

# CMPE 202

---

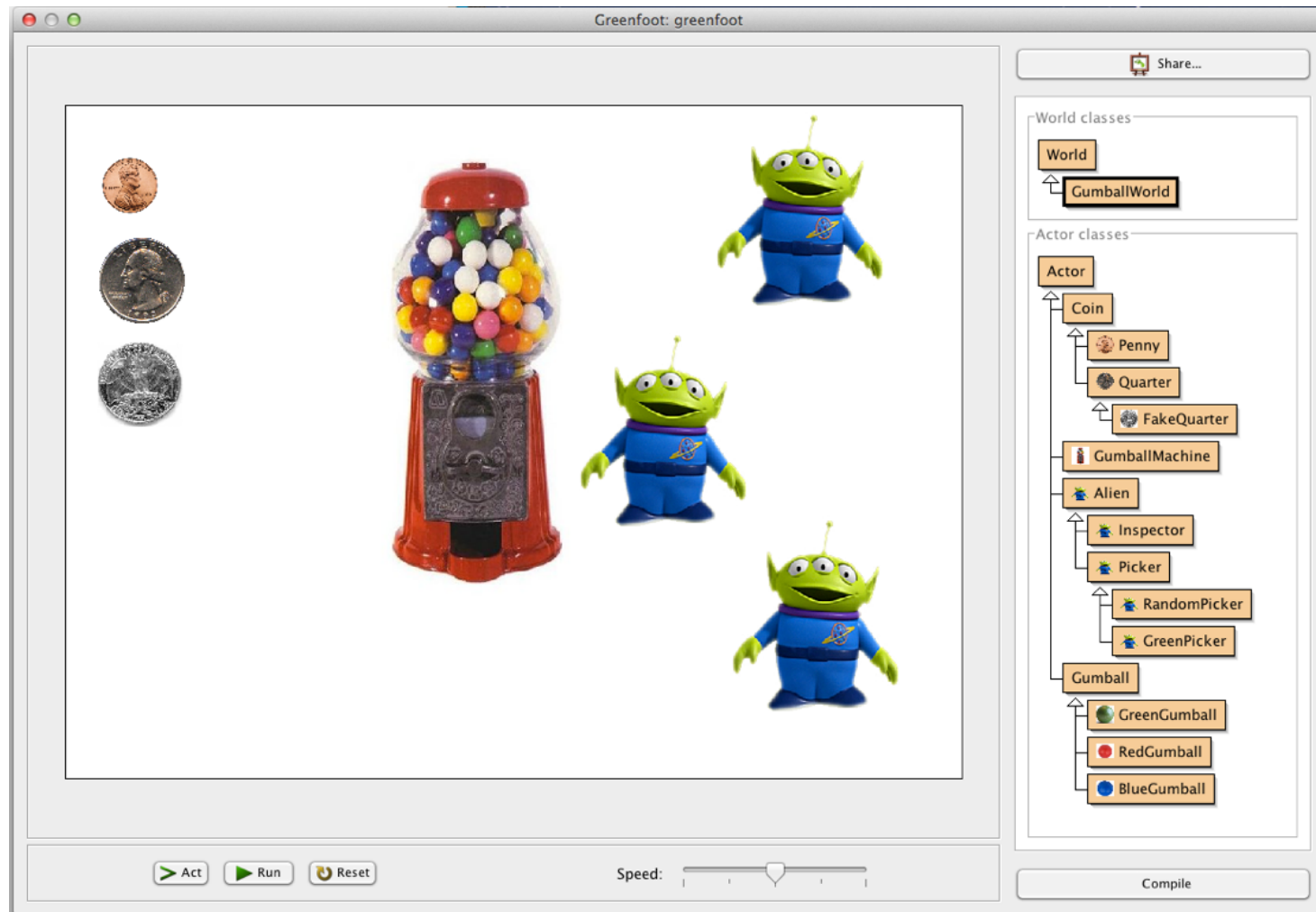
Gumball Machines Pre-Req Lab

# The Gumball Machine

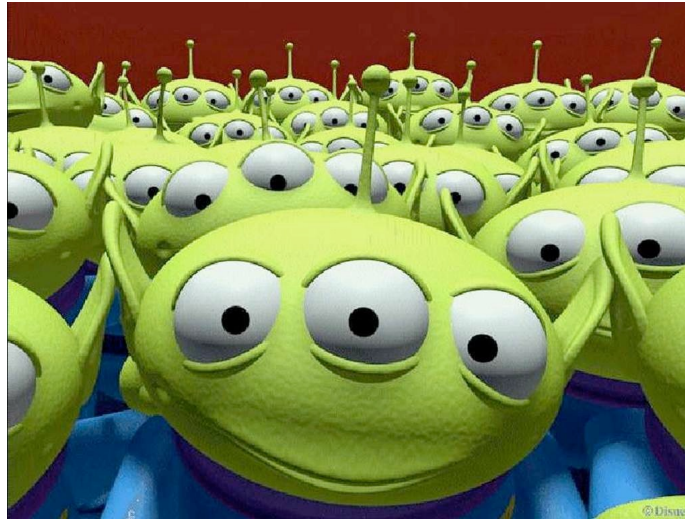
---



# Gumball Machine (Greenfoot)



# Implementation in Java (Greenfoot)



COINS



Gumball  
PICKERS



Gumball  
Machine



fake  
quarter

Coin  
Inspector

Gumballs

# Greenfoot Gumball Machine

---

- Implement the following features in Greenfoot using the starter code provided:
  - Dragging a coin over (i.e. touching) the gumball machine results in the gumball machine printing the message “have coin” to the screen. The coin should then disappear from the screen.
  - Clicking on the gumball machine (i.e. turn crank) will get the aliens started with their work. Implement the object collaborations you (your team) designed for the aliens.
  - The turn crank process should end with a gumball appearing on the screen if the coin inserted was a “real” quarter.

# Hints

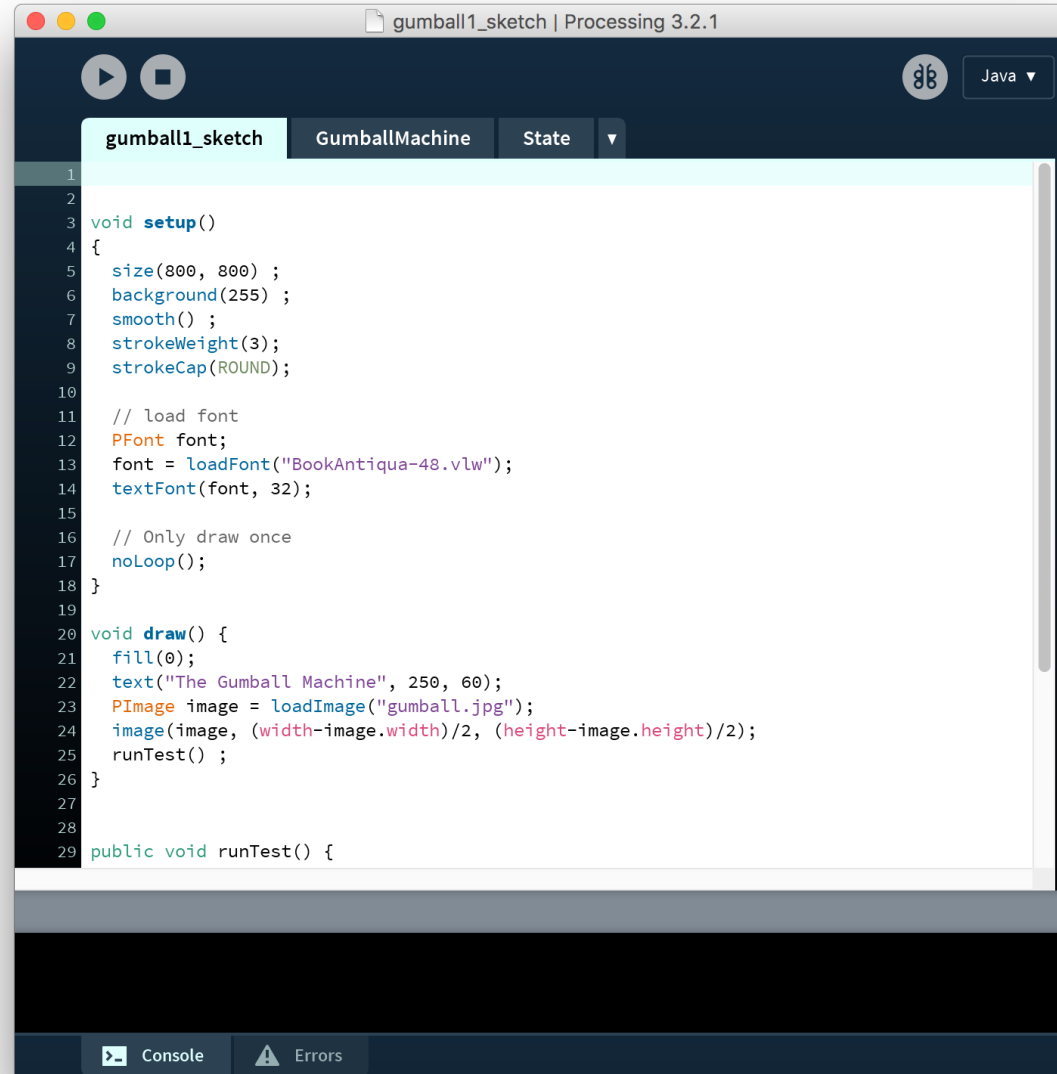
---

- Use The following online resources:
  - See Demo of Sample Solution from Instructor
  - Greenfoot Documentation: <http://www.greenfoot.org/doc>
  - Greenfoot API: <http://www.greenfoot.org/files/javadoc/>



# Gumball Machine (Processing)

---





gumball1\_sketch | Processing 3.2.1

gumball1\_sketch GumballMachine State ▾

```
1
2 public class GumballMachine {
3
4     State soldOutState;
5     State noQuarterState;
6     State hasQuarterState;
7     State soldState;
8
9     State state = soldOutState;
10    int count = 0;
11
12    public GumballMachine(int numberGumballs) {
13        soldOutState = new SoldOutState(this);
14        noQuarterState = new NoQuarterState(this);
15        hasQuarterState = new HasQuarterState(this);
16        soldState = new SoldState(this);
17
18        this.count = numberGumballs;
19        if (numberGumballs > 0) {
20            state = noQuarterState;
21        }
22    }
```

Console Errors

gumball1\_sketch | Processing 3.2.1

gumball1\_sketch GumballMachine State ▾

```
1
2 public interface State {
3
4     public void insertQuarter();
5     public void ejectQuarter();
6     public void turnCrank();
7     public void dispense();
8 }
9
10
11 public class HasQuarterState implements State {
12     GumballMachine gumballMachine;
13
14     public HasQuarterState(GumballMachine gumballMachine) {
15         this.gumballMachine = gumballMachine;
16     }
17
18     public void insertQuarter() {
19         System.out.println("You can't insert another quarter");
20     }
21
22     public void ejectQuarter() {
```

Console Errors

# The Gumball Machine



# Processing Gumball Machine

---

- Implement the following features in Processing using the starter code provided (Processing Gumball Version 1):
  - Add two buttons: “**Insert Quarter**” and “**Turn Crank**” with *white* background. When the mouse hovers over these button, change the background color to *grey*. When a user clicks inside the button, change the background color to *black* and also initiate the proper method to the Gumball Machine.
  - All Output can be to **Console** (*i.e. count of gumballs in the machine, error/success messages, etc...*)
  - **Optional:** Display the Inventory and Status of the Gumball Machine to the User (*i.e. not just debug messages to Console*)

# Hints

---

- Use The following online resources:
  - Processing Documentation: <https://processing.org/reference/>
  - Processing Tutorials: <https://processing.org/tutorials/>
  - Processing Examples: <https://processing.org/examples/>