

Errata for
The Design and Implementation of Multimedia Software

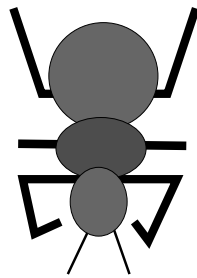
David Bernstein

Jones & Bartlett Learning

First Edition, 2011

ISBN-13: 978-0-7637-7812-5

ISBN-10: 0-7637-7812-5



Errata

Preface

Page vii

Last line: “Fielder” should be “Fiedler”.

Chapter 2

Page 24

The second paragraph should start with:

Swing has many other components including `JCheckBox`, `JList`, `JSlider`, `JSpinner`, `JTextArea`, and `TextField`.

Page 41

Question 12: Should be deleted. (A variant of this question is included in Chapter 3 where it belongs.)

Chapter 4

Page 74

Figure 4.2 should appear as follows:

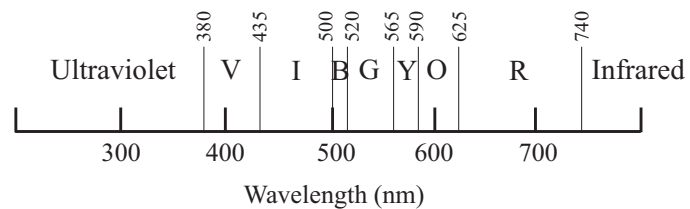


Figure 1 The Visible Spectrum

4 Chapter 0 Errata

Page 98

The `repaint()` method in the `visualization` class should be `public` so that objects other than descendants can start the rendering process.

Chapter 5

Page 133

Figure 5.15 should use the (more recent) `HashMap` class rather than the `Hashtable` class.

Page 134

`createOp()` should use the (more recent) `HashMap` class rather than the `Hashtable` class.

Page 138

`BufferedImageOpFactory` class should use the (more recent) `HashMap` class rather than the `Hashtable` class.

The code fragment is incomplete (i.e., does not include the declaration and construction of the `HashMap` of `HashMap` objects).

Chapter 6

Page 160

Equations (6.14) and (6.15) should be:

$$\begin{aligned}x(u) &= (1 - u)^3 p_x + 3(1 - u)^2 u r_x + 3(1 - u) u^2 s_x + u^3 q_x \\y(u) &= (1 - u)^3 p_y + 3(1 - u)^2 u r_y + 3(1 - u) u^2 s_y + u^3 q_y\end{aligned}$$

Page 170

The second example includes the comment:

```
// Fill in gray
```

It should say:

```
// Fill in yellow
```

Chapter 7

Page 188

In the first paragraph, the sentence that begins “The biggest shortcoming” should be: The biggest shortcoming of this design is that both the `Visualization` and `VisualizationView` classes must now distinguish between `Content` objects (or, really, objects that implement the `SimpleContent` interface) and `AggregateContent` objects.

Chapter 8

Page 218

In the first paragraph of the aside, `FilenamFilter` should be `FilenameFilter`.

Chapter 9

Page 243

In the text immediately following F9.3, the phrase “described dynamic content” should be replaced with “sampled dynamic content” in two places.

Page 244

In the second paragraph of the aside, the phrase “subinterface extends to superinterfaces” should be replaced with “subinterface extends two superinterfaces”.

Page 266

The `handleTick()` method should immediately return for small values of `time` [e.g., add the statement `if (time < 100) return;`] to ensure that the initial rendering occurs before intersections are checked.

Page 267

`speed` should be assigned the value `1 + rng.next(10)` to ensure that the balloon does not have a speed of 0.

Pages 268-269

The `TweeningSprite` class should use `ArrayList` objects rather than `Vector` objects since these collections do not need to be thread safe.

6 Chapter 0 Errata

Pages 276-277

The `SampledSprite` class should use an `ArrayList` object rather than a `Vector` object since the collection does not need to be thread safe.

Pages 280-281

The `DescribedSprite` class should use an `ArrayList` object rather than a `Vector` object since the collection does not need to be thread safe.

Pages 281-282

The `getContent()` should not use the local variable `result`. Instead, it should always update the attribute `tweened` and return it, as follows:

```
public visual.statik.TransformableContent getContent()
{
    int                current, next;
    AggregateContent   currentCTC, nextCTC;

    current = getKeyTimeIndex();
    next    = getNextKeyTimeIndex();
    tweened = null;

    if (current >= 0)
    {
        currentCTC = content.get(current);
        nextCTC    = content.get(next);

        tweened    = currentCTC;

        if (currentCTC != nextCTC)
        {
            tweenShape(currentCTC, nextCTC, getInterpolationFraction());
        }
    }
    return tweened;
}
```

Page 285

In question 11, `actionPerformed()` should be `handleTick()`.

Chapter 11

Page 301

The `InputStream` should be decorated as a `BufferedInputStream` (which supports `mark()` and `skip()`) before the `AudioInputStream` is constructed.

Chapter 12

Page 334

The text “*-idx:footnote*” should not be in footnote 2.

Page 340

The `startPlaying()` and `stopPlaying()` methods should be as follows (to account for rests):

```
protected void startPlaying(MidiChannel channel)
{
    if (midiNumber >= 0) channel.noteOn(midiNumber, 127);
}

protected void stopPlaying(MidiChannel channel)
{
    if (midiNumber >= 0) channel.noteOff(midiNumber, 127);
}
```

Page 346

The example of how to construct the F-Major scale should appear as follows:
To construct the F-Major scale, start with

```
F  G  A  B  C  D  E  F
T  T  T  S  T  T  S
```

and then add sharps or flats to adjust for the desired pattern:

```
T  T  S  T  T  T  S
F  G  A  Bb C  D  E  F
```

Pages 352-356

The `Score` and `Orchestra` classes should use the (more recent) `HashMap` class and `Iterator` interface rather than the `Hashtable` class and `Enumeration` interface.

The constructor in the `Orchestra` class uses a method called `findSoundbank()` the implementation of which is not discussed. The original implementation was as follows.

```
private Soundbank findSoundbank() throws MidiUnavailableException
{
    InputStream          is;
    ResourceFinder       finder;
    Soundbank            sb;
    URL                  url;

    sb = null;

    finder = ResourceFinder.createInstance(this);
    is     = finder.findInputStream("soundbank-mid.gm");

    try
    {
        sb = MidiSystem.getSoundbank(is);
    }
    catch (Exception e)
    {
        throw(new MidiUnavailableException());
    }

    return sb;
}
```

This implementation will not load the default soundbank (i.e., `soundbank-mid.gm`) in versions 1.7 and later of Java because the format of that soundbank is not supported in those version of Java. Hence, you may need to use a a (more recent) sound font instead (and modify the code accordingly).