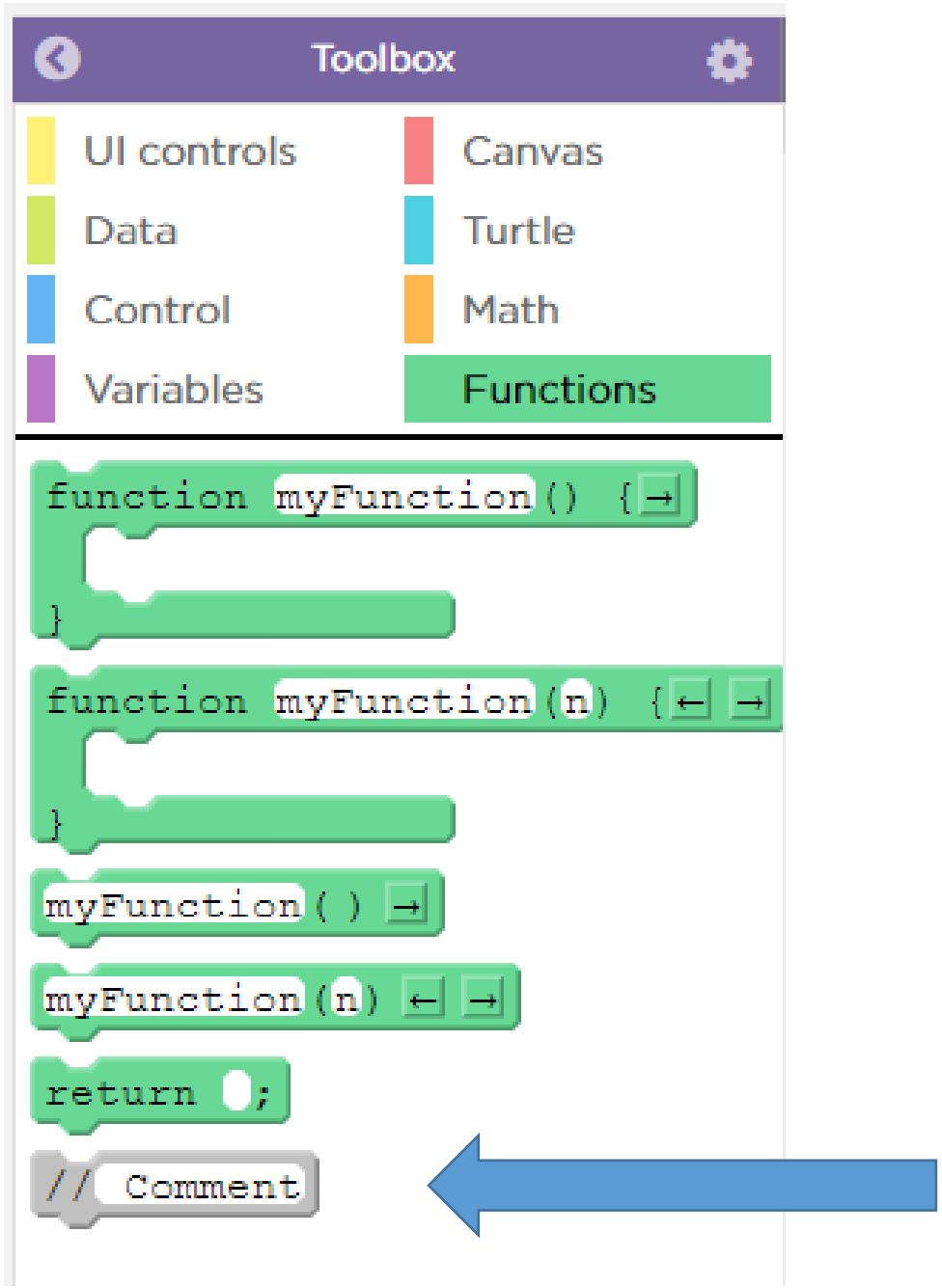
Commenting and Indenting



A comment is code that doesn't run.

This seems like it should be useless, but it really isn't.

PegGame_AG.java* - Ready to Program

File Edit Search Mark Run Help

Stop Pause Open Save Indent Print Cut Copy Paste Find Find Next Replace

```
//Name: Ida Knowe
//Date: Jan 22, 2024
//Purpose: Final Project, Peg Solitaire
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.applet.Applet;
public class PegGame_AG extends Applet implements ActionListener
```

- Title comments – Name, Date and Purpose
- Appear at the top of the code
- Are used to “sign” your work.

PegGame_AG.java* - Ready to Program

File Edit Search Mark Run Help

Stop Pause Open Save Indent Print Cut Copy Paste Find Find Next Replace

```
//Game screen
JLabel turnPic;
int turn = 1; //1 for white, 2 for black
int last = -1; //tracks previous click's p

//Grid
int row = 5;
int col = 5;
//On-screen JButtons
JButton a[] = new JButton [row * col];
//Tracking Array - 1 = open, 2 = piece, 0 = wall
int b[] [] = {{2, 2, 1, 2, 2}, {2, 0, 1, 1, 2}, {0, 0, 0, 0, 0}, {2,
int levelCount = 10;

//Formatting
Color backgroundColour = Color.pink;
```

- Comments added to Global variables
- They note what the variables are used for

PegGame_AG.java* - Ready to Program

File Edit Search Mark Run Help

Stop Pause Open Save Indent Print Cut Copy Paste Find Find Next Replace

```
public void opening ()
{ //Screen 1 - The opening "Splash" screen
    card1 = new Panel ();
    card1.setBackground (backgroundColour);
    JLabel title = new JLabel ("Welcome to _____!");
    title.setFont (new Font ("Arial", Font.PLAIN, 30));
    title.setForeground (titleColour);
    JButton next = new JButton ("Enter");
    next.setPreferredSize (new D
    next.setActionCommand ("card
    next.addActionListener (this
    next.setBackground (buttonCo
    next.setForeground (buttonTe
    card1.add (title);
    card1.add (next);
    p card.add ("1", card1);
```

- Comments added to top of each screen
- Outline what the screen is used for and its number

```
//Game Functionality Section ----------------------------
public void redraw ()
{
    int m = 0;
    for (int i = 0 ; i < row ; i++)
```

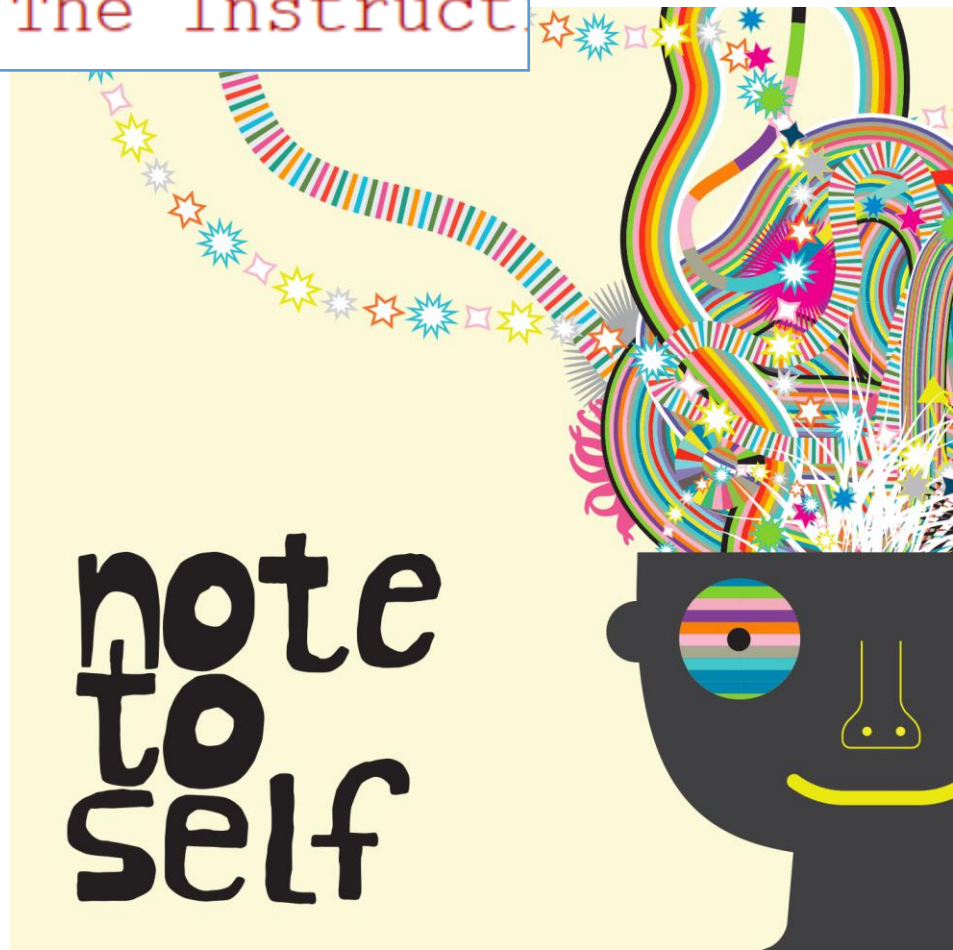
```
//Screen Set up Section ----------------------------

public void opening ()
{ //Screen 1 - The opening "Splash" screen
    card1 = new Panel ();
    card1.setBackground (backgroundColour);
```

- Subtitles in the code
- We can use ------ or other symbols to separate and organize our code.

```
public void instructions ()
{ //TO DO: Fill this comment in
    card2 = new Panel ();
    card2.setBackground (backgroundColour);
    JLabel title = new JLabel ("The Instruct
```

- Notes to yourself
- Sometimes it is handy to leave yourself a note

- All of the TO DO comments should be fixed
- The “TO DO” instruction should be removed.



```
public boolean isValidDownRight (int pos, int last)
{//Checks if can go in a pos down, to the right (x+1)(y+1)
    int endX = pos / col;
    int endY = pos % col;
    int startX = last / col;
    int startY = last % col;
    //end must be blank
    if (b [endX] [endY] != 0)
        return false;
    //start must be peg
    else if (b [startX] [startY]
        return false;
    //middle must be peg
    else if (startX + 1 < row && startY + 1 < col && b [start
        return false;
    //start and end form
    else if (startX + 2 <
        return true;
    //otherwise, it's all
    else
        return false;
}
```

tX

- Comments at the top of each method
- Outline what the method does.

Examples:

- //Redraw – uses tracking array to update screen
- //Move up – moves character up, if possible
- //Returns true if horizontal win, otherwise false
- //Returns true if game is over, otherwise false

```
public boolean isValidDownRight (int pos, int last)
{//Checks if can go in a pos down, to the right (x+1)(y+1)
     int endX = pos / col;
     int endY = pos % col;
     int startX = last / col;
     int startY = last % col;
     //end must be blank
     if (b [endX] [endY] != 0)
          return false;
     //start must be peg
     else if (b [startX] [startY]
          return false;
     //middle must be peg
     else if (startX + 1 < row
          return false;
     //start and end form cor
     else if (startX + 2 < ro
          return true;
     //otherwise, it's all bad
     else
          return false;
}
```

- Comments before major loops and ifs
- Explains what each piece does

Examples:

- //if wall, doesn't move
- //black pawn kill condition
- //bishop move diagonal up & right

Comments Done?

- ❑ Title comments at the top
  - //Name: Ida Knowe
  - //Date: Jan 23, 2023
  - //Purpose: Star Wars Flow Free, ICS3U final project
- ❑ In Global variables
  - //To track score
  - //To set up board
  - //For screens
- ❑ In screens
  - //Sets up screen 1 – splash screen
  - //Sets up screen 3 – game screen, has grid
  - //To set up grid
  - //Save, open, reset buttons at bottom
- ❑ Before ALL methods
  - //Redraw – uses tracking array to update screen
  - //Move up – moves character up, if possible
  - //Returns true if horizontal win, otherwise false
  - //Returns true if game is over, otherwise false
- ❑ Before major loops and ifs
  - //if wall, doesn't move
  - //black pawn kill condition
  - //bishop move diagonal up & right
- ❑ In ActionPerformed
  - //movement between screens
  - //buttons on game screen
  - //all grid movement on game screen
  - //calls all win conditions to check for win

```
/* This part isn't working right now
onEvent("id", "click", function( ) {
 setText("id", "text");
 playSound("sound://default.mp3", false);
});
*/
```

- Commenting out code
- Use /* and */ to temporarily take out code
- Lets you save code that isn’t running, but still run the code to test other things.

```
// I am not sure if we need this, but too scared to delete.

// Magic. Do not touch.


// Dear maintainer:
//
// Once you are done trying to 'optimize' this routine,
// and have realized what a terrible mistake that was,
// please increment the following counter as a warning
// to the next guy:
//
// total_hours_wasted_here = 42


// TODO make this work
```

```
// Dear future me. Please forgive me.
// I can't even begin to express how sorry I am.


// it was hard to write
// so it should be hard to read


// Houston, we have a problem


// NO COMMENT


// If you're reading this, that means you have been
// put in charge of my previous project.
// I am so, so sorry for you.
```