

CS106A

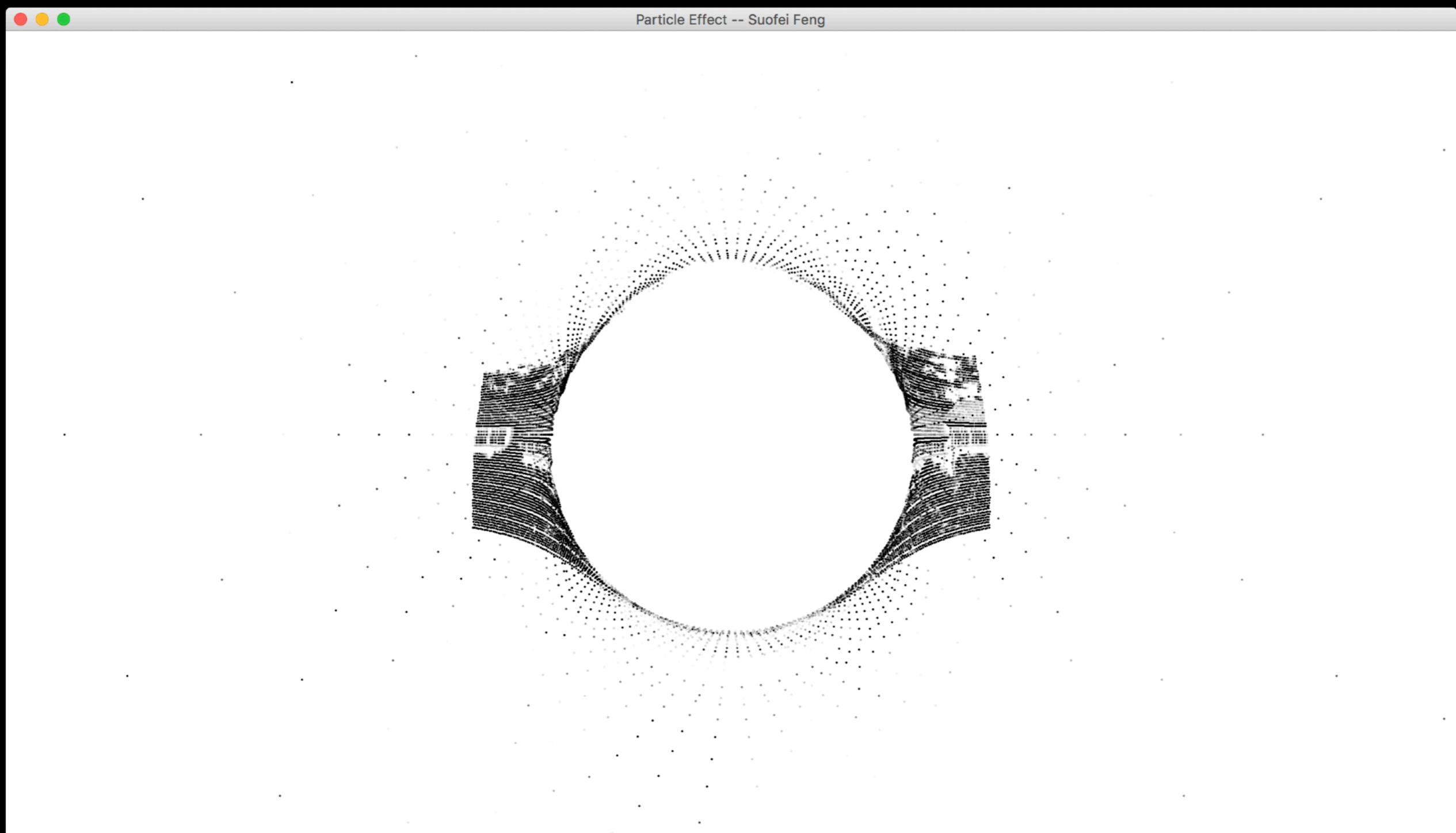
Final Exam Review

Part 2: complements Monday's review lecture

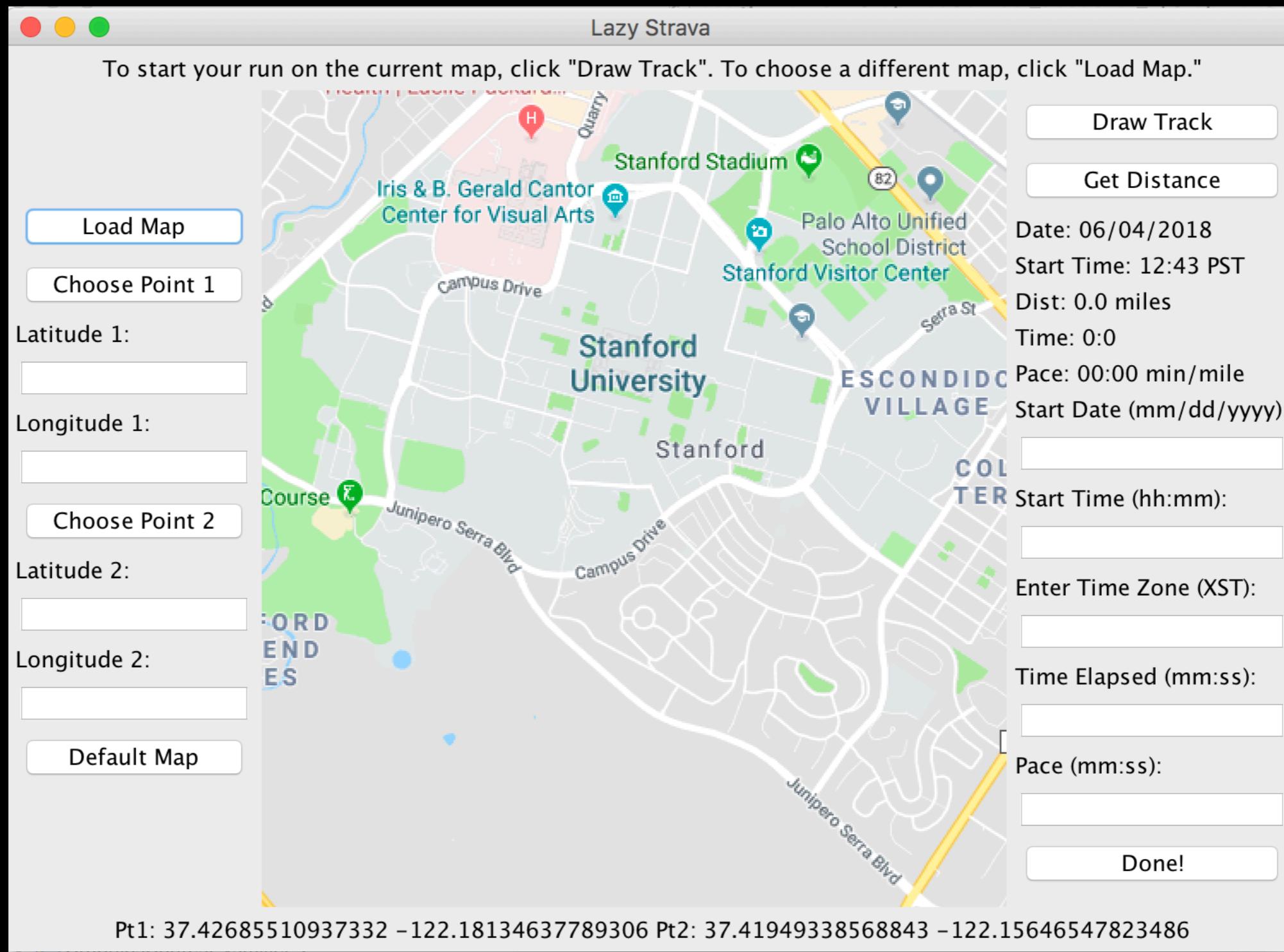
Julia Daniel

Graphics Contest!

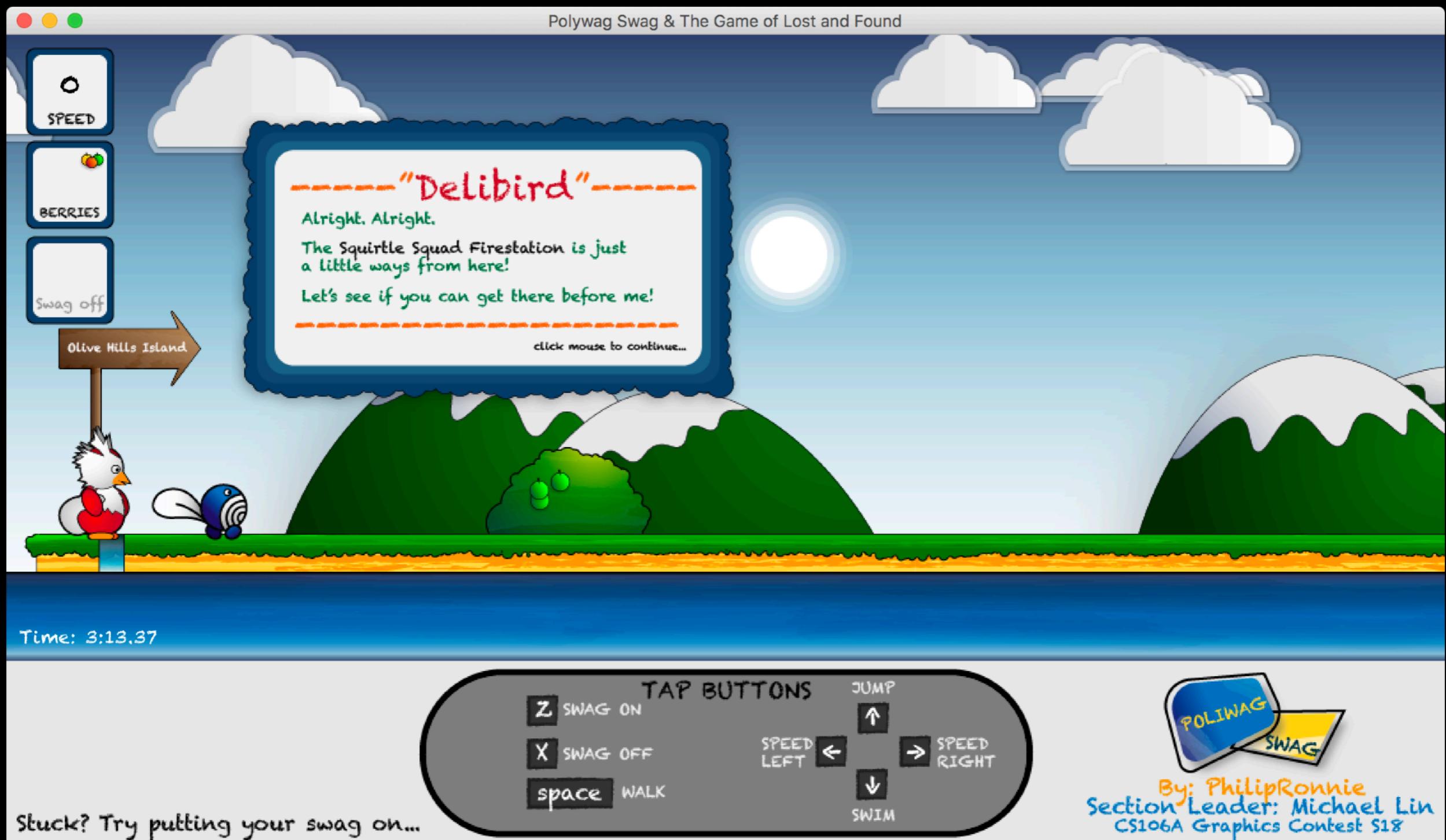
Graphics Contest: Suofei Feng



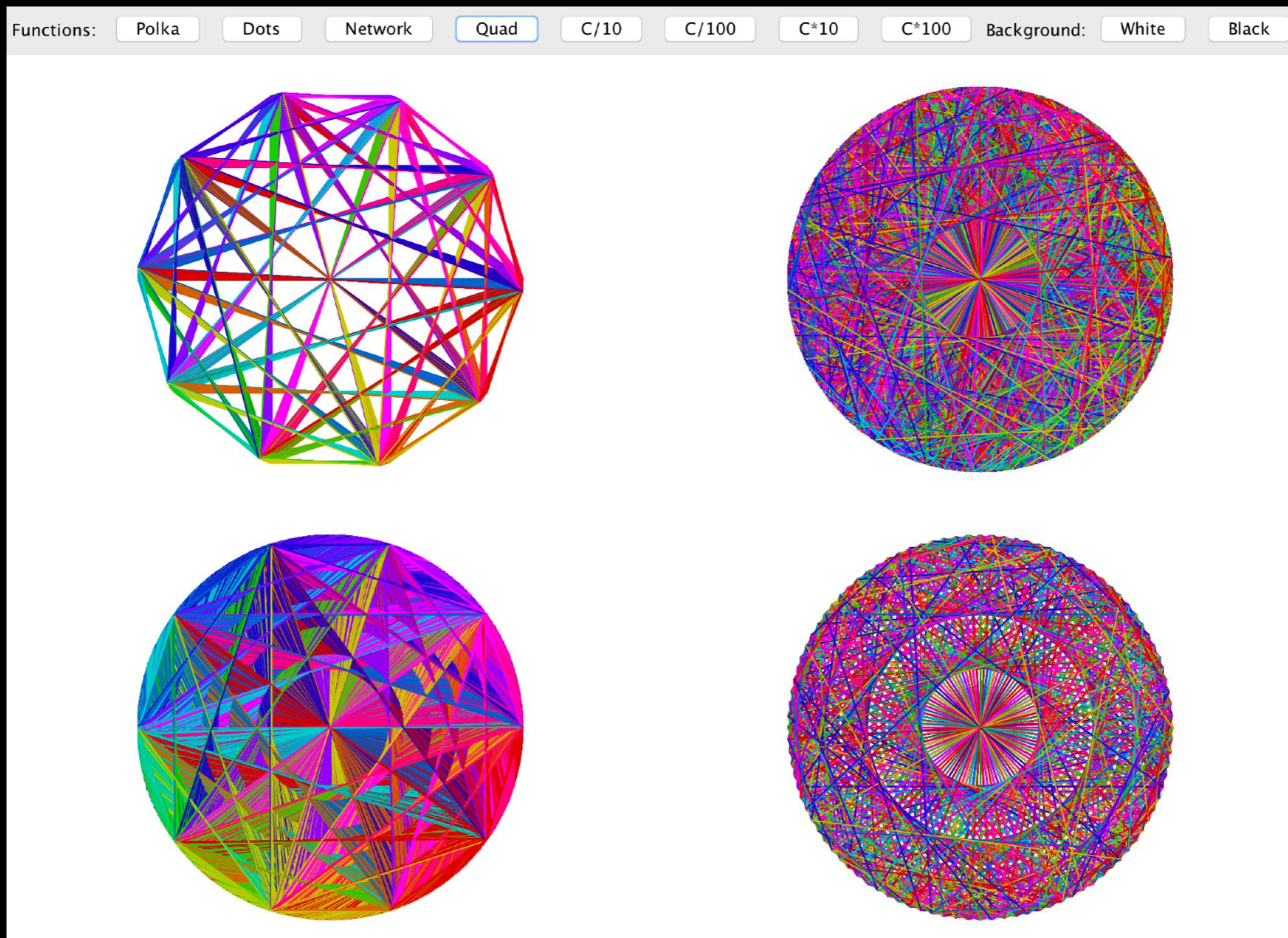
Graphics Contest: Ellen Bouchard



Graphics Contest: Phil Quimba



Graphics Contest: Alex Lin



Overview

- Graphical Programs
 - Graphics
 - Interactors
- Data Storage
 - Strings
 - HashMaps
 - Classes
 - Internet

Overview

- **Graphical Programs**
 - **Graphics**
 - **Interactors**
- Data Storage
 - Strings
 - HashMaps
 - Classes
 - Internet

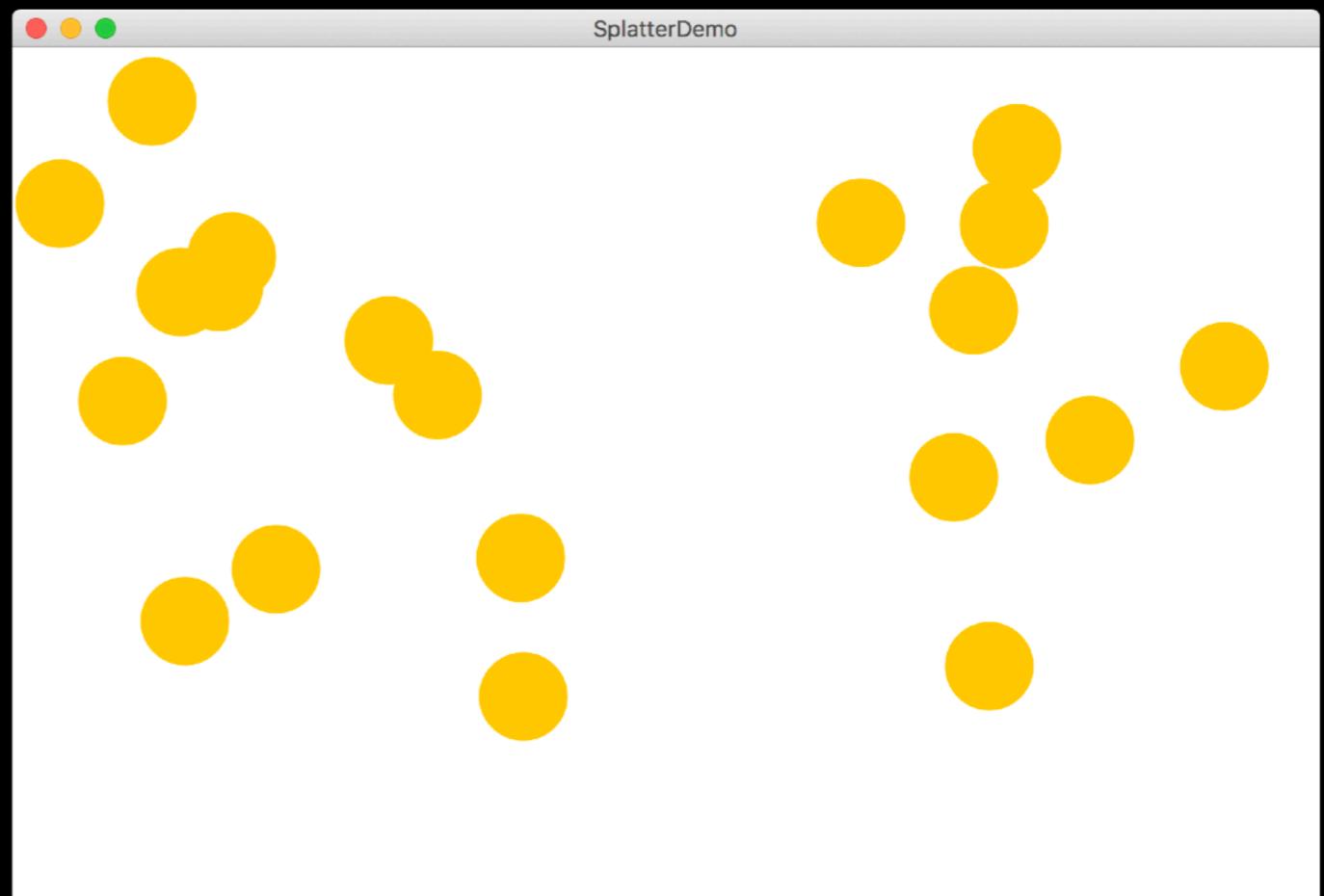
Graphics

Graphics

- Graphics programs have a GCanvas that can have GObjects added to it
GCanvas has dimensions getWidth() x getHeight()
- GObjects: GOval, GRect, GLabel, GLine...
- Methods: setFilled, setColor, add

Graphics example: splatter

- Write a **splatter** method that takes in an int **n** and draws **n** filled orange circles on the screen
- Each circle should be 50×50 px, in a random location such that no circle goes over the edge of the canvas



Graphics example: splatter

```
private void splatter(int n) {  
    for (int i = 0; i < n; i++) {  
        GOval paint = new GOval(50, 50);  
        paint.setFilled(true);  
        paint.setColor(Color.ORANGE);  
        double randX = rgen.nextDouble(0, getWidth() - 50);  
        double randY = rgen.nextDouble(0, getHeight() - 50);  
        add(paint, randX, randY);  
    }  
}
```

Interactors

Interactors

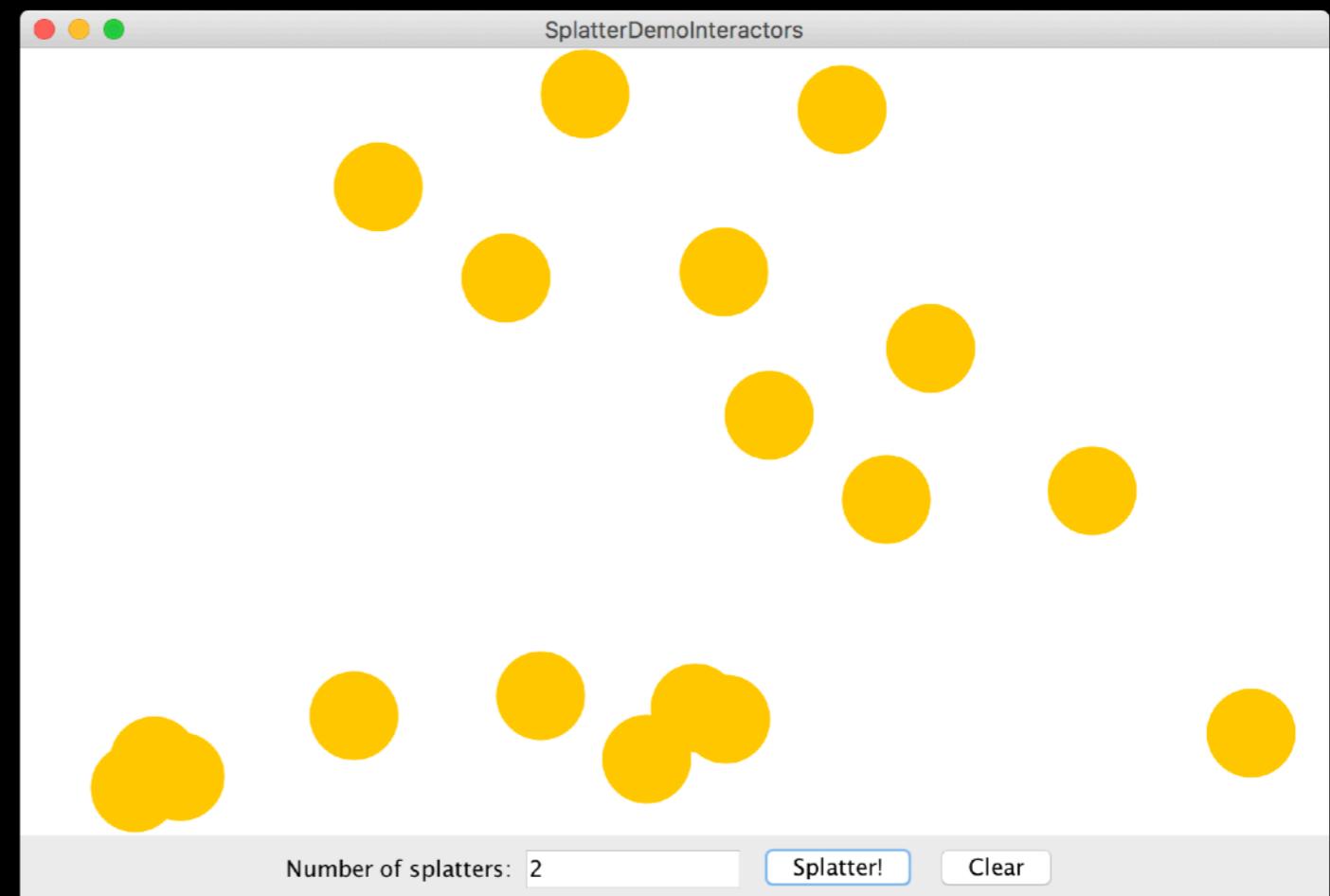
- Add interactors in init()
- addActionListeners() to listen for button presses
- addActionListener(this) on any text field to detect ENTER
- implement actionPerformed method
- Java will call actionPerformed when any action occurs.

Interactors

```
public void actionPerformed(ActionEvent e) {  
    if (e.getActionCommand().equals("My Button")) {  
        ...  
    }  
}
```

Interactors example: splatter

- Write a program that uses the **splatter** method from before
- Add a label and a text field
- Splatter! button calls **splatter** method with the number in the text field
- Clear button removes everything from canvas



Graphics example: splatter

```
public class SplatterDemoInteractors extends GraphicsProgram {  
  
    private RandomGenerator rgen = new RandomGenerator();  
    private JTextField numField;  
  
    public void init() {  
        add(new JLabel("Number of splatters:"), SOUTH);  
        numField = new JTextField(10);  
        numField.setText("20");  
        add(numField, SOUTH);  
        add(new JButton("Splatter!"), SOUTH);  
        add(new JButton("Clear"), SOUTH);  
        addActionListeners();  
    }  
  
    public void actionPerformed(ActionEvent e) {  
        if (e.getActionCommand().equals("Splatter!")) {  
            splatter(Integer.parseInt(numField.getText()));  
        } else if (e.getActionCommand().equals("Clear")) {  
            removeAll();  
        }  
    }  
  
    private void splatter(int n) {  
        ...  
    }  
}
```

we could also do this with
an instance variable for
the number of splatters
and an action listener on
the text field!

Overview

- Graphical Programs
 - Graphics
 - Interactors
- Data Storage
 - Strings
 - HashMaps
 - Classes
 - Internet

Data Storage

	String	Array	2D Array	ArrayList	HashMap
Model	Sequence of letters or symbols	Fixed length elements in a list	Grid / Matrix of elements	Growable list of elements	Key/Value mapping
Type of element	chars	Objects & Primitives	Objects & Primitives	Objects	Object/Object
Access Elements	<code>str.charAt(i);</code>	<code>arr[i];</code>	<code>arr[r][c];</code>	<code>list.get(i);</code> <code>list.set(i, elem)</code> <code>list.add(elem)</code>	<code>map.put(key, value)</code> <code>map.get(key);</code>
Special notes	Immutable	Watch bounds!	Row, col structure	Just fantastic	Each key must be unique. Unordered
Examples	"Hello world"	Histogram	ImageShop pixels	Hangman words, entries in namesurfer	NSDatabase, FPDatabase

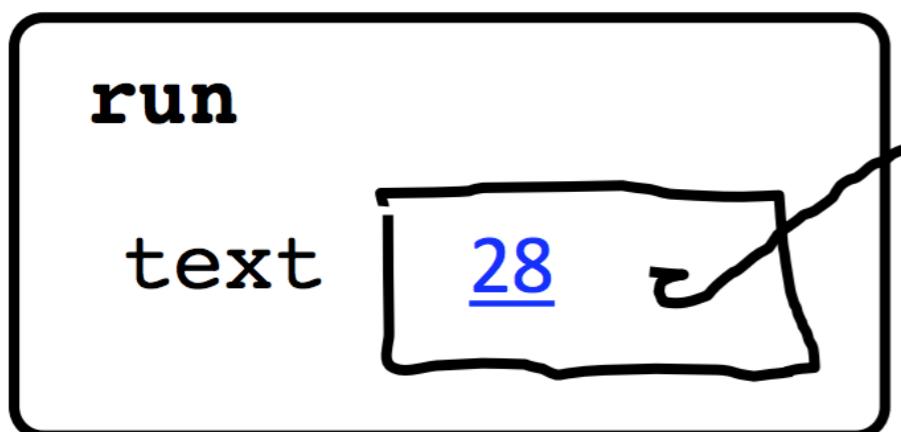
Strings

	String	Array	2D Array	ArrayList	HashMap
Model	Sequence of letters or symbols	Fixed length elements in a list	Grid / Matrix of elements	Growable list of elements	Key/Value mapping
Type of element	chars	Objects & Primitives	Objects & Primitives	Objects	Object/Object
Access Elements	<code>str.charAt(i);</code>	<code>arr[i];</code>	<code>arr[r][c];</code>	<code>list.get(i);</code> <code>list.set(i, elem)</code> <code>list.add(elem)</code>	<code>map.put(key, value)</code> <code>map.get(key);</code>
Special notes	Immutable	Watch bounds!	Row, col structure	Just fantastic	Each key must be unique. Unordered
Examples	"Hello world"	Histogram	ImageShop pixels	Hangman words, entries in namesurfer	NSDatabase, FPDatabase

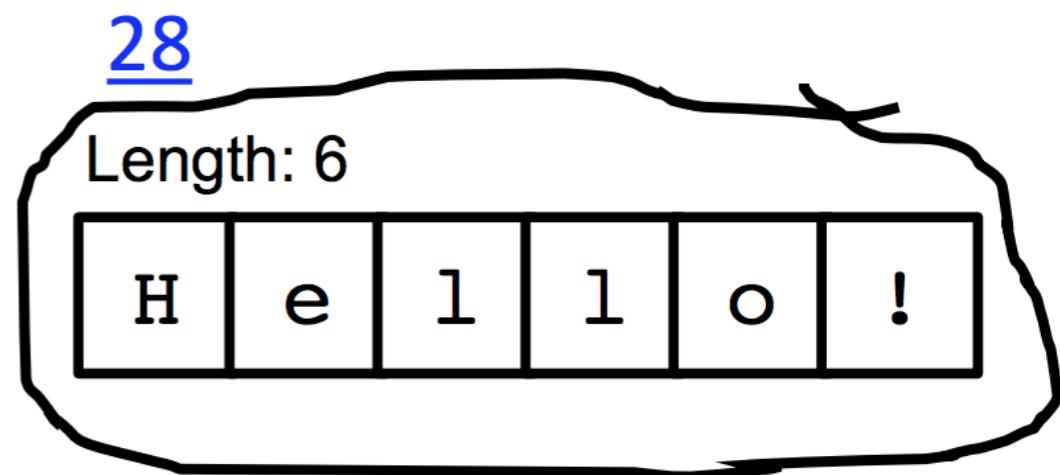
Strings

```
public void run() {  
    String text = "hello!";  
}
```

stack



heap



HashMaps

	String	Array	2D Array	ArrayList	HashMap
Model	Sequence of letters or symbols	Fixed length elements in a list	Grid / Matrix of elements	Growable list of elements	Key/Value mapping
Type of element	chars	Objects & Primitives	Objects & Primitives	Objects	Object/Object
Access Elements	<code>str.charAt(i);</code>	<code>arr[i];</code>	<code>arr[r][c];</code>	<code>list.get(i);</code> <code>list.set(i, elem)</code> <code>list.add(elem)</code>	<code>map.put(key, value)</code> <code>map.get(key);</code>
Special notes	Immutable	Watch bounds!	Row, col structure	Just fantastic	Each key must be unique. Unordered
Examples	"Hello world"	Histogram	ImageShop pixels	Hangman words, entries in namesurfer	NSDatabase, FPDatabase

HashMaps

HashMap<String, String> pets

“Chris” “Julia” “Brahm” “Annie”



“dog”

“cat”

“parrot”

“dog”

Classes

Classes

- Define a new variable type
- Need to know 3 things:
 1. what type of information it contains (**instance variables**)
 2. what it can do (**methods**)
 3. how to make a new one (**constructor**)

Classes example: Airplane

Let's write a class **Airplane** that implements the following methods. **Airplane** is initialized with its capacity.

```
/** Boards a single passenger, at front or back */
public void boardPassenger(String name, boolean priority);
/** Returns whether the vehicle is full */
public boolean isFull();
/** Unboards/returns next passenger */
public String unboardPassenger();
```

Classes example: Airplane

example usage of Airplane class

```
Airplane plane = new Airplane(100);
while (!plane.isFull()) {
    String passengerName = readLine("Name: ");
    boolean priority = readBoolean("Priority? ");
    plane.boardPassenger(passengerName, priority);
}

// fly...
String firstPassengerOff = plane.unboardPassenger();
println(firstPassengerOff);
```

Classes example: Airplane

- Need an ArrayList of passenger names
- Need an int to store the maximum capacity

Classes example: Airplane

```
public class Airplane {  
    private ArrayList<String> passengers;  
    private int capacity;  
  
    public Airplane(int numSeats) {  
        passengers = new ArrayList<String>();  
        capacity = numSeats;  
    }  
    ...
```

Classes example: Airplane

```
public void boardPassenger(String name, boolean priority) {  
    if (priority && !isFull()) {  
        passengers.add(0, name);  
    } else if (!isFull()) {  
        passengers.add(name);  
    }  
}  
...
```

Classes example: Airplane

```
public boolean isFull() {  
    return capacity == passengers.size();  
}  
  
public String unboardPassenger() {  
    if (passengers.size() > 0) {  
        return passengers.remove(0);  
    } else {  
        return "";  
    }  
}
```

Classes example: Airplane



Internet

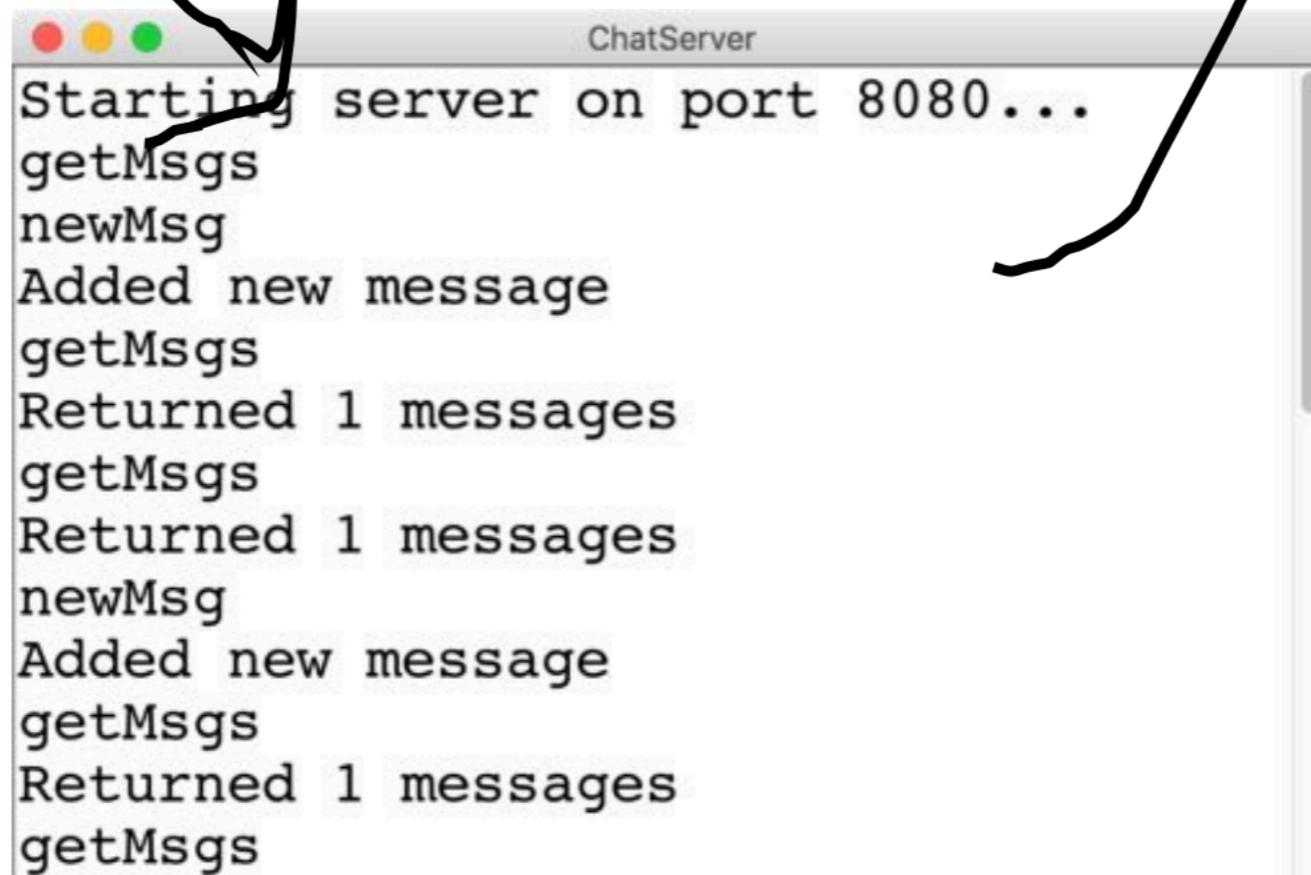
Internet

- Requests sent from client to server
- Responses sent from server to client
- Both are communicated as Strings
- Client interprets String as appropriate (even into images!)

Internet

Request
someRequest

String
serverResponse



ChatServer
Starting server on port 8080...
getMsgs
newMsg
Added new message
getMsgs
Returned 1 messages
getMsgs
Returned 1 messages
newMsg
Added new message
getMsgs
Returned 1 messages
getMsgs

Internet: Server logic

1

```
public String requestMade(Request request) {  
    // server code goes here  
}
```

2

```
// make a Server object  
private SimpleServer server  
= new SimpleServer(this, 8000);
```

3

```
public void run(){  
    // start the server  
    server.start();  
}
```

Event response on one slide

- 3 different event response types we've learned
- Mouse events in graphics programs: add mouse listeners, write 1 void method for each type of mouse action (mouseClicked, mouseMoved...)
- Interactor action events: add action listeners, 1 void method ActionPerformed with logic inside to decide which type of action it was.
- Server request received: start server, 1 method requestMade with logic inside to decide which type of request it was
must return a String!

Words of advice

- Try to get to every problem
- Don't rush to code too quickly - make sure you understand problem statement
- Pseudocoding & decomposition
- Look over practice finals, section handouts, and book for more practice
- Be proud of how much you've learned in 10 weeks!