

CSE 113 B

October 25 – 29, 2010

ANNOUNCEMENTS

- ⊙ Lab Practical 1 this week
- ⊙ Lab 7 week of November 1st
- ⊙ Exam 3 November 8th in lecture



GREENFOOTIMAGE

- ③ We can change our images by actually painting to them by using methods that are available in GreenfootImage
- ③ Can paint to the image (we did this in the example ImageFun)
 - ③ fill method
 - ③ drawOval method



CHAPTER 7 - DRAWING STARS/ DRAWING POLKA DOTS

- ⦿ Drawing polka dots on the screen
 - ⦿ Create method for drawing dots and call it from constructor of Background
 - ⦿ Inside method we retrieve the background image and draw ovals at random locations
 - ⦿ We also added functionality to create dots in random shades of blue, and totally random colors.



JAVA.AWT.COLOR

- ⦿ There are several pre-defined colors in Java that you can use
- ⦿ `java.awt.Color.PINK`, `java.awt.Color.RED`,
`java.awt.Color.ORANGE`, `java.awt.Color.YELLOW`,
`java.awt.Color.GREEN`, `java.awt.Color.CYAN`,
`java.awt.Color.BLUE`, `java.awt.Color.MAGENTA`,
`java.awt.Color.LIGHT_GRAY`, `java.awt.Color.GRAY`,
`java.awt.Color.DARK_GRAY`, `java.awt.Color.BLACK`,
`java.awt.Color.WHITE`



JAVA.AWT.COLOR

- ① You can also create a color using

```
new java.awt.Color(red, green, blue)
```
- ① where you substitute a number within the range 0-255 for each of red, green, and blue



FOR-LOOP (BASIC SYNTAX)

```
for ( /* initialization, condition, increment */ )  
{  
    //code to be repeated  
}
```



FOR-LOOP (INITIALIZATION)

- ③ The initialization part of a for-loop (typically) creates and initializes a loop counter. The loop counter is simply a variable whose value we can check (in the condition part) and change (in the increment part).
- ③ A sample initialization step looks like this:

```
int count = 0;
```
- ③ You can name your variable whatever you'd like and initialize it to whatever value is appropriate to your task.



FOR-LOOP (CONDITION)

- ③ The condition part of a for-loop tells the loop when to stop. While the condition is true, the code will keep getting repeated, when it is false, the repetition will stop. Typically, the condition involves the loop counter variable's value.
- ③ A sample condition looks like this:

```
count < 100;
```
- ③ The condition must evaluate to true or false (must be a boolean expression) and typically gives a clue to the number of times the loop will execute.



FOR-LOOP (INCREMENT)

- ⦿ In increment part of a for-loop indicates how the loop counter will be incremented. In many cases, the counter is incremented by one each time (so it keeps count of how many times the loop has executed).

- ⦿ A sample increment step looks like this:

```
count = count + 1
```

- ⦿ Note that there is a (syntactic) shortcut for the above that is typically used:

```
count++
```



FOR-LOOP (REPEATED CODE)

- ⦿ This is the code that is to be repeated. There is no restriction on what is placed in the {}. Any valid Java code can be repeated.



FOR-LOOPS (A NOTE ABOUT THESE GUIDELINES)

- ◎ The usage of the for-loop described here corresponds with its most common usage. However, the only thing that the Java language specifies is that within the parentheses there be three statements separated by commas and that the statements in between the {} are the code that is repeated.

- ◎ So, that means

```
for(;;)  
{  
}
```

- ◎ Is a perfectly syntactically correct for-loop. That loop will run forever (it will never stop), but perform no actions. This is not pointed out to panic anyone, but rather to indicate that there may be other uses of the for loop that you can see that do not conform to the usage we've discussed here.















ANIMATION EFFECTS

- ① Note that with the example for increasing the size of the graphic during animation (the bomb and explosion), we first created an array with the different sizes of images.
- ① Note that the `initializeImages` method in the `Bomb` sets up this array.



ANIMATION EFFECTS

- ③ The `act()` method in bomb and explosion simply goes through each image in this array and shows the next one when `act()` is called until it reaches the end of the array of images.



SMOKE

- ③ The smoke effect decreases the size of an image and also increases its transparency over successive act() calls.

