HW 3: Breakout

You are here

The River of Java

Memory

Events

Animation

Graphics Programs
Announcements

- Assignment 2 due tomorrow (Wed, July 10th), at 10AM
- Assignment 3 out tomorrow after class
- Assignment 1 grades released
- Lecture Assignments for feedback available for those who joined late
Plan for Today

- Review: Graphics
- Boolean Returns
- Animation
- Randomness
- Sand Art
Plan for Today

- Review: Graphics
- Boolean Returns
- Animation
- Randomness
- Sand Art
Review: Working with Graphics

We make graphics programs by creating **graphics objects** and manipulating their properties.
We can create graphics objects using the `new` keyword:

```java
GRect rect = new GRect(100, 200);
```
Review: Working with Graphics

- We can **create** graphics objects using the `new` keyword:

  ```java
  GRect rect = new GRect(100, 200);
  ```

- We can **manipulate** graphics objects by calling methods on those objects:

  ```java
  rect.setColor(Color.BLUE);
  ```

  Who?  What?  What specifically?
Plan for Today

- Review: Graphics
- Boolean Returns
- Animation
- Randomness
- Sand Art
public void run() {
    for (int i = 1; i <= 100; i++) {
        if (/* check if i is a square number */) {
            println(i);
        }
    }
}
public void run() {
    for (int i = 1; i <= 100; i++) {
        if (isSquare(i)) {
            println(i);
        }
    }
}
public void run() {
    for (int i = 1; i <= 100; i++) {
        if (isSquare(i)) {
            println(i);
        }
    }
}

private boolean isSquare(int x) {
    double root = Math.sqrt(x);
    if (root == Math.floor(root)) {
        return true;
    } else {
        return false;
    }
}
```java
public void run() {
    for (int i = 1; i <= 100; i++) {
        if (isSquare(i)) {
            println(i);
        }
    }
}

private boolean isSquare(int x) {
    double root = Math.sqrt(x);
    if (root == Math.floor(root)) {
        return true;
    } else {
        return false;
    }
}
```
public void run() {
    for (int i = 1; i <= 100; i++) {
        if (isSquare(i)) {
            println(i);
        }
    }
}

private boolean isSquare(int x) {
    double root = Math.sqrt(x);
    return root == Math.floor(root);
}
public void run() {
    for (int i = 1; i <= 100; i++) {
        if (isSquare(i)) {
            println(i);
        }
    }
}

private boolean isSquare(int x) {
    double root = Math.sqrt(x);
    return root == (int)root;
}
Plan for Today

- Review: Multiple Return, More Graphics
- Boolean Returns
- Animation
- Randomness
- Sand Art
public void run() {
    // setup

    while (condition) {
        // update world

        // pause
        pause(Delay);
    }
}
public void run() {
  // setup

  while (condition) {
    // update world

    // pause
    pause(Delay);
  }
}
public void run() {
    // setup

    while (condition) {
        // update world

        // pause
        pause(DELAY);
    }
}
```java
public void run() {
  // setup

  while (condition) {
    // update world
    // pause
    pause(DELAY);
  }
}
```

Each heartbeat, update the world by a small amount (one frame)
public void run() {
    // setup

    while (condition) {
        // update world
        // pause
        pause(Delay);
    }
}
public void run() {
    // setup
    GRect square = makeSquare();

    while (true) {
        // update world
        square.move(1, 0);

        // pause
        pause(DELAY);
    }
}
Graphical Methods

- These methods for graphical objects can be useful for animation:

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>obj.getX()</code></td>
<td>the left x-coordinate of the shape</td>
</tr>
<tr>
<td><code>obj.getY()</code></td>
<td>the top y-coordinate of the shape</td>
</tr>
<tr>
<td><code>obj.getWidth()</code></td>
<td>number of pixels wide the shape is</td>
</tr>
<tr>
<td><code>obj.getHeight()</code></td>
<td>number of pixels tall the shape is</td>
</tr>
<tr>
<td><code>obj.move(dx, dy);</code></td>
<td>adjusts location by the given amount</td>
</tr>
<tr>
<td><code>obj.setLocation(x, y);</code></td>
<td>change the object's x/y position</td>
</tr>
<tr>
<td><code>obj.setSize(w, h);</code></td>
<td>change the object's width*height size</td>
</tr>
</tbody>
</table>

- The GraphicsProgram itself has these methods too:

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>getWidth()</td>
<td>number of pixels wide the window is</td>
</tr>
<tr>
<td>getHeight()</td>
<td>number of pixels tall the window is</td>
</tr>
<tr>
<td>setCanvasSize(w, h)</td>
<td>change the canvas’s width*height size</td>
</tr>
</tbody>
</table>
### Graphical Methods

- These methods for graphical objects can be useful for animation:

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>obj.getX()</code></td>
<td>the left x-coordinate of the shape</td>
</tr>
<tr>
<td><code>obj.getY()</code></td>
<td>the top y-coordinate of the shape</td>
</tr>
<tr>
<td><code>obj.getWidth()</code></td>
<td>number of pixels wide the shape is</td>
</tr>
<tr>
<td><code>obj.getHeight()</code></td>
<td>number of pixels tall the shape is</td>
</tr>
<tr>
<td><code>obj.move(dx, dy);</code></td>
<td>adjusts location by the given amount</td>
</tr>
<tr>
<td><code>obj.setLocation(x, y);</code></td>
<td>change the object's x/y position</td>
</tr>
<tr>
<td><code>obj.setSize(w, h);</code></td>
<td>change the object's width*height size</td>
</tr>
</tbody>
</table>

- The GraphicsProgram itself has these methods too:

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>getWidth()</td>
<td>number of pixels wide the window is</td>
</tr>
<tr>
<td>getHeight()</td>
<td>number of pixels tall the window is</td>
</tr>
<tr>
<td>setCanvasSize(w, h)</td>
<td>change the canvas’s width*height size</td>
</tr>
</tbody>
</table>
Let’s Code It!
Plan for Today

- Review: Graphics
- Boolean Returns
- Animation
- Randomness
- Sand Art
RandomGenerator

// this variable can generate random values
RandomGenerator rgen = RandomGenerator.getInstance();
// this variable can generate random values
RandomGenerator rgen = RandomGenerator.getInstance();
RandomGenerator

// this variable can generate random values
RandomGenerator rgen = RandomGenerator.getInstance();

// random number from 0-9, inclusive
int digit = rgen.nextInt(0, 9);
// this variable can generate random values
RandomGenerator rgen = RandomGenerator.getInstance();

// random number from 0-9, inclusive
int digit = rgen.nextInt(0, 9);

// print “hello” between 3 and 6 times
int numTimes = rgen.nextInt(3, 6);
for (int i = 0; i < numTimes; i++) {
    println(“hello”);
}
// this variable can generate random values
RandomGenerator rgen = RandomGenerator.getInstance();

// random number from 0-9, inclusive
int digit = rgen.nextInt(0, 9);

// print “hello” between 3 and 6 times
int numTimes = rgen.nextInt(3, 6);
for (int i = 0; i < numTimes; i++) {
    println(“hello”);
}

// random Color
Color myColor = rgen.nextColor();
The `RandomGenerator` class defines the following methods:

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>nextInt(int low, int high)</code></td>
<td>Returns a random <code>int</code> between <code>low</code> and <code>high</code>, inclusive.</td>
</tr>
<tr>
<td><code>nextInt(int n)</code></td>
<td>Returns a random <code>int</code> between 0 and <code>n - 1</code>.</td>
</tr>
<tr>
<td><code>nextDouble(double low, double high)</code></td>
<td>Returns a random <code>double</code> <code>d</code> in the range <code>low ≤ d &lt; high</code>.</td>
</tr>
<tr>
<td><code>nextDouble()</code></td>
<td>Returns a random <code>double</code> <code>d</code> in the range <code>0 ≤ d &lt; 1</code>.</td>
</tr>
<tr>
<td><code>nextBoolean()</code></td>
<td>Returns a random <code>boolean</code> value, which is <code>true</code> 50 percent of the time.</td>
</tr>
<tr>
<td><code>nextBoolean(double p)</code></td>
<td>Returns a random <code>boolean</code>, which is <code>true</code> with probability <code>p</code>, where <code>0 ≤ p ≤ 1</code>.</td>
</tr>
<tr>
<td><code>nextColor()</code></td>
<td>Returns a random color.</td>
</tr>
</tbody>
</table>
Let's make some random numbers!
Two powerful tools to create awesome programs.
Plan for Today

- Review: Graphics
- Boolean Returns
- Animation
- Randomness
- Sand Art
Sand Art

https://craftingagreenworld.com/articles/create-sand-art-in-vintage-bottles/
Sand Art

CS106A edition :)

---

38
Milestone #1
Let’s Code It!
Plan for Today

- Review: Graphics
- Boolean Returns
- Animation
- Randomness
- Sand Art

**Next time:** Interactive Graphics Programs