

Drawing Blocks

1 Overview

In this lab we will utilize Java's Swing library to draw a picture of a falling block puzzle (of no particular variety) using rectangles. In this game tiles fall from the ceiling, are rotated and land either at the bottom of the screen or on top of other blocks. We don't have the necessary tools to create such a game at this point in the course, so we will simply be creating a picture of such a game in progress. In this game there are several varieties of blocks that can fall, each with its own shape and color. We will create these block configurations and stamp them onto our canvas.

2 Learning Outcomes

By the end of this project students should be able to:

- Instantiate objects;
- Create a simple graphical displays;
- Use the Java API to solve a problem;
- work effectively with a partner using pair-programming;
- write an effective report that describes the students' problem solving process.

3 Pre-Lab Instructions

Do this part before you come to lab:

This project will require you to utilize classes from the Java Class Library.

- Read Big Java Chapter 2

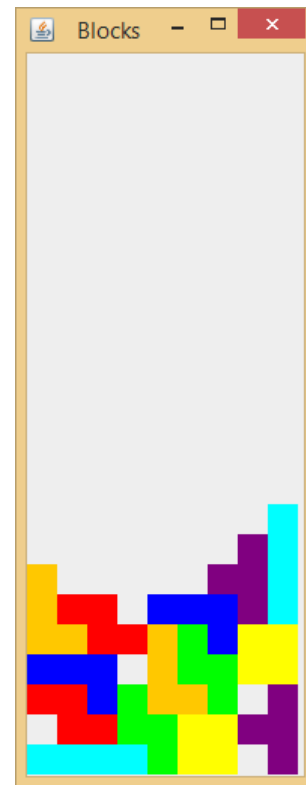


Fig. 1: A novel falling blocks game

- Look up the Java API documentation for the class `Graphics2D`. Determine what methods might help you accomplish the following tasks:
 - Change the color of the pen
 - Draw an outline
 - Draw a solid shape
 - Change the angle of the next thing you draw
 - Change the location of the next thing you draw

4 Lab Instructions

Do this part in lab:

Step 1

To start your code use the examples shown at the end of Big Java chapter 2 to create a custom component class and to generate that component in a `JFrame`. Once you have your frame working, attempt to draw a single block in the top left corner of the screen by instantiating rectangles and a color. Each square in the block should be 20x20. Every block type should have a unique and consistent color. In this assignment you will have seven different block types to create.

- I block - A block consisting of 4 consecutive squares in a straight line. (colored cyan in the example screenshot)
- J block - A block consisting of 3 consecutive squares in a line, with a single block coming out horizontal at one end. (colored blue in the example screenshot)
- L block - The mirror image of the J block. (colored orange in the example screenshot)
- O block - A block consisting of 4 squares forming a larger square. (colored yellow in the example screenshot)
- S block - A block consisting of 4 squares connected horizontally, vertically, then horizontally in an S shape. (colored green in the example screenshot)
- Z block - The mirror image of the S block. (colored red in the example screenshot)
- T block - A block in the shape of a T. (colored purple in the example screenshot)

Develop the rectangles and colors required for each of these types in different variables, and modify your draw code to be able to easily draw any of these blocks that you wish.

Step 2

Now that you can draw any block type, build your JFrame to be 10 squares wide, and draw each of these blocks at the bottom of the screen in a legal arrangement. Each block should align with the others, and must be resting at the bottom of the screen or directly above another block. Each block should be drawn at least twice, but no new rectangle should be created for the second drawing. Once you are complete you should have a drawing similar to the example screenshot.

Hint: Rather than trying to modify your rectangles, try moving the entire Graphics2D object!

5 Lab Report

Each pair of students will write a single lab report together and each student will turn in that same lab report on BBLearn. Submissions from each student on a pair should be identical.

Your lab report should begin with a preamble that contains:

- The lab assignment number and name
- Your name(s)
- The date
- The lab section

It should then be followed by four numbered sections:

1. Problem Statement

In this section you should describe the problem in **your** own words. The problem statement should answer questions like:

- What are the important features of the problem?
- What are the problem requirements?

This section should also include a reasonably complete list of requirements in the assignment. Following your description of the problem, include a bulleted list of specific features to implement. If there are any specific functions, classes or numeric requirements given to you, they should be represented in this bulleted list.

2. Planning

In the second section you should describe what planning you did in order to solve the problem. You should include planning artifacts like sketches, diagrams, or pseudocode you may have used. You should also describe your planning process. List the specific data structures or techniques you plan on using, and why.

3. Implementation and Testing

In the third section you should describe how you implemented your plan. As directed by the lab instructor you should (as appropriate) include:

- a copy of your source code (Submitted in BBLearn as a .java files)
- a screen shot of your running application / solution
- results from testing

4. Reflection

In the last section you should reflect on the project. Consider different things you could have done to make your solution better. This might include code organization improvements, design improvements, etc.

You should also ask yourself what were the key insights or features of your solution? Were there alternative approaches or techniques you could have employed? How would these alternatives have impacted a different solution?

5. Partner Rating

Every assignment you are required to rate your partner with a score -1, 0 or +1. This should be submitted in the comment section of the BBLearn submission, and not in the report document. If you don't want to give your partner a negative rating making sure not to use a dash before listing the number! You do not have to tell your partner the rating you assign them. A rating of 1 indicates that your partner was particularly helpful or contributed exceptional effort. A rating of 0 indicates that your partner met the class expectations of them. Rating your partner at -1 means that they refused contribute to the project, failed to put in a reasonable effort or actively blocked you from participating. If a student receives three ratings of -1 they must attend a mandatory meeting with the instructor to discuss the situation, and receiving additional -1 ratings beyond that, the student risks losing a letter grade, or even failing the course.

Colophon

This project was developed by Richard Lester and Dr. James Dean Palmer of Northern Arizona University. Except as otherwise noted, the content of this document is licensed under the [Creative Commons Attribution-ShareAlike 4.0 International License](#).