

# Exercise Set: Control Flow and Data Models

A lot of the code snippets used in this exercise are taken from JHotDraw, an open-source project that makes it easy to write graphical editors, such as providing a drawing program. For more information on JHotDraw, take a look at <http://www.jhotdraw.org/>. For more information on open source, you can visit <http://www.opensource.org/>.

In this exercise set, we have marked questions we think are harder than others with a [‡]. We have also marked questions for which solutions are provided at the end of the set ([SP]). To check solutions for other questions than those marked with [SP], ask one of the instructors or TAs or post a question to the Piazza site!

Note that the exercises may also introduce some new Java syntax. Use resources on the web and the Java tutorials to help you decipher unknown syntax. If that doesn't work, ask a TA, an instructor or on Piazza!

1. Extract an intra-method control-flow diagram (flowchart) from each of the following code snippets. Given that we haven't provided you with all the relevant source code, you won't be able to use the Eclipse debugger to help. We'd like you to recognize control structures that you've seen in class or in the lab, recall what they do and thereby construct the flowchart.

a)

```
public void promptOpen() {
    JFileChooser openFileDialog = createOpenFileChooser();
    if (openDialog.showOpenDialog(this) == JFileChooser.APPROVE_OPTION) {
        StorageFormat foundFormat = findStorageFormat(openDialog
            .getFilter());
        // if there is no format associated,
        // try to find one that supports the file
        if (foundFormat == null) {
            foundFormat = findStorageFormat(openDialog.getSelectedFile());
        }
        if (foundFormat != null) {
            loadDrawing(foundFormat);
        }
        else {
            showStatus("Not a valid file format: ");
        }
    }
}
```

b)

```
public void mouseDown(MouseEvent e, int x, int y) {
    setView((DrawingView)e.getSource());
    if (e.getClickCount() == 2) {
        Figure figure = drawing().findFigure(e.getX(), e.getY());
        if (figure != null) {
            inspectFigure(figure);
            return;
        }
    }
    super.mouseDown(e, x, y);
}
```

```

c)
/**
 * Draws a pattern background pattern by replicating an image.
 */
private void drawPattern(Graphics g, Image image, DrawingView view) {
    int iwidth = image.getWidth(view);
    int iheight = image.getHeight(view);
    Dimension d = view.getSize();
    int x = 0;
    int y = 0;

    while (y < d.height) {
        while (x < d.width) {
            g.drawImage(image, x, y, view);
            x += iwidth;
        }
        y += iheight;
        x = 0;
    }
}

```

#### d) [‡], [SP]

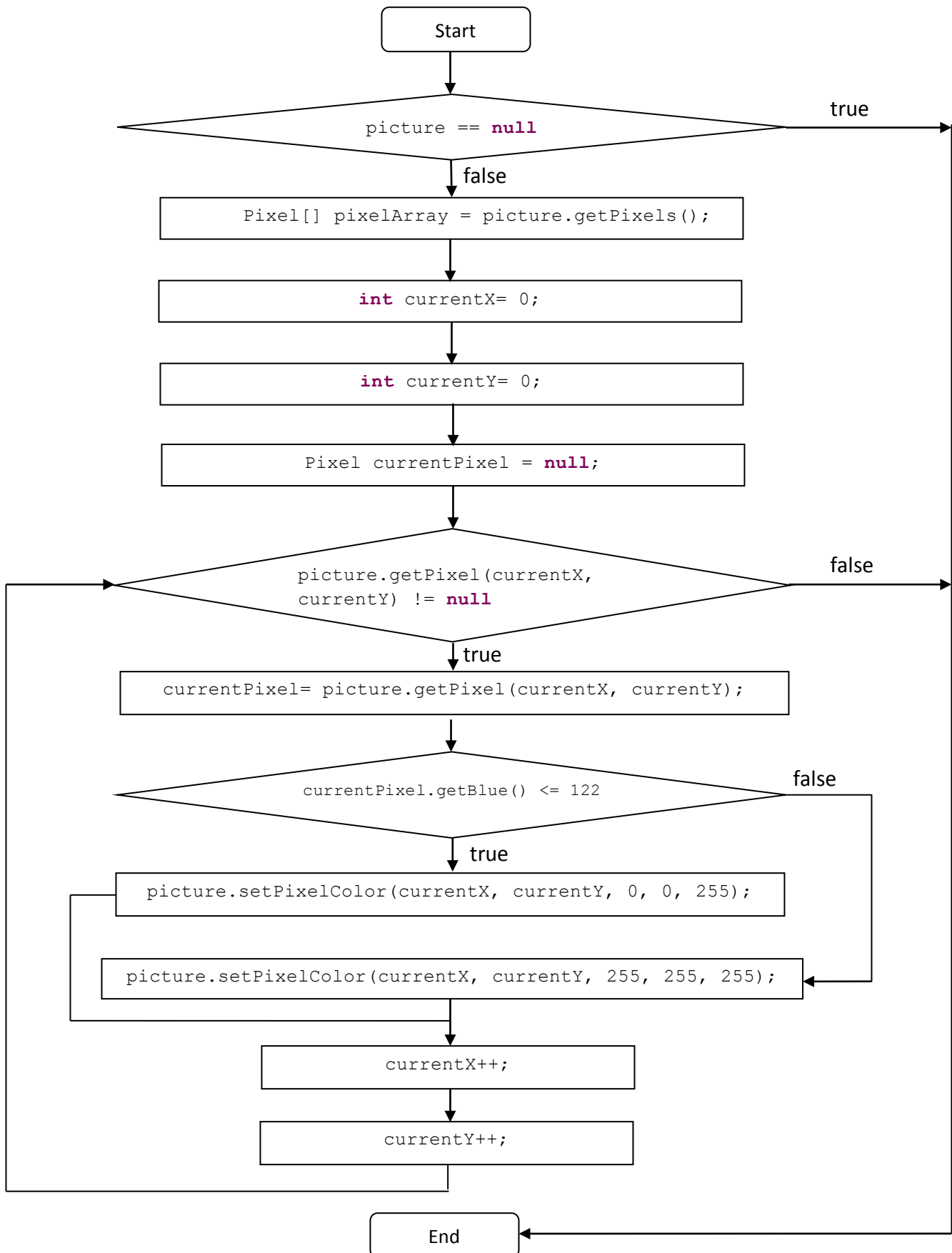
```

/**
 * Find a StorageFormat that can be used according to a file object to store
 * a Drawing in a file or restore it from a file respectively.
 *
 * @param file
 *         a File object to be matched
 * @return StorageFormat, if a matching file extension could be found,
 *         <code>null</code> otherwise
 */
public StorageFormat findStorageFormat(File file) {
    StorageFormat currentStorageFormat;
    for (int i = 0; i < myStorageFormats.size(); i++) {
        currentStorageFormat = (StorageFormat) myStorageFormats.get(i);
        if (currentStorageFormat.getFileFilter().accept(file)) {
            return currentStorageFormat;
        }
    }
    return null;
}

```

**2. Extracting models of inter-method control flow:** Extract a call graph starting from the `GamePanel.drawGame` method in the `SpaceInvadersBase` system. Do not include calls to methods in the Java library. **[SP]**

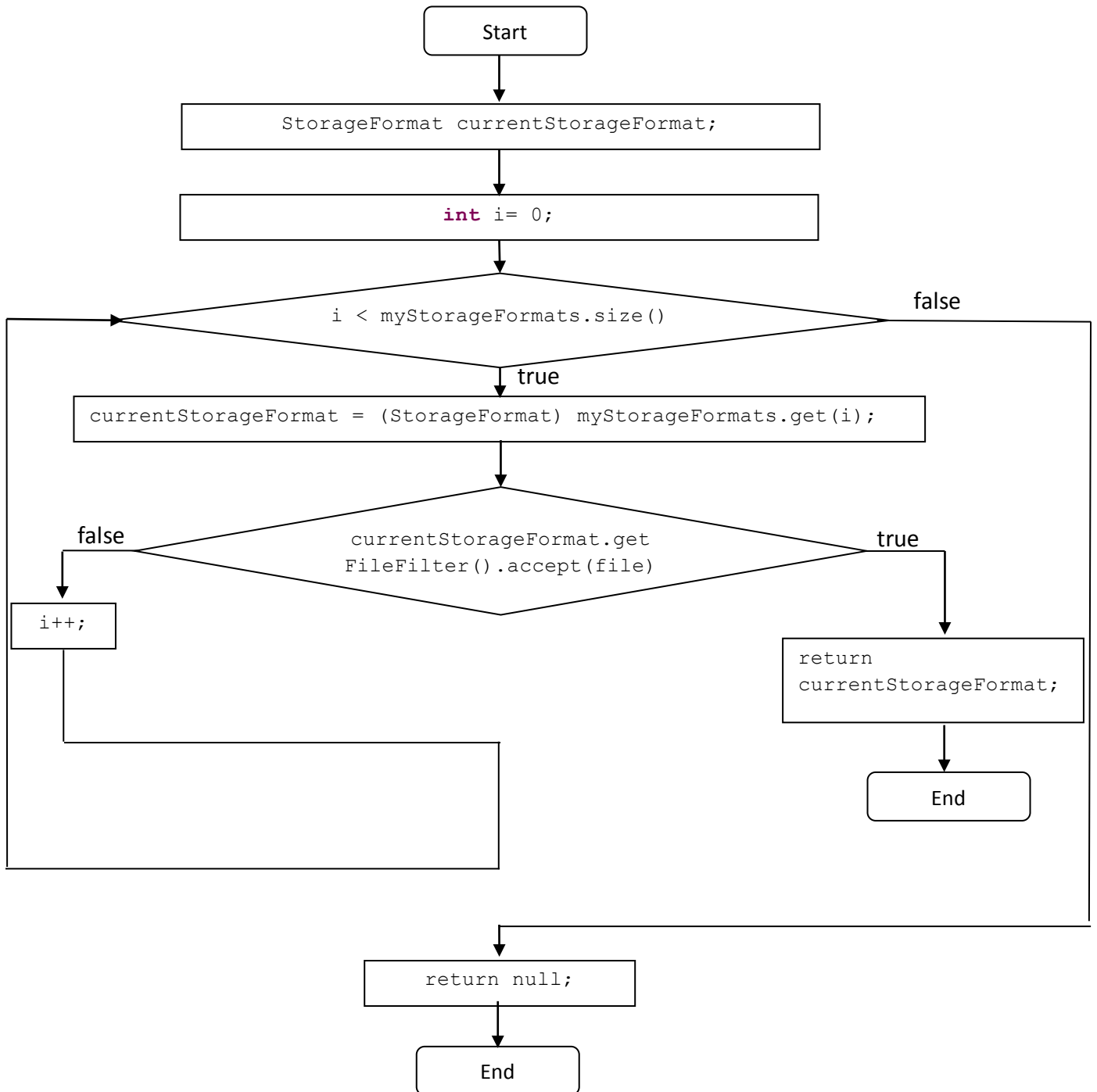
3. Given the following flowchart, explain in words what task the method accomplishes. [‡]



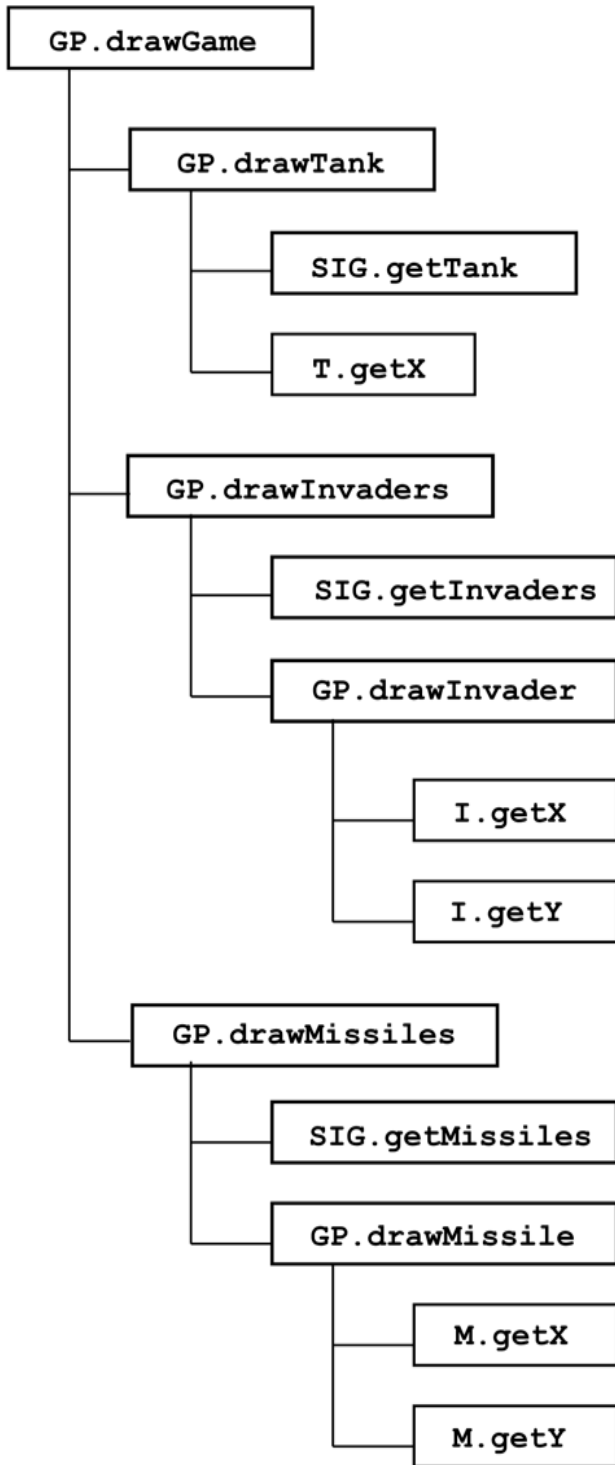
**4.** Extract a sequence diagram starting from the `GamePanel.drawGame` method in the `SpaceInvadersBase` system. Do not include calls to methods in the Java library.

## SOLUTIONS:

1. d)



2a)



GP - GamePanel

SIG - SIGame

T - Tank

I - Invader

M - Missile