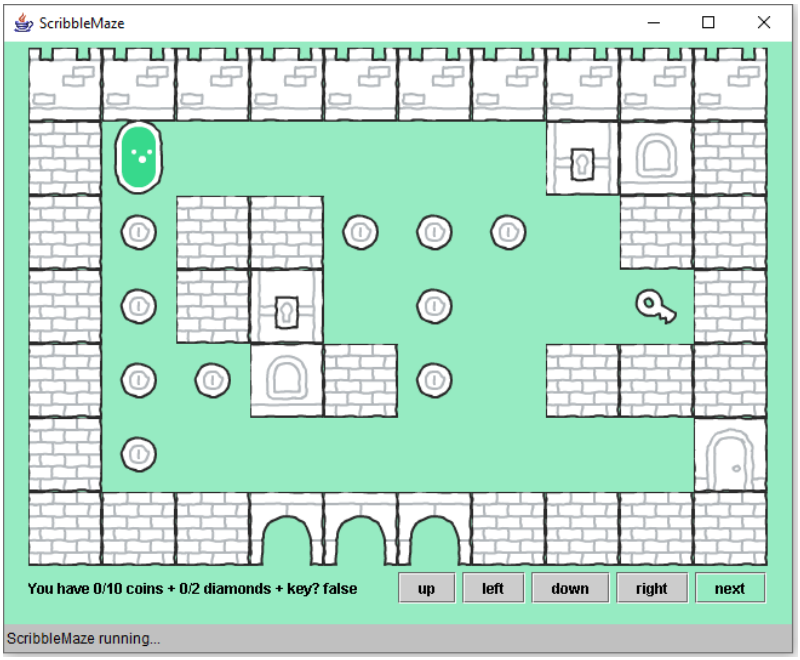
Scribble Maze #1 - Movement

The starter code creates the maze shown on the right.

You need to code the up, left, down, and right buttons so that the character can move around.

The Down direction is shown below. The starter code is on the last page.




```
//Down direction ------------------------------
else if (e.getActionCommand ().equals ("down"))
{
    if (x + 1 >= row)
        showStatus ("Off the board!!");
    else if (b [x + 1] [y] == 0)
        move (x + 1, y, "down");
    else
        showStatus ("There is a wall.");
}
```

Scribble Maze #2 – Coins Pick Up

Make it so that you can pick up coins.

The down direction is shown below. You will need to complete the other directions yourself.


```
//Down direction ---------------------------------
else if (e.getActionCommand ().equals ("down"))
{
    if (x + 1 >= row)
        showStatus ("Off the board!!");
    else if (b [x + 1] [y] == 0)
        move (x + 1, y, "down");
    else if (b [x + 1] [y] == 7)
    {
        coins++;
        b [x + 1] [y] = 0;
        move (x + 1, y, "down");
    }
    else
        showStatus ("There is a wall.");
}
```

Make it so that you can pick up the key.

The down direction is shown below. You will need to complete the other directions yourself.

```
//Down direction ---------------------------------
else if (e.getActionCommand ().equals ("down"))
{
    if (x + 1 >= row)
        showStatus ("Off the board!!");
    else if (b [x + 1] [y] == 0)
        move (x + 1, y, "down");
    else if (b [x + 1] [y] == 7){
        coins++;
        b [x + 1] [y] = 0;
        move (x + 1, y, "down");
    }
    else if (b [x + 1] [y] == 9){
        key = true;
        b [x + 1] [y] = 0;
        move (x + 1, y, "down");
    }
    else
        showStatus ("There is a wall.");
}
```

Scribble Maze #3 – Levels

Add 3 more level arrays to the beginning (this is NOT a good level, yours should be better).

```
int b2[] [] = {{5, 7, 7, 7, 7, 7, 5, 5, 5, 5},
        {4, 0, 0, 0, 0, 0, 0, 6, 3, 4},
        {4, 7, 4, 4, 7, 7, 0, 0, 4, 4},
        {4, 7, 4, 6, 0, 7, 0, 0, 9, 4},
        {4, 7, 7, 3, 4, 7, 0, 4, 4, 4},
        {4, 7, 0, 0, 0, 0, 0, 0, 0, 1},
        {4, 4, 4, 2, 2, 2, 4, 4, 4, 4}};
```

You will have a b2, b3 and b4 when you are done.

Call your new levels in actionPerformed when the next button is pressed:

```
public void actionPerformed (ActionEvent e)
{
    if (e.getActionCommand ().equals ("next"))
    {
        lvl++;
        if (lvl == 2)
            copyOver (b, b2);
        else if (lvl == 3)
            copyOver (b, b3);
        else if (lvl == 4)
            copyOver (b, b4);
        else
        {
            lvl = 1;
            copyOver (b, b1);
        }
    }
    //Up direction ------------------------------
```

Scribble Maze #4 – Opening the Safe, Ending the Level

This code opens the safe if you have a key.

You will need to code the other directions yourself.


```
//Down direction ----------------------------------
else if (e.getActionCommand ().equals ("down"))
{
    if (x + 1 >= row)
        showStatus ("Off the board!!");
    else if (b [x + 1] [y] == 0)
        move (x + 1, y, "down");
    else if (b [x + 1] [y] == 7){
        coins++;
        b [x + 1] [y] = 0;
        move (x + 1, y, "down");
    }
    else if (b [x + 1] [y] == 9){
        key = true;
        b [x + 1] [y] = 0;
        move (x + 1, y, "down");
    }
    else if (b [x + 1] [y] == 6 && key == true){
        move (x + 1, y, "down");
        showDiamonds ();
    }
    else
        showStatus ("There is a wall.");
}
```

Finally, you can't win this game yet. We need to add that.

Add code so that if you have at least 10 coins and 2 diamonds and you are on the door (1), then you win.

Make a popup appear in this case:

```
JOptionPane.showMessageDialog (null, "You have made it out!", "Well done.", JOptionPane.ERROR_MESSAGE);
```

Starter Code

```
import javax.swing.*;
import java.applet.*;
import java.awt.event.*;
import java.awt.*;
public class ScribbleMaze extends Applet implements ActionListener
{
    int row = 7;
    int col = 10;
    int b[] [] = {{5, 5, 5, 5, 5, 5, 5, 5, 5, 5},
            {4, 0, 0, 0, 0, 0, 0, 6, 3, 4},
            {4, 7, 4, 4, 7, 7, 0, 0, 4, 4},
            {4, 7, 4, 6, 7, 7, 0, 0, 9, 4},
            {4, 7, 7, 3, 4, 7, 0, 4, 4, 4},
            {4, 7, 0, 0, 0, 0, 0, 0, 0, 1},
            {4, 4, 4, 2, 2, 2, 4, 4, 4, 4}};
    int b1[] [] = {{5, 5, 5, 5, 5, 5, 5, 5, 5, 5},
            {4, 0, 0, 0, 0, 0, 0, 6, 3, 4},
            {4, 7, 4, 4, 7, 7, 0, 0, 4, 4},
            {4, 7, 4, 6, 7, 7, 0, 0, 9, 4},
            {4, 7, 7, 3, 4, 7, 0, 4, 4, 4},
            {4, 7, 0, 0, 0, 0, 0, 0, 0, 1},
            {4, 4, 4, 2, 2, 2, 4, 4, 4, 4}};

    JLabel pics[] = new JLabel [row * col];
    int sqDimension = 61;
    String picStart = "sc";
    String picFileType = ".png";
    String characterPic = "c1.png";
    String blankPic = "sc0.png";
    String diamondPic = "sc8.png";

    boolean key = false;
    int coins = 0;
    int diamonds = 0;
    JLabel score;
    int x = 1;
    int y = 1;
    int lvl = 1;

    public void init ()
    {
        resize (650, 480);
        setBackground (new Color (150, 235, 194));
        Panel grid = new Panel (new GridLayout (row, col));
        int m = 0;
        for (int i = 0 ; i < row ; i++)
        {
            for (int j = 0 ; j < col ; j++)
            {
                pics [m] = new JLabel (createImageIcon (picStart + b [i] [j] + picFileType));
                pics [m].setPreferredSize (new Dimension (sqDimension, sqDimension));
                grid.add (pics [m]);
                m++;
            }
        }
        add (grid);
        score = new JLabel ("You have " + coins + "/10 coins + " + diamonds + "/2 diamonds + key? " + key);
        score.setPreferredSize (new Dimension (300, 20));
        add (score);
        pics [x * col + y].setIcon (createImageIcon (characterPic));

        JButton up = new JButton ("up");
        up.setActionCommand ("up");
        up.addActionListener (this);
        add (up);
        JButton left = new JButton ("left");
        left.setActionCommand ("left");
        left.addActionListener (this);
        add (left);
        JButton down = new JButton ("down");
        down.setActionCommand ("down");
        down.addActionListener (this);
        add (down);
        JButton right = new JButton ("right");
        right.setActionCommand ("right");
        right.addActionListener (this);
        add (right);
        JButton next = new JButton ("next");
        next.setActionCommand ("next");
        next.setBackground (new Color (150, 235, 194));
        next.addActionListener (this);
        add (next);

    }


    public void move (int newx, int newy, String dir)
    {
        showStatus ("OK! Moving " + dir + ".");
        pics [x * col + y].setIcon (createImageIcon (blankPic));
        x = newx;
        y = newy;
        pics [x * col + y].setIcon (createImageIcon (characterPic));
    }


    public void showDiamonds ()
    {
        b [1] [1] = 8;
        b [5] [3] = 8;
        pics [1 * col + 1].setIcon (createImageIcon (diamondPic));
        pics [5 * col + 3].setIcon (createImageIcon (diamondPic));
    }


    public void copyOver (int a[] [], int c[] [])
    {
        int m = 0;
        for (int i = 0 ; i < row ; i++)
```

```
        {
            for (int j = 0 ; j < col ; j++)
            {
                a [i] [j] = c [i] [j];
                pics [m].setIcon (createImageIcon (picStart + b [i] [j] + picFileType));
                m++;
            }
        }
        x = 1;
        y = 1;
        coins = 0;
        diamonds = 0;
        key = false;
        pics [x * col + y].setIcon (createImageIcon (characterPic));
        showStatus ("New Level begins.");
    }


    public void actionPerformed (ActionEvent e)
    {
        if (e.getActionCommand ().equals ("next"))
        {

        }
        //Up direction ------------------------------------------------
        else if (e.getActionCommand ().equals ("up"))
        {

        }
        //Down direction ----------------------------------------------
        else if (e.getActionCommand ().equals ("down"))
        {

        }
        //Right direction ---------------------------------------------
        else if (e.getActionCommand ().equals ("right"))
        {

        }
        //Left direction ----------------------------------------------
        else if (e.getActionCommand ().equals ("left"))
        {

        }
        score.setText ("You have " + coins + "/10 coins + " + diamonds + "/2 diamonds + key? " + key);

    }



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