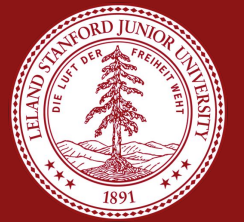
Data Structures: Bringing It All Together

Lecture 20

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

- Blank code is available to download from the website.

- Assignment 5: Image Shop was released yesterday.
  - Due Aug. 5th at 10am

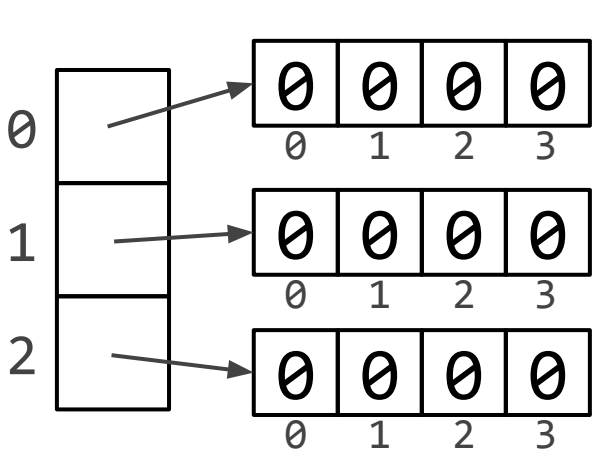
Plan for Today

- Review: Arrays, 2D Arrays, ArrayLists
- Putting ArrayLists inside of HashMaps
- Example: Ice Cream Anyone?
- Example: Picking Berries!

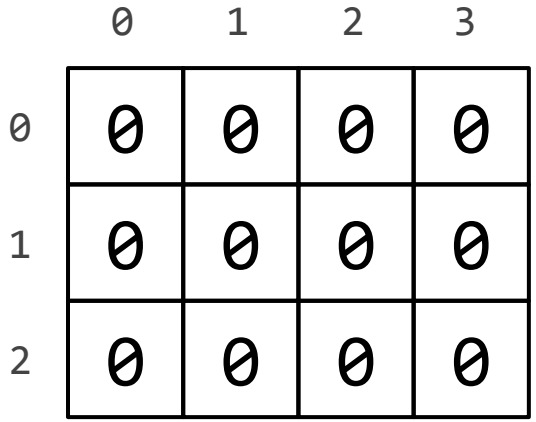
Review 2D: Arrays

Review 2D: Arrays

```
int[][] matrix = new int[3][4];
```



An array of arrays



A unit (ie, a grid/matrix)

Reivew: 2D Arrays ❤️ For Loops

- The canonical way to loop over a 2D array is with a double `for` loop

```
type[][] arr = ...
for (int row = 0; row < arr.length; row++) {
    for (int col = 0; col < arr[0].length; col++) {
        // do something with arr[row][col] ...
    }
}
```

Our First ArrayList

Type of thing your ArrayList will store

Same type here.

```
ArrayList<String> myArrayList = new ArrayList<String>();
```

Looping over ArrayLists

```
ArrayList<String> myArrayList = new ArrayList<String>();

// Adds elements to the back
myArrayList.add("hi");
myArrayList.add("there");

// Access elements by index (starting at 0!)
for (int i = 0; i < myArrayList.size(); i++) {
     String str = myArrayList.get(i);
     println(str);
}

// A beautiful way to access each element
for (String str : myArrayList) {
     println(str);
}
```

Our First HashMap

Data Type of "keys" in our HashMap

Data Type of "values" in our HashMap

Repeated types of key->value pairs

```
HashMap<String, String> firstMap = new HashMap<String, String>();
```

Review: HashMap Commands

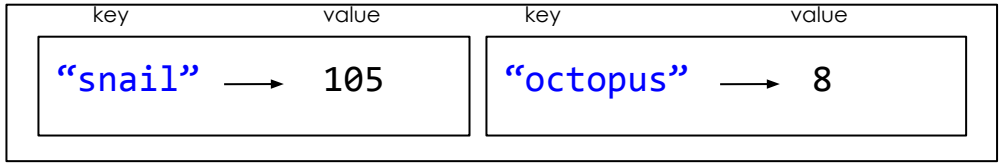
```
HashMap<String, String> myMap = new HashMap<String, String>();

// Adds key->value pairs
myMap.put("dog", "woof");
myMap.put("cat", "meow");

// Removes key and the associated value
myMap.remove("dog", "woof");

// A beautiful way to access each key
for (String key : myMap.keySet()) {
     println(key);
     // print the value associated with the key
     println(myMap.get(key));
}
```

Review: What Does This Code Do?

```
import java.util.*;
HashMap<String, Integer> numLimbs = new HashMap<>();
numLimbs.put("dog", 4);
numLimbs.remove("dog");
numLimbs.put("snail", 1);
numLimbs.put("snail", 105);
numLimbs.put("octopus", 8);
println(numLimbs.get("dog"));
println(numLimbs.get("snail"));
println(numLimbs.get("octopus"));
```

```
null
105
8
```

numLimbs



Learning Goal for Today

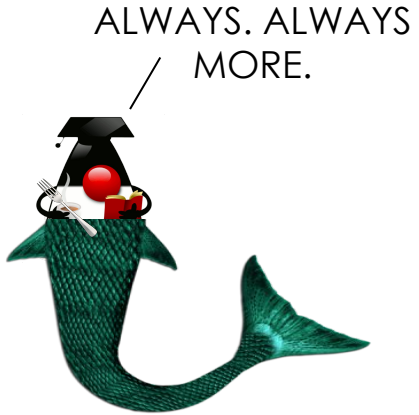
Combining many of the Data Structures we've learned!

2D Arrays, HashMaps, ArrayLists, and Graphics

Can we Store Even MORE Data?

So, we can associate *one* piece of data with *one other* piece of data.

Can we associate *multiple* pieces of data with *multiple other* pieces of data?



Of Course We Can!

Look, an ArrayList of Strings!

```
HashMap<String, ArrayList<String>> firstMap = new HashMap<>();
```

Our First HashMap (Again)

Data Type of "keys"
in our HashMap

```
HashMap<String, ArrayList<String>> firstMap = new HashMap<>();
```

Our First HashMap (Again)

Data Type of "values"
in our HashMap


```
HashMap<String, ArrayList<String>> firstMap = new HashMap<>();
```

Our First HashMap (Again)

We can leave this blank because of type inference!

```
HashMap<String, ArrayList<String>> firstMap = new HashMap<>();
```

Our First HashMap (Again)

```
import java.util.*;


HashMap<String, ArrayList<String>> firstMap = new HashMap<>();
```

Our First HashMap (Again)

```
import java.util.*;

HashMap<String, ArrayList<String>> firstMap = new HashMap<>();
```

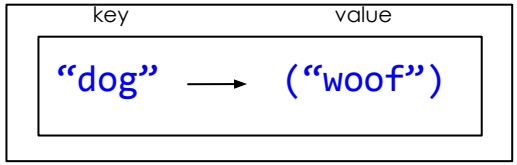
Let's create a HashMap that maps from animal names to animal sounds!

Basically, we're creating a dictionary where, if we look up an animal's name, it will tell us which sound it makes!

Our First HashMap (Again)

```
import java.util.*;

HashMap<String, ArrayList<String>> animalSounds = new HashMap<>();

ArrayList<String>> dogSounds = new ArrayList<>();

dogSounds.add("woof");

animalSounds.put("dog", dogSounds);
```

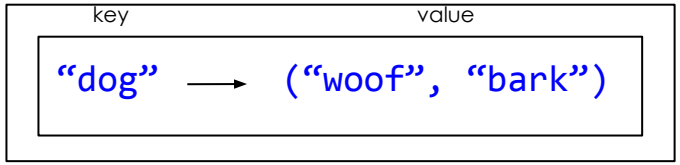
sounds



Our First HashMap (Again)

```
import java.util.*;

HashMap<String, ArrayList<String>> animalSounds = new HashMap<>();

ArrayList<String>> dogSounds = new ArrayList<>();

dogSounds.add(“woof”);

dogSounds.add(“bark”);

animalSounds.put(“dog”, dogSounds);
```

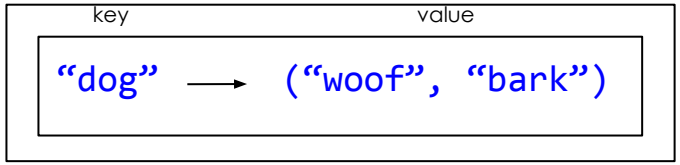
sounds



Our First HashMap (Again)

```
...
animalSounds.put("dog", dogSounds);

...
//("woof", "bark")
ArrayList<String> currSounds = animalSounds.get("dog");
```

sounds

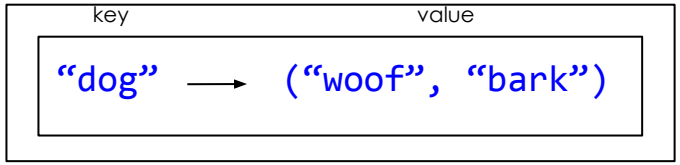


Our First HashMap (Again)

```
...
animalSounds.put("dog", dogSounds);

...
//("woof", "bark")
ArrayList<String> currSounds = animalSounds.get("dog");

for(String sound : currSounds){
     println(sound);
}
```

```
woof
bark
```

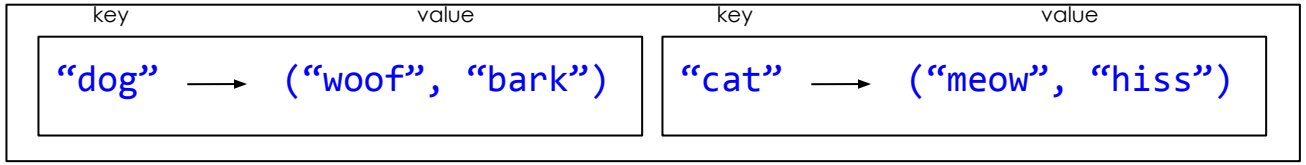
sounds



What Happens Here?

```
...
animalSounds.put("cat", catSounds); // (meow, hiss)

...
for(String animal : animalSounds.getKeySet()){

    ArrayList<String> sounds = animalSounds.get(animal);

    for(String sound : sounds){
        println(sound);
    }
}
```

sounds



Our First HashMap (Again)

```
...
animalSounds.put("cat", catSounds); // (meow, hiss)

...
for(String animal : animalSounds.getKeySet()){

    ArrayList<String> sounds = animalSounds.get(animal);

    for(String sound : sounds){
        println(sound);
    }
}
```

animal

"dog"

sounds

key	value	key	value
"dog"	("woof", "bark")	"cat"	("meow", "hiss")

Our First HashMap (Again)

```
...
animalSounds.put("cat", catSounds); // (meow, hiss)

...
for(String animal : animalSounds.getKeySet()){

    ArrayList<String> sounds = animalSounds.get(animal);

    for(String sound : sounds){
        println(sound);
    }
}
```

animal

"dog"

sounds

key	value	key	value
"dog"	("woof", "bark")	"cat"	("meow", "hiss")

Our First HashMap (Again)

```
...
animalSounds.put("cat", catSounds); // (meow, hiss)

...
for(String animal : animalSounds.getKeySet()){

    ArrayList<String> sounds = animalSounds.get(animal);

    for(String sound : sounds){
        println(sound);
    }
}
```

animal

"dog"

sound

"woof"

sounds

key	value	key	value
"dog"	("woof", "bark")	"cat"	("meow", "hiss")

Our First HashMap (Again)

```
...
animalSounds.put("cat", catSounds); // (meow, hiss)

...
for(String animal : animalSounds.getKeySet()){

    ArrayList<String> sounds = animalSounds.get(animal);

    for(String sound : sounds){
        println(sound);
    }
}
```

```
woof
```

animal | sound

"dog"	"woof"

sounds

key	value	key	value
"dog"	("woof", "bark")	"cat"	("meow", "hiss")

Our First HashMap (Again)

```
...
animalSounds.put("cat", catSounds); // (meow, hiss)

...
for(String animal : animalSounds.getKeySet()){

    ArrayList<String> sounds = animalSounds.get(animal);

    for(String sound : sounds){
        println(sound);
    }
}
```

```
woof
```

animal | sound

"dog"	"bark"

sounds

key	value	key	value
"dog"	("woof", "bark")	"cat"	("meow", "hiss")

Our First HashMap (Again)

```
...
animalSounds.put("cat", catSounds); // (meow, hiss)

...
for(String animal : animalSounds.getKeySet()){

    ArrayList<String> sounds = animalSounds.get(animal);

    for(String sound : sounds){
        println(sound);
    }
}
```

```
woof
bark
```

animal	sound
"dog"	"bark"

sounds

key	value	key	value
"dog"	("woof", "bark")	"cat"	("meow", "hiss")

Our First HashMap (Again)

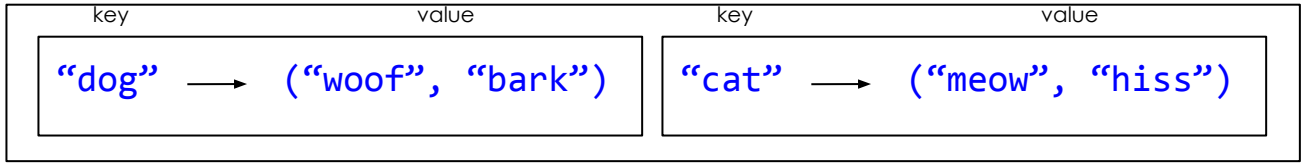
```
...
animalSounds.put("cat", catSounds); // (meow, hiss)

...
for(String animal : animalSounds.getKeySet()){

    ArrayList<String> sounds = animalSounds.get(animal);

    for(String sound : sounds){
        println(sound);
    }
}
```

```
woof
bark
```

animal

"cat"

sounds



Our First HashMap (Again)

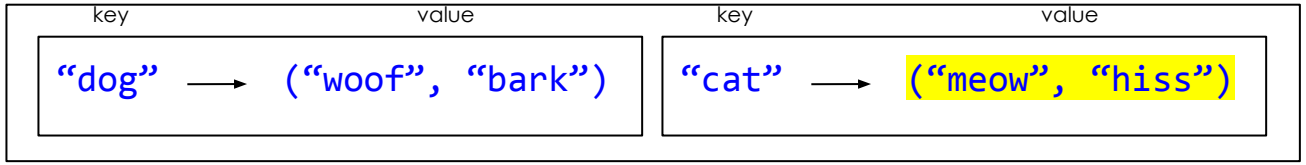
```
...
animalSounds.put("cat", catSounds); // (meow, hiss)

...
for(String animal : animalSounds.getKeySet()){

    ArrayList<String> sounds = animalSounds.get(animal);

    for(String sound : sounds){
        println(sound);
    }
}
```

```
woof
bark
```

animal

"cat"

sounds



Our First HashMap (Again)

```
...
animalSounds.put("cat", catSounds); // (meow, hiss)

...
for(String animal : animalSounds.getKeySet()){

    ArrayList<String> sounds = animalSounds.get(animal);

    for(String sound : sounds){
        println(sound);
    }
}
```

```
woof
bark
```

animal | sound

"cat"	"meow"

sounds

key	value	key	value
"dog"	("woof", "bark")	"cat"	("meow", "hiss")

Our First HashMap (Again)

```
...
animalSounds.put("cat", catSounds); // (meow, hiss)

...
for(String animal : animalSounds.getKeySet()){

    ArrayList<String> sounds = animalSounds.get(animal);

    for(String sound : sounds){
        println(sound);
    }
}
```

```
woof
bark
meow
```

animal	sound
"cat"	"meow"

sounds

key	value	key	value
"dog"	("woof", "bark")	"cat"	("meow", "hiss")

Our First HashMap (Again)

```
...
animalSounds.put("cat", catSounds); // (meow, hiss)

...
for(String animal : animalSounds.getKeySet()){

    ArrayList<String> sounds = animalSounds.get(animal);

    for(String sound : sounds){
        println(sound);
    }
}
```

```
woof
bark
meow
```

animal | sound

"cat"	"hiss"

sounds

key	value	key	value
"dog"	("woof", "bark")	"cat"	("meow", "hiss")

Our First HashMap (Again)

```
...
animalSounds.put("cat", catSounds); // (meow, hiss)

...
for(String animal : animalSounds.getKeySet()){

    ArrayList<String> sounds = animalSounds.get(animal);

    for(String sound : sounds){
        println(sound);
    }
}
```

```
woof
bark
meow
hiss
```

animal | sound

"cat"	"hiss"

sounds

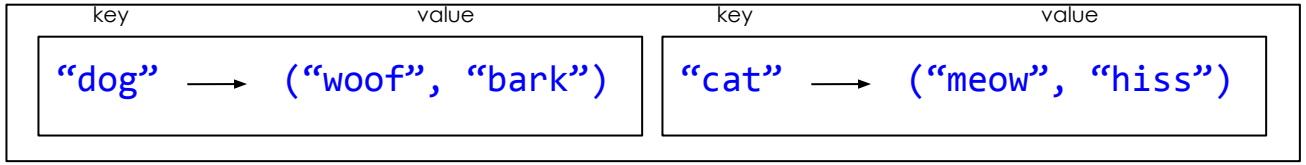
key	value	key	value
"dog"	("woof", "bark")	"cat"	("meow", "hiss")

Our First HashMap (Again)

```
...
animalSounds.put("cat", catSounds); // (meow, hiss)

...
for(String animal : animalSounds.getKeySet()){

    ArrayList<String> sounds = animalSounds.get(animal);

    for(String sound : sounds){
        println(sound);
    }
}
```

```
woof
bark
meow
hiss
```

sounds



Time for a Break with Duke!*

Want to get ice cream?! 🍨😊🍦

* Okay, we're actually writing more code...

Finding a Common Time

Let's write a program that will find a common time that we can all meet for ice cream!

You've probably seen a program that does something like this before. Now we're going to write it!

Finding a Common Time

```
Let's go get ice cream! When are you free?

What's your name? Sarai
Format availabilities like: 1:20AM or 12:30PM
Input one time you're free to get ice cream: 1:30AM
Input another time you're free: 2:30PM
Input another time you're free: 3:45PM
Input another time you're free: 5:00PM
Input another time you're free:
What's your name? Laura
Format availabilities like: 1:20AM or 12:30PM
Input one time you're free to get ice cream: 2:30PM
Input another time you're free: 1:00AM
Input another time you're free: 5:00PM
Input another time you're free:
What's your name? Karel
Format availabilities like: 1:20AM or 12:30PM
Input one time you're free to get ice cream: 5:00PM
Input another time you're free: 3:45AM
Input another time you're free:
What's your name?
Sarai is available at:
1:30AM
2:30PM
3:45PM
5:00PM
Laura is available at:
2:30PM
1:00AM
5:00PM
Karel is available at:
5:00PM
3:45AM
You friends are all available at:
5:00PM
```

Let’s Code It!

Code: Our Ice Cream Program

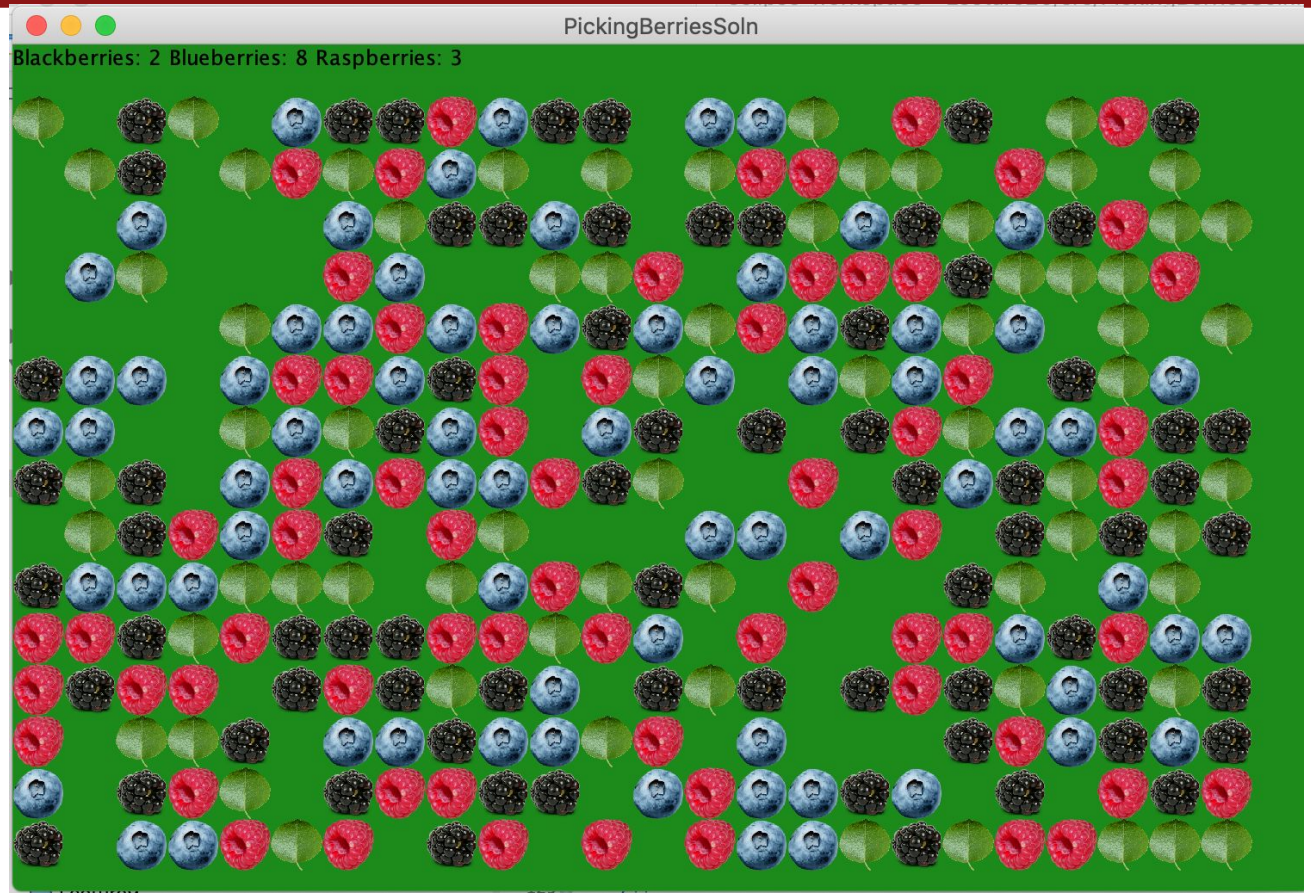
```
public void run(){
    HashMap<String, ArrayList<String>> friendAvailability = new HashMap<String, ArrayList<String>>();
    ArrayList<String> bestTimes = new ArrayList<String>();

    println("Let's go get ice cream! When are you free? \n");

    // Add friends (fencepost)
    String name = readLine("What's your name?");
    while(!name.equals(SENTINEL)) {
        ArrayList<String> allAvailabilities = new ArrayList<String>();

        // Add friend availability (fencepost)
        println("Format availabilities like: 1:20AM or 12:30PM");
        String oneAvailability = readLine("Input one time you're free to get ice cream: ");
        while(!oneAvailability.equals(SENTINEL)) {
            allAvailabilities.add(oneAvailability);
            oneAvailability = readLine("Input another time you're free: ");
        }

        // Add friend and their availability to the map
        friendAvailability.put(name, allAvailabilities);
        if(friendAvailability.size() == 1) {
            bestTimes.addAll(allAvailabilities);
        } else {
            // this only saves the times these two ArryLists share
            bestTimes.retainAll(allAvailabilities);
        }

        name = readLine("What's your name?");

    }

    // Loop through friends
    for(String currFriend : friendAvailability.keySet()) {
        println(currFriend + " is available at: ");

        ArrayList<String> currFriendAvailabilities = friendAvailability.get(currFriend);
        // Loop through friends' availability and print it!
        for(String time : currFriendAvailabilities) {
            println(time);
        }
    }

    // prints best times!
    println("You friends are all available at: ");
    for(String time: bestTimes) {

        println(time);
    }
```

Picking Berries

I always like toppings for my ice cream!

Let's pick some berries!

Picking Berries!



Let’s Code It!

Code: Picking Berries (Part 1)

```
/* Constants for our program. We shouldn't need to hard code in anything! */
private static final int NONE = 0;
private static final int LEAF = 1;
private static final int RASPBERRY = 2;
private static final int BLUEBERRY = 3;
private static final int BLACKBERRY = 4;

private static final int ROW = 15;
private static final int COL = 24;

private static final String LEAF_IMAGE = "leaf.png";
private static final String RASPBERRY_IMAGE = "raspberry.png";
private static final String BLUEBERRY_IMAGE = "blueberry.png";
private static final String BLACKBERRY_IMAGE = "blackberry.png";
private static final int IMAGE_WIDTH = 30;

/* Instance variables so we can edit our bush, berryCounts, and label inside of mouseClicked. */
private int[][] bush = new int[ROW][COL];
private HashMap<Integer, Integer> berryCounts = new HashMap<>();
private GLabel berryLabel = new GLabel("");

public void run(){
    setup();
    RandomGenerator rg = new RandomGenerator();

    for(int row = 0; row < numRows(bush); row++) {
        for(int col = 0; col < numCols(bush); col++) {
            bush[row][col] = rg.nextInt(0, 4);
        }
    }

    // iterate through rows
    for(int row = 0; row < numRows(bush); row++) {
        // iterate through columns
        for(int col = 0; col < numCols(bush); col++) {
            int fruitNum = bush[row][col];
            GImage currFruitImage = createFruitImage(fruitNum);
            if (currFruitImage != null) {
                add(currFruitImage, col * IMAGE_WIDTH, row * IMAGE_WIDTH + IMAGE_WIDTH);
            }
        }
    }
}
```

Code: Picking Berries (Part 2)

```
public void mouseClicked(MouseEvent e) {
    int mouseX = e.getX();
    int mouseY = e.getY();

    GImage fruitImage = getElementAt(mouseX, mouseY);

    if(fruitImage != null){
        int row = calculateRow(fruitImage);
        int col = calculateCol(fruitImage);
        int fruitNum = bush[row][col];

        if(fruitNum != LEAF) {

            if(berryCounts.containsKey(fruitNum)) {
                int newCount = berryCounts.get(fruitNum) + 1;
                berryCounts.put(fruitNum, newCount);
            } else {
                berryCounts.put(fruitNum, 1);
            }
            bush[row][col] = NONE;
            remove(fruitImage);
            updateLabel();
        }
    }
}

private void updateLabel() {
    String labelString = "";
    if(berryCounts.containsKey(BLACKBERRY)) {
        labelString += "Blackberries: " + berryCounts.get(BLACKBERRY) + "\t";
    }
    if(berryCounts.containsKey(BLUEBERRY)) {
        labelString += "Blueberries: " + berryCounts.get(BLUEBERRY) + "\t";
    }
    if(berryCounts.containsKey(RASPBERRY)) {
        labelString += "Raspberries: " + berryCounts.get(RASPBERRY) + "\t";
    }
    berryLabel.setLabel(labelString);
}
```

Plan for Today

- Review: Arrays, 2D Arrays, ArrayLists
- Putting ArrayLists inside of HashMaps
- Example: Ice Cream Anyone?
- Example: Picking Berries!

Next Time: Classes