

Animation and Randomness

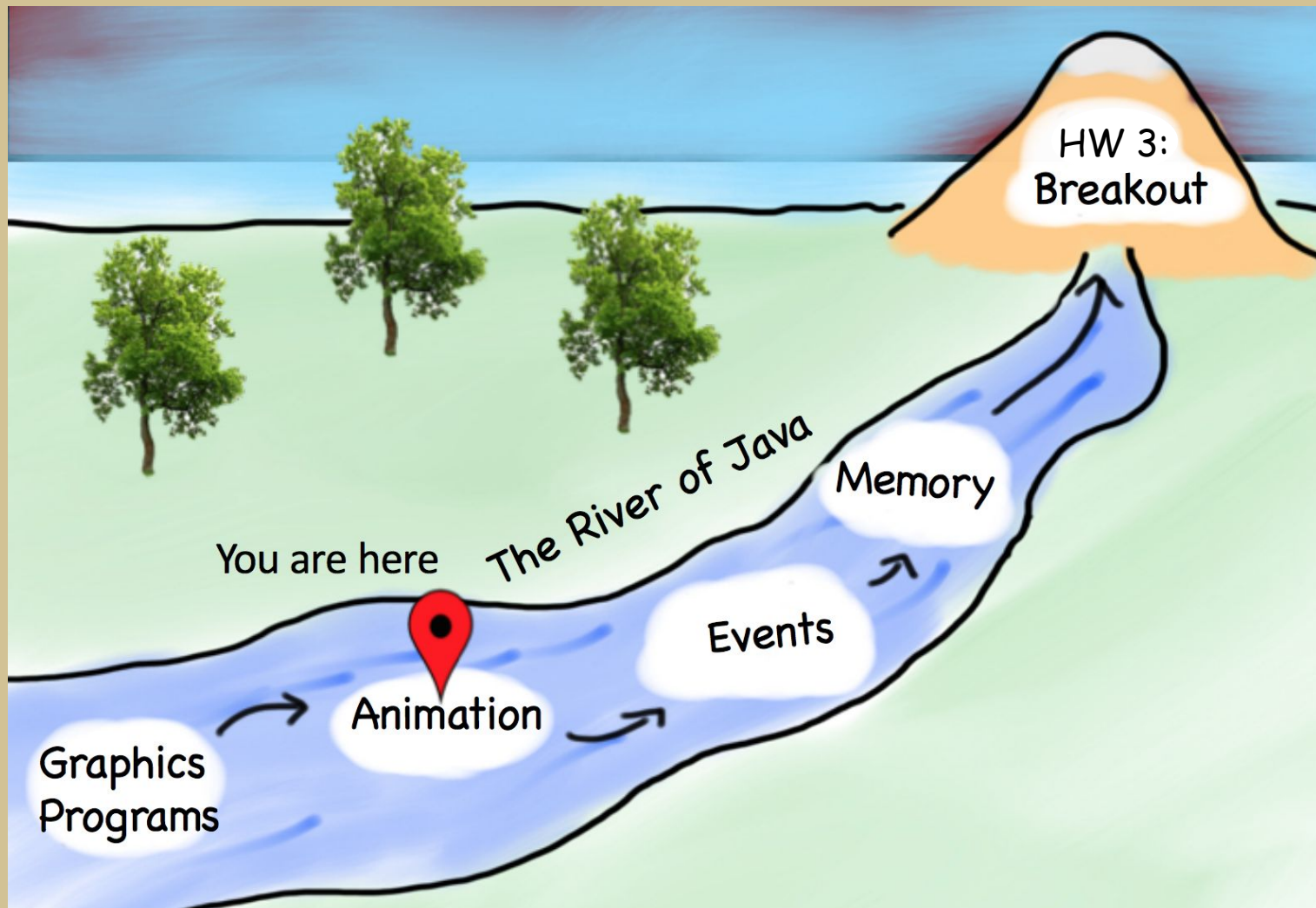
Lecture 9

CS106A, Summer 2019

Sarai Gould && Laura Cruz-Albrecht

With inspiration from slides created by Keith Schwarz, Mehran Sahami, Eric Roberts, Stuart Reges, Chris Piech and others.





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

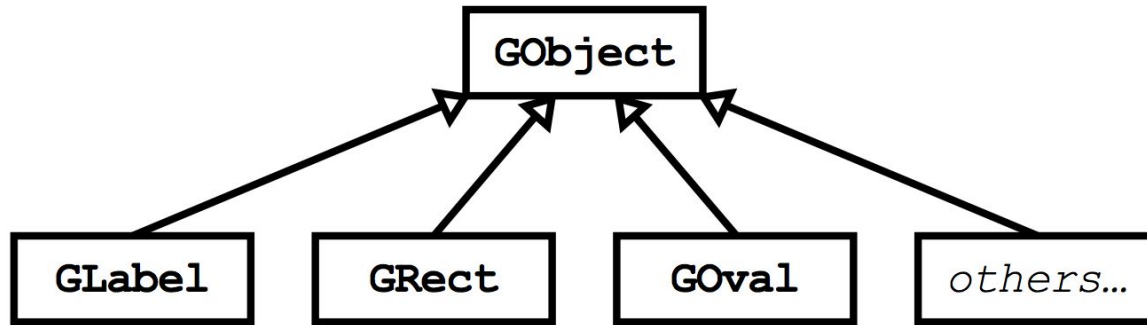
Plan for Today

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

Review: Working with Graphics



We make graphics programs by creating **graphics objects** and manipulating their properties.



Review: Working with Graphics

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

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



Review: Working with Graphics

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

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



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

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

who?

what?

what specifically?

Plan for Today

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

Is Square

```
public void run() {  
    for (int i = 1; i <= 100; i++) {  
        if ( /* check if i is a square number */ ) {  
            println(i);  
        }  
    }  
}
```

Is Square

```
public void run() {  
    for (int i = 1; i <= 100; i++) {  
        if (isSquare(i)) {  
            println(i);  
        }  
    }  
}
```

Boolean Return

```
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;  
    }  
}
```

Boolean Return

```
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;  
    }  
}
```



Boolean Return

```
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);  
}
```

Boolean Return

```
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

Animation Loop

```
public void run() {  
    // setup  
  
    while (condition) {  
        // update world  
  
        // pause  
        pause(DELAY);  
    }  
}
```

Animation Loop

```
public void run() {  
    // setup  
  
    while (condition) {  
        // update world  
  
        // pause  
        pause(DELAY);  
    }  
}
```

Make all the variables you need. Add graphics to the screen.

Animation Loop

```
public void run() {  
    // setup  
  
    while (condition) {  
        // update world  
  
        // pause  
        pause(DELAY);  
    }  
}
```

The animation loop is a repetition of “heartbeats”

Animation Loop

```
public void run() {  
    // setup  
  
    while (condition) {  
        // update world  
  
        // pause  
        pause(DELAY);  
    }  
}
```

Each heartbeat, update the world by a small amount (one frame)

Animation Loop

```
public void run() {  
    // setup  
  
    while (condition) {  
        // update world  
  
        // pause  
        pause(DELAY);  
    }  
}
```

Pause so humans can see it :)

Simple Animation

```
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:

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

- The GraphicsProgram itself has these methods too:

<code>getWidth()</code>	number of pixels wide the window is
<code>getHeight()</code>	number of pixels tall the window is
<code>setCanvasSize(w, h)</code>	change the canvas's width*height size

Graphical Methods

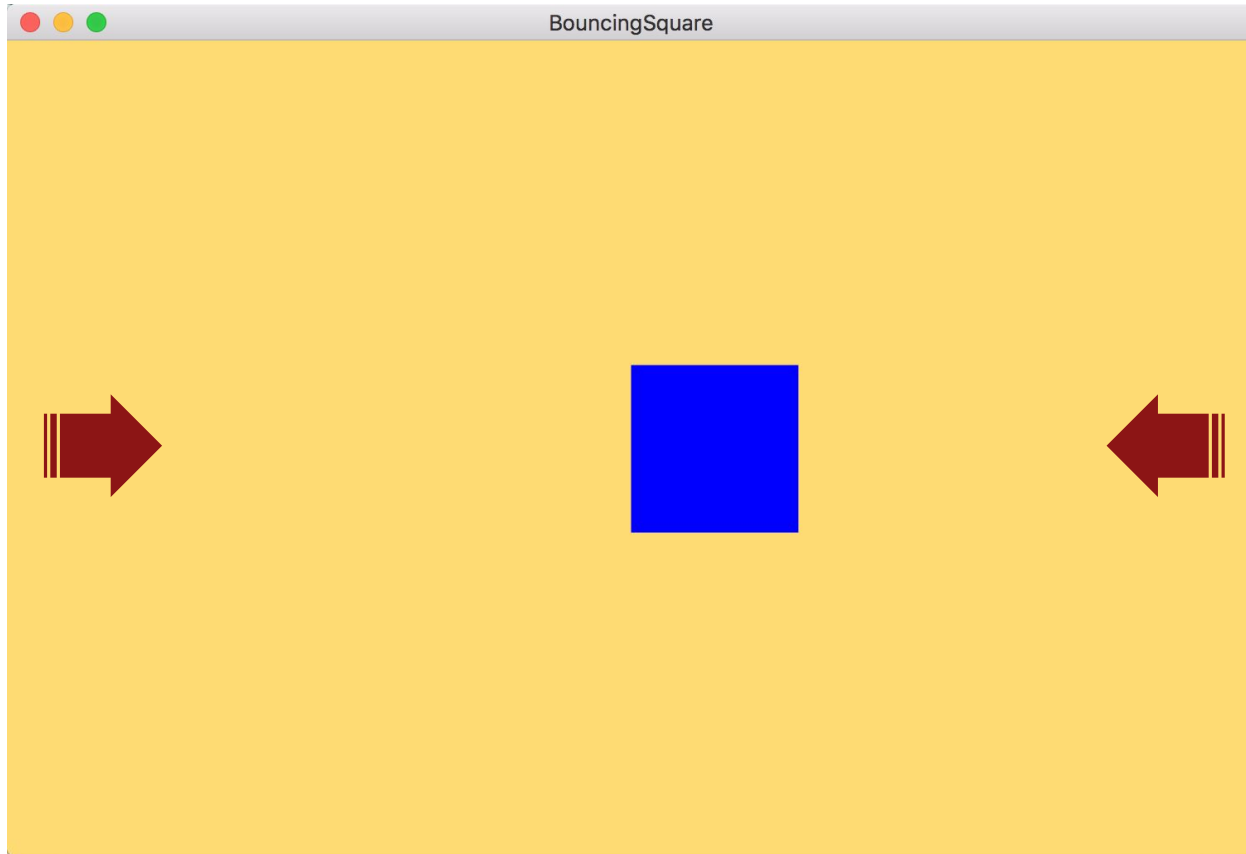
- These methods for graphical objects can be useful for animation:

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

- The GraphicsProgram itself has these methods too:

<code>getWidth()</code>	number of pixels wide the window is
<code>getHeight()</code>	number of pixels tall the window is
<code>setCanvasSize(w, h)</code>	change the canvas's width*height size

Bouncing Square



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();
```

RandomGenerator

```
// 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);
```

RandomGenerator

```
// 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");
}
```

RandomGenerator

```
// 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();
```


RandomGenerator

The RandomGenerator class defines the following methods:

int nextInt(int low, int high)

Returns a random **int** between **low** and **high**, inclusive.

int nextInt(int n)

Returns a random **int** between 0 and **n** - 1.

double nextDouble(double low, double high)

Returns a random **double** d in the range $\text{low} \leq d < \text{high}$.

double nextDouble()

Returns a random **double** d in the range $0 \leq d < 1$.

boolean nextBoolean()

Returns a random **boolean** value, which is **true** 50 percent of the time.

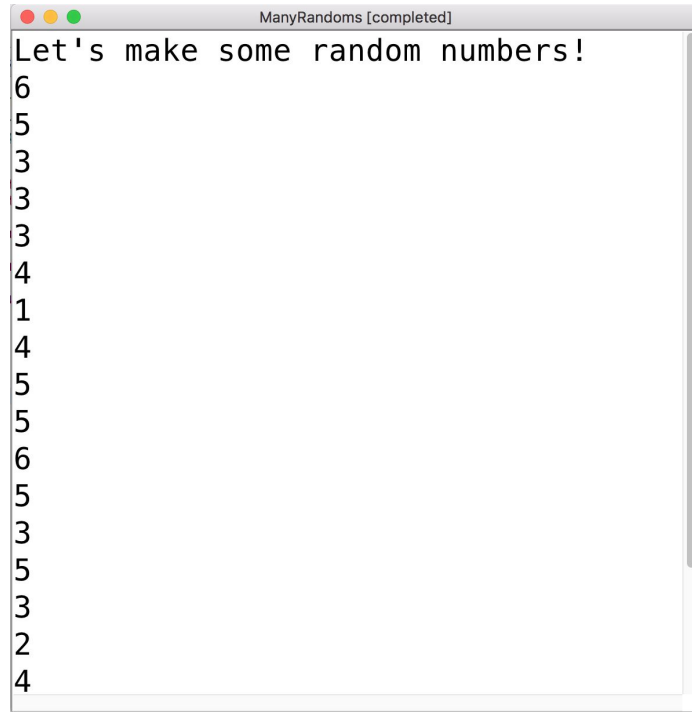
boolean nextBoolean(double p)

Returns a random **boolean**, which is **true** with probability **p**, where $0 \leq p \leq 1$.

Color nextColor()

Returns a random color.

RandomGenerator



```
ManyRandoms [completed]
Let's make some random numbers!
6
5
3
3
3
4
1
4
5
5
6
5
3
5
3
2
2
4
```

Animation + Randomness

Two powerful tools to create
awesome programs.

Plan for Today

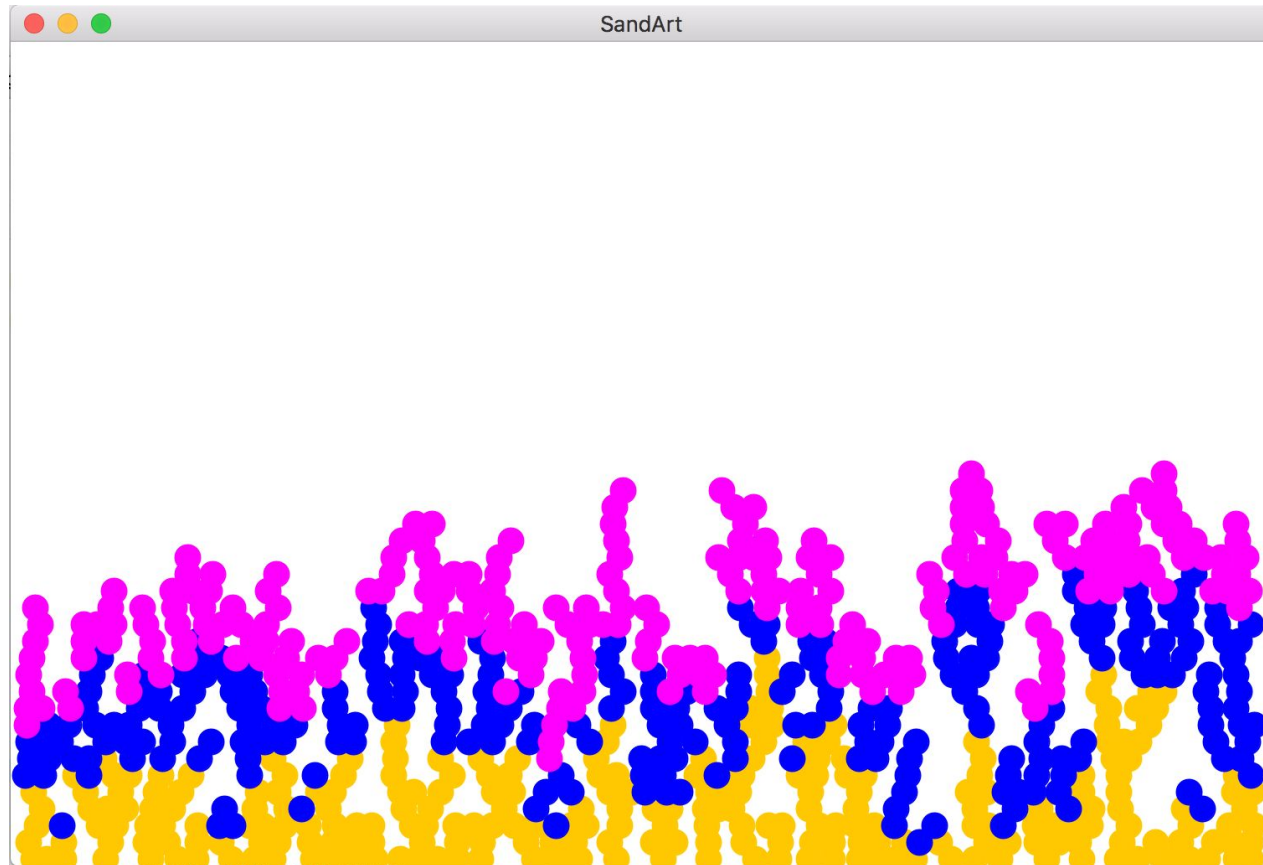
- Review: Graphics
- Boolean Returns
- Animation
- Randomness
- Sand Art

Sand Art

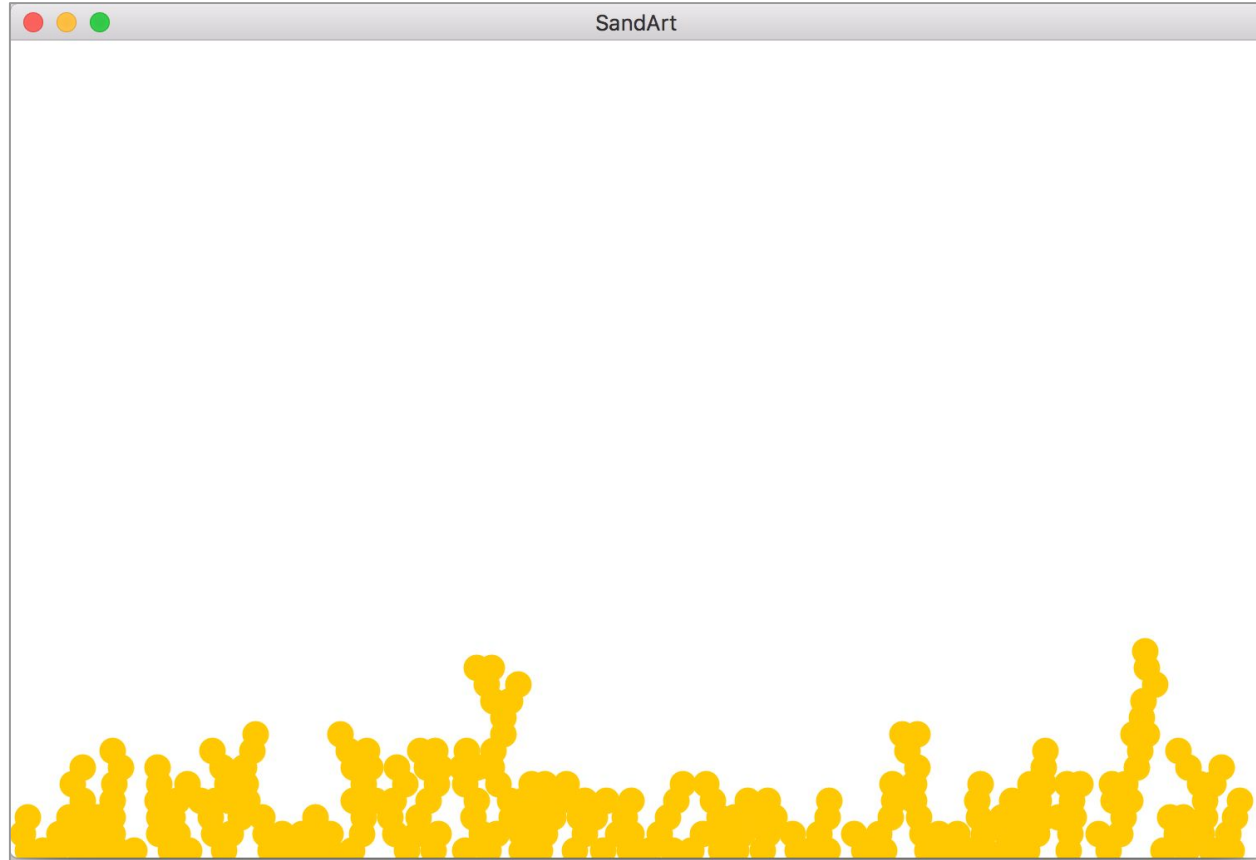


Sand Art

CS106A
edition :)



Milestone #1



Let's Code It!

Plan for Today

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

Next time: Interactive Graphics Programs