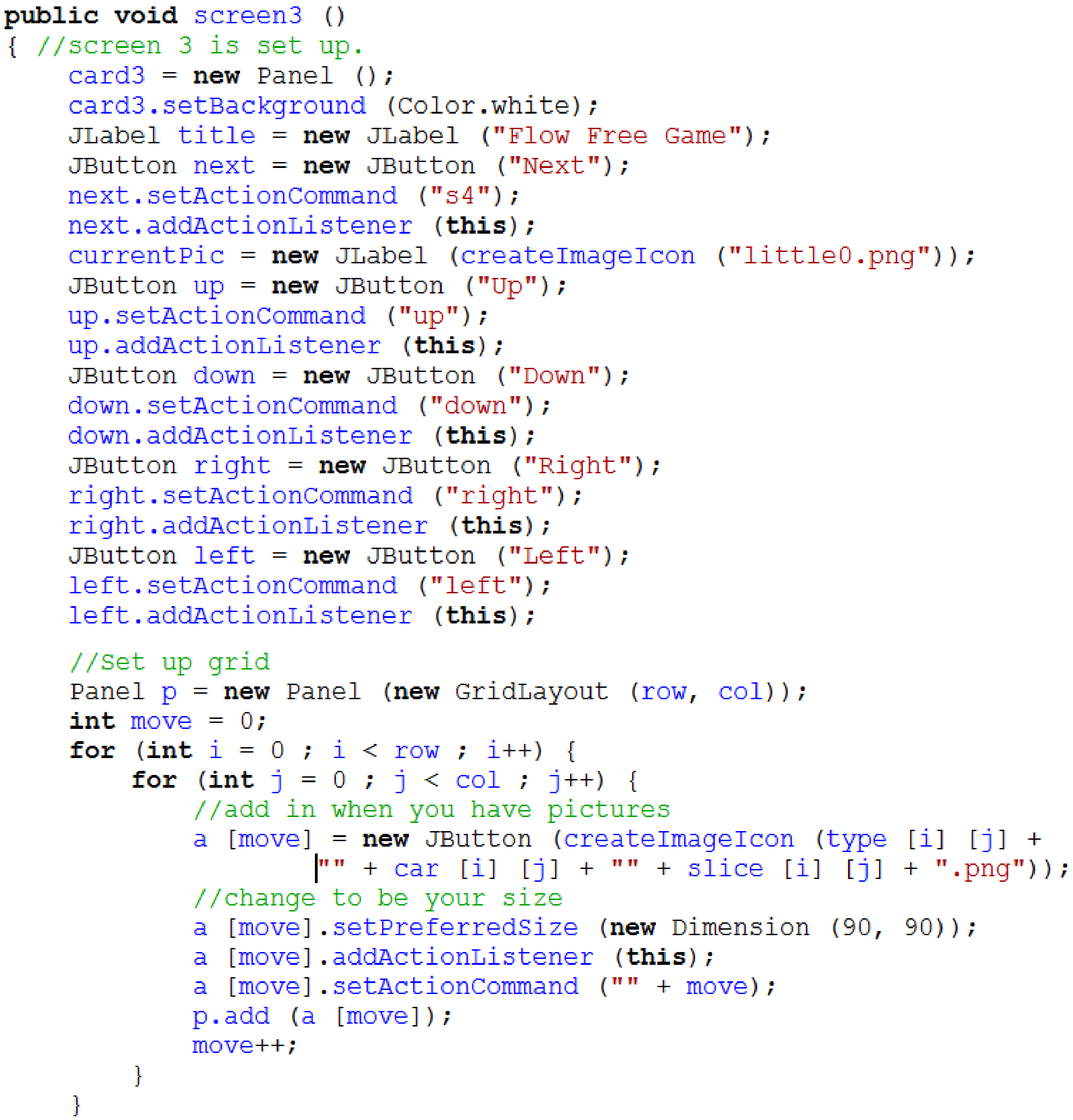
Copy in level 2's arrays

Copy in level 3's arrays

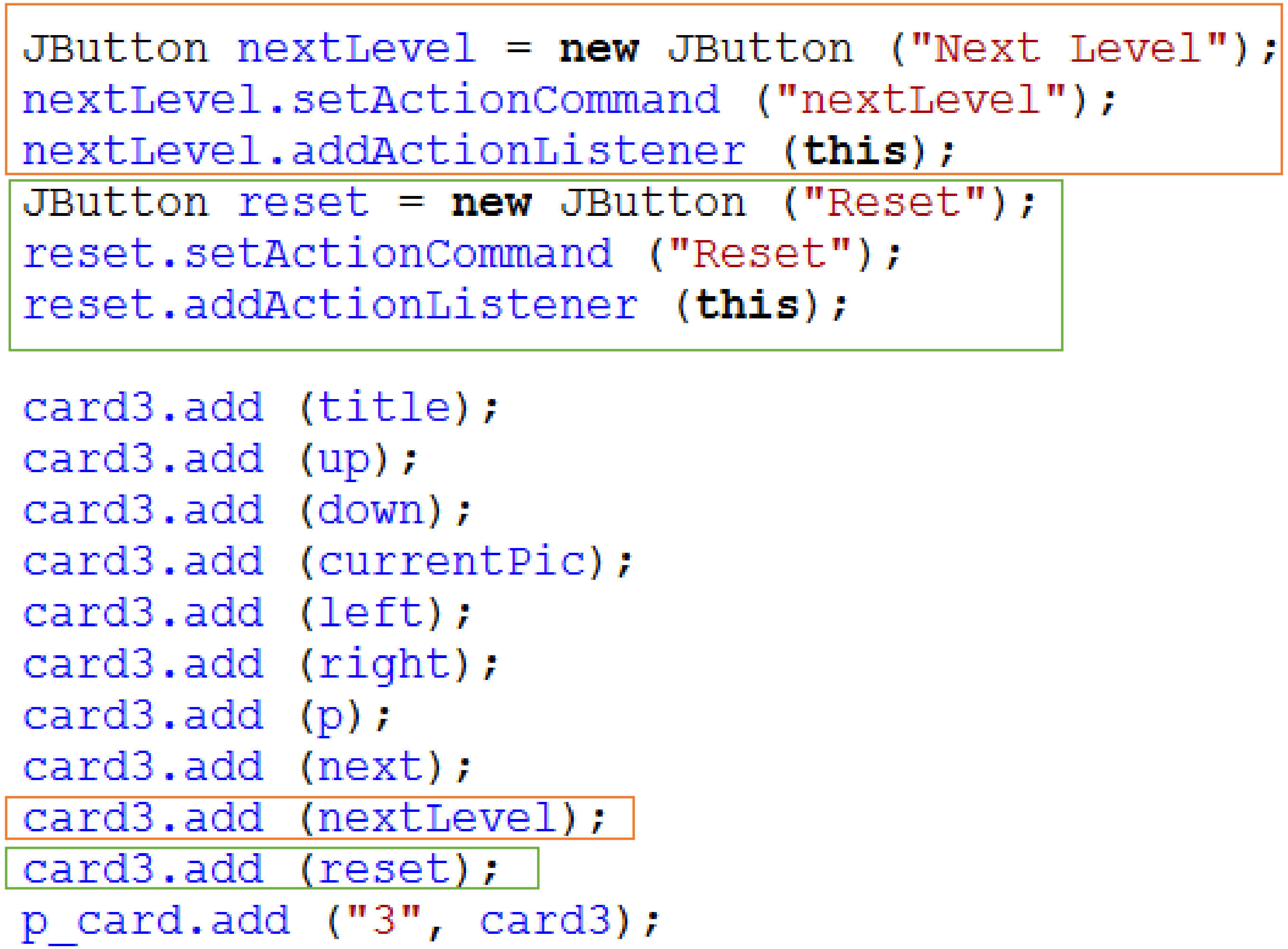
Copy in level 1's arrays

Fix up the current car and redraw

Add a next level and reset button to the game screen

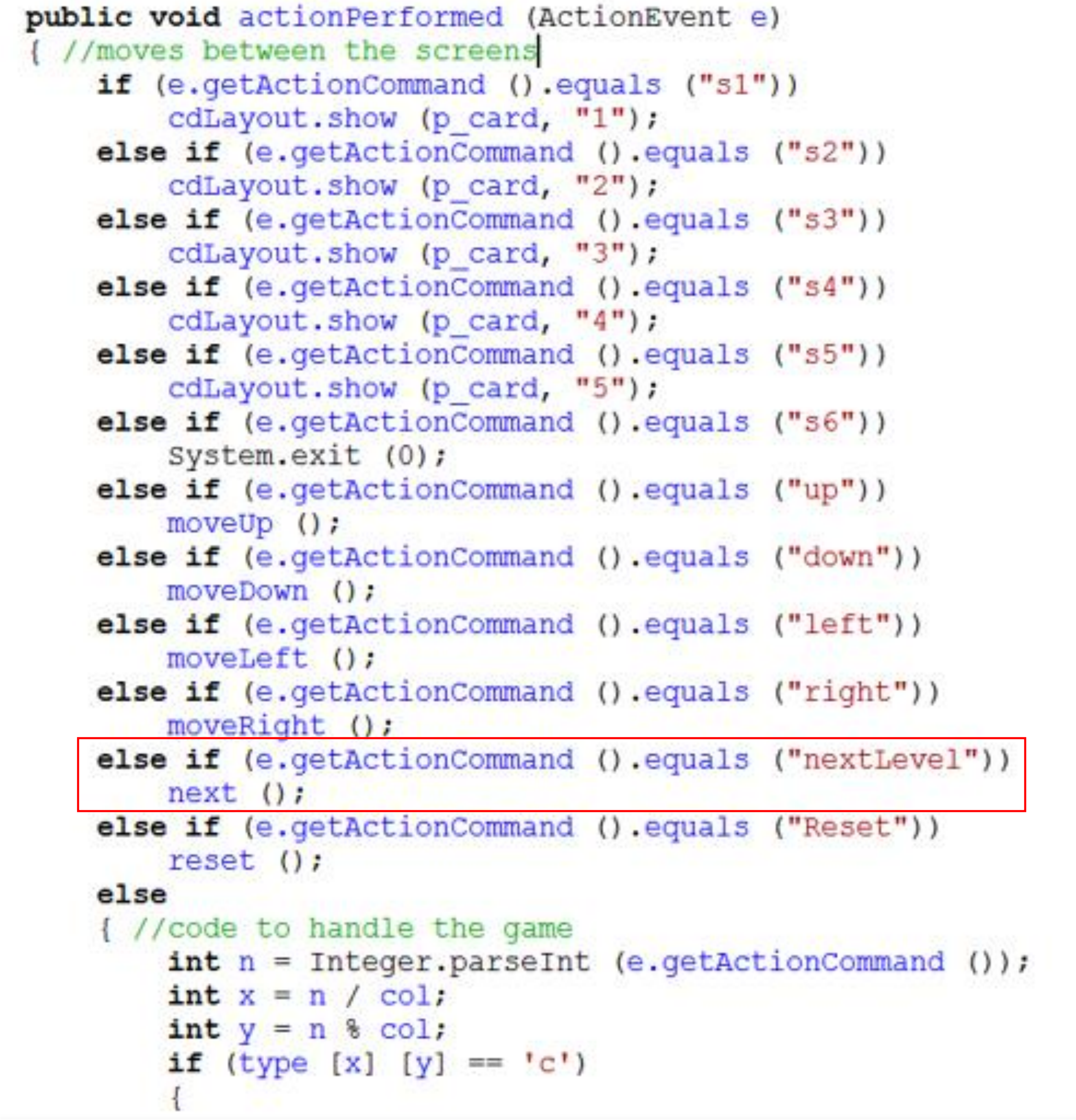


```
public void screen3 ()
{ //screen 3 is set up.
    card3 = new Panel ();
    card3.setBackground (Color.white);
    JLabel title = new JLabel ("Flow Free Game");
    JButton next = new JButton ("Next");
    next.setActionCommand ("s4");
    next.addActionListener (this);
    currentPic = new JLabel (createImageIcon ("little0.png"));
    JButton up = new JButton ("Up");
    up.setActionCommand ("up");
    up.addActionListener (this);
    JButton down = new JButton ("Down");
    down.setActionCommand ("down");
    down.addActionListener (this);
    JButton right = new JButton ("Right");
    right.setActionCommand ("right");
    right.addActionListener (this);
    JButton left = new JButton ("Left");
    left.setActionCommand ("left");
    left.addActionListener (this);

    //Set up grid
    Panel p = new Panel (new GridLayout (row, col));
    int move = 0;
    for (int i = 0 ; i < row ; i++) {
        for (int j = 0 ; j < col ; j++) {
            //add in when you have pictures
            a [move] = new JButton (createImageIcon (type [i] [j] +
                    "" + car [i] [j] + "" + slice [i] [j] + ".png"));
            //change to be your size
            a [move].setPreferredSize (new Dimension (90, 90));
            a [move].addActionListener (this);
            a [move].setActionCommand ("" + move);
            p.add (a [move]);
            move++;
        }
    }
```



```
JButton nextLevel = new JButton ("Next Level");
nextLevel.setActionCommand ("nextLevel");
nextLevel.addActionListener (this);
JButton reset = new JButton ("Reset");
reset.setActionCommand ("Reset");
reset.addActionListener (this);

card3.add (title);
card3.add (up);
card3.add (down);
card3.add (currentPic);
card3.add (left);
card3.add (right);
card3.add (p);
card3.add (next);
card3.add (nextLevel);
card3.add (reset);
p_card.add ("3", card3);
```

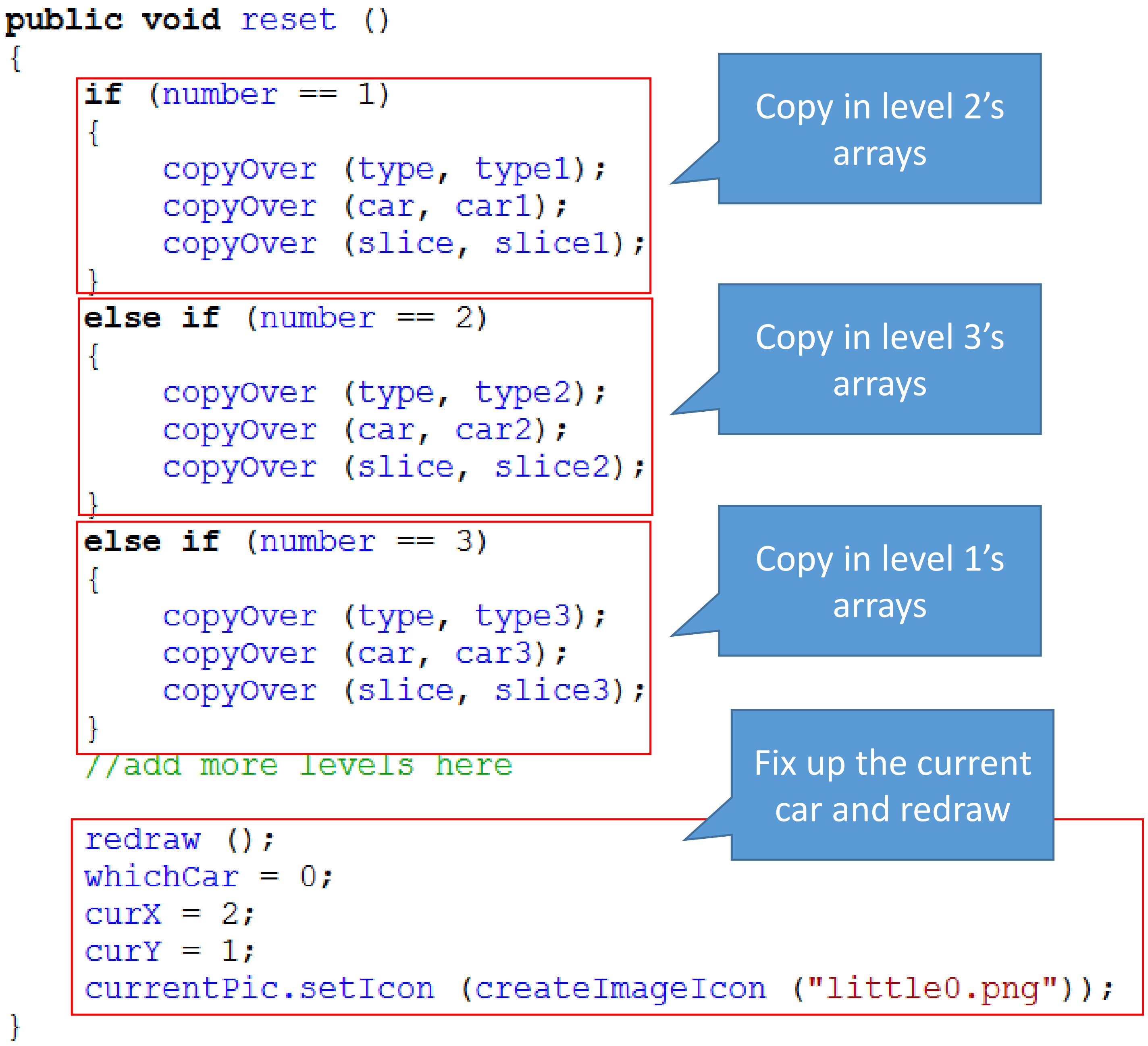
Add the action command to the actionPerformed and call the next method



```java
public void actionPerformed (ActionEvent e)
{ //moves between the screens
    if (e.getActionCommand ().equals ("s1"))
        cdLayout.show (p_card, "1");
    else if (e.getActionCommand ().equals ("s2"))
        cdLayout.show (p_card, "2");
    else if (e.getActionCommand ().equals ("s3"))
        cdLayout.show (p_card, "3");
    else if (e.getActionCommand ().equals ("s4"))
        cdLayout.show (p_card, "4");
    else if (e.getActionCommand ().equals ("s5"))
        cdLayout.show (p_card, "5");
    else if (e.getActionCommand ().equals ("s6"))
        System.exit (0);
    else if (e.getActionCommand ().equals ("up"))
        moveUp ();
    else if (e.getActionCommand ().equals ("down"))
        moveDown ();
    else if (e.getActionCommand ().equals ("left"))
        moveLeft ();
    else if (e.getActionCommand ().equals ("right"))
        moveRight ();
    else if (e.getActionCommand ().equals ("nextLevel"))
        next ();
    else if (e.getActionCommand ().equals ("Reset"))
        reset ();
    else
    { //code to handle the game
        int n = Integer.parseInt (e.getActionCommand ());
        int x = n / col;
        int y = n % col;
        if (type [x] [y] == 'c')
        {
```

TEST completely.

The reset method



```
public void reset ()
{
    if (number == 1)
    {
        copyOver (type, type1);
        copyOver (car, car1);
        copyOver (slice, slice1);
    }
    else if (number == 2)
    {
        copyOver (type, type2);
        copyOver (car, car2);
        copyOver (slice, slice2);
    }
    else if (number == 3)
    {
        copyOver (type, type3);
        copyOver (car, car3);
        copyOver (slice, slice3);
    }
    //add more levels here

    redraw ();
    whichCar = 0;
    curX = 2;
    curY = 1;
    currentPic.setIcon (createImageIcon ("little0.png"));
}
```

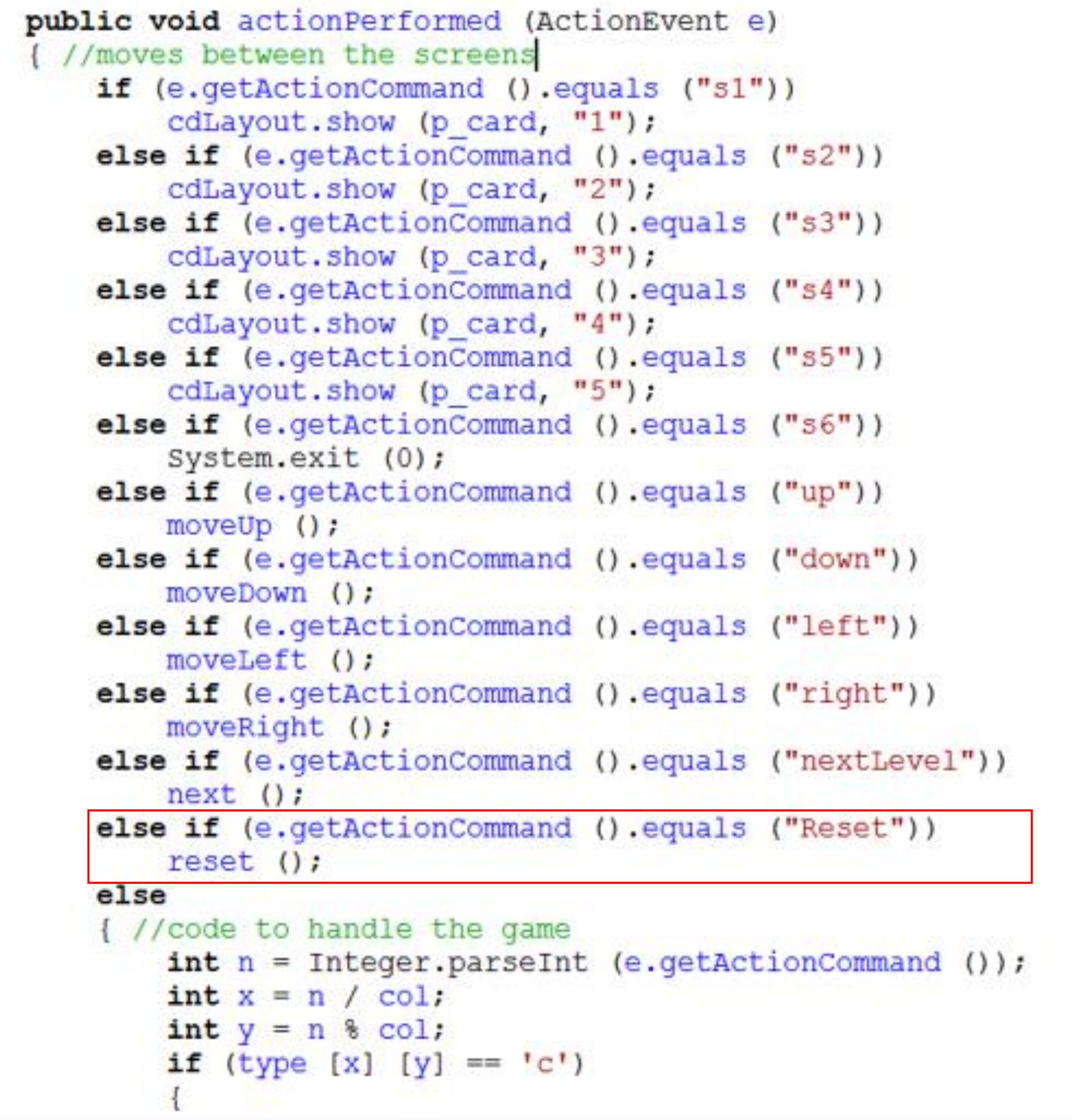
Copy in level 2's arrays

Copy in level 3's arrays

Copy in level 1's arrays

Fix up the current car and redraw

Add the action command to the actionPerformed and call the reset method



```
public void actionPerformed (ActionEvent e)
{ //moves between the screens
    if (e.getActionCommand ().equals ("s1"))
        cdLayout.show (p_card, "1");
    else if (e.getActionCommand ().equals ("s2"))
        cdLayout.show (p_card, "2");
    else if (e.getActionCommand ().equals ("s3"))
        cdLayout.show (p_card, "3");
    else if (e.getActionCommand ().equals ("s4"))
        cdLayout.show (p_card, "4");
    else if (e.getActionCommand ().equals ("s5"))
        cdLayout.show (p_card, "5");
    else if (e.getActionCommand ().equals ("s6"))
        System.exit (0);
    else if (e.getActionCommand ().equals ("up"))
        moveUp ();
    else if (e.getActionCommand ().equals ("down"))
        moveDown ();
    else if (e.getActionCommand ().equals ("left"))
        moveLeft ();
    else if (e.getActionCommand ().equals ("right"))
        moveRight ();
    else if (e.getActionCommand ().equals ("nextLevel"))
        next ();
    else if (e.getActionCommand ().equals ("Reset"))
        reset ();
    else
    { //code to handle the game
        int n = Integer.parseInt (e.getActionCommand ());
        int x = n / col;
        int y = n % col;
        if (type [x] [y] == 'c')
        {
```

TEST completely.