CSSE 220 Day 10

Communities of interacting objects; UML
BallWorlds Intro
Work on BallWorlds

CSSE 220 Day 10

- Grader comments for JUnit and BigRational assignments should be in your repository.
- There seems to be a problem there with JUnit. I will check with the grader about it.

About Thursday's exam

- Closed book part.
  - You may bring one 8.5 x 11 inch paper with anything you want written on it)
  - Questions on basics of the Java language, OOP, or about specific programming assignments
  - Written homework and ANGEL quizzes are good examples of the kinds of problems I might ask
  - Weiss Sections 1.1-4.5 plus appendices on Swing and event-handling

- Programming part
  - Approximately 60% of the credit but 75% of the time for the exam..
  - Two small programs to get working on your computer
  - Almost all credit will be for correctness, none for comments, efficiency, style, or "effort"
  - Resourceds allowed: Eclipse and web browser only. You may use the 220 ANGEL pages, course web pages, and any site linked from them).

  - **Both parts**: No communication with others, no use of any device with earphones or headphones

Programming problems on exam

1. (similar in style to BigRational).
   - Define a class that meets certain specifications.
   - I will give you part of the code for that class, you must fill in the rest.
   - I will give you JUnit tests that your code must pass.
2. GUI programming using Swing.
   - The problem I give you will be variations on one of our examples (in-class or homework).

   Don't dawdle on this part.

Today's agenda

- Returning multiple values from a method
- Interacting communities of Objects
- BallWorlds Introduction

Returning Multiple Values From a Method

- In Python we could simply write
  return x, y
- In C, we could pass pointers to variables and change what they pointed to.
- What can we do in Java?
- This is a simple example of what is called the Composite Pattern.
  The returned value is a composition of two or more values that may be unrelated other than by the need to be returned from a function

BallWorlds Intro: recap

- So far, we have written "from scratch" programs.
- Most programmers do not get that luxury.
- They write a small part of a program that is designed/written by a larger team.
- Their part has to "fit" with the other parts.
- They have to understand enough of the other parts to be able to make their part work.
- In BallWorlds, you will experience that.

Goals of BallWorlds

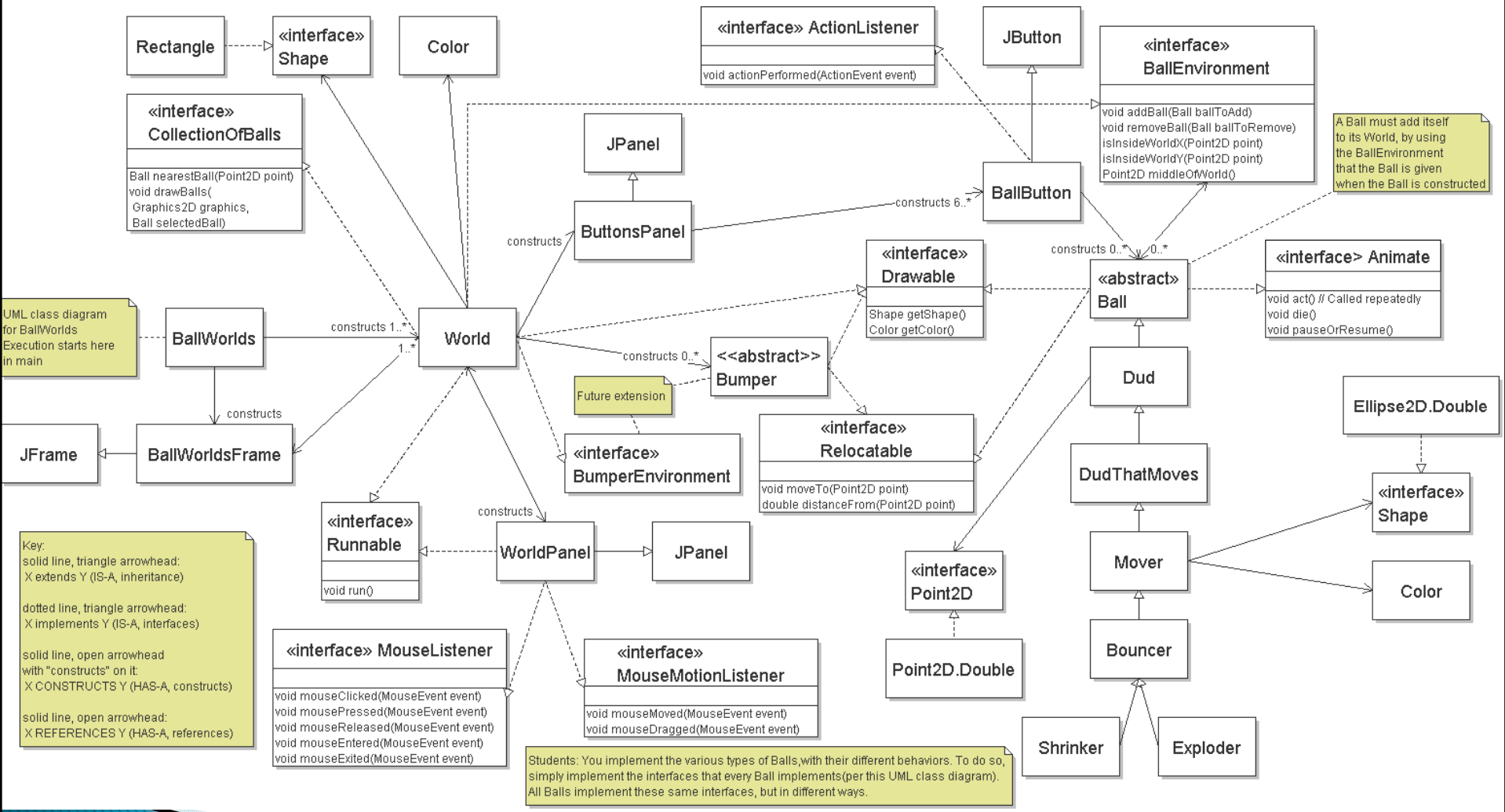
- Understand things about a program based on its UML Class Diagram
- Figure out which parts are relevant to what you have to do
- Experience the power of inheritance

- **DEMO:**
  Demonstrate the program
  How many worlds are there?

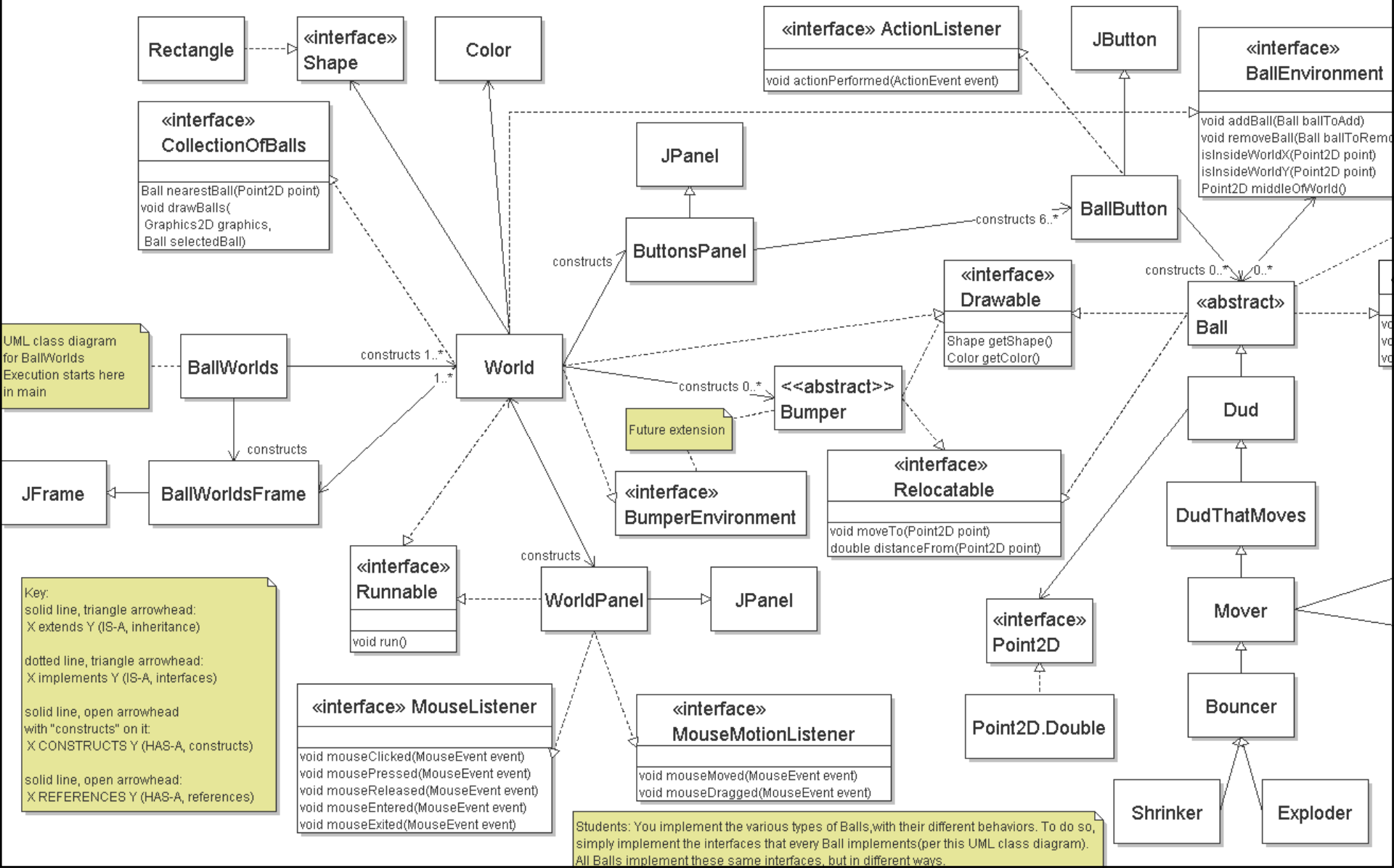
Creation of the Worlds

```
/*
 * Makes the given number of Worlds, giving each the given frame.
 * Rotates between 3 pre-assigned sizes and colors for the Worlds.
 */
private static void makeWorlds(int numberOfWorlds,
                               BallWorldsFrame frame) {
    ArrayList<Dimension> dimensions = new ArrayList<Dimension>();
    ArrayList<Color> colors = new ArrayList<Color>();

    dimensions.add(BallWorlds.world1Size);
    dimensions.add(BallWorlds.world2Size);
    dimensions.add(BallWorlds.world3Size);

    colors.add(BallWorlds.world1Color);
    colors.add(BallWorlds.world2Color);
    colors.add(BallWorlds.world3Color);

    for (int k = 0; k < numberOfWorlds; ++k) {
       new World(dimensions.get(k % 3), colors.get(k % 3), frame);
    }
}
```

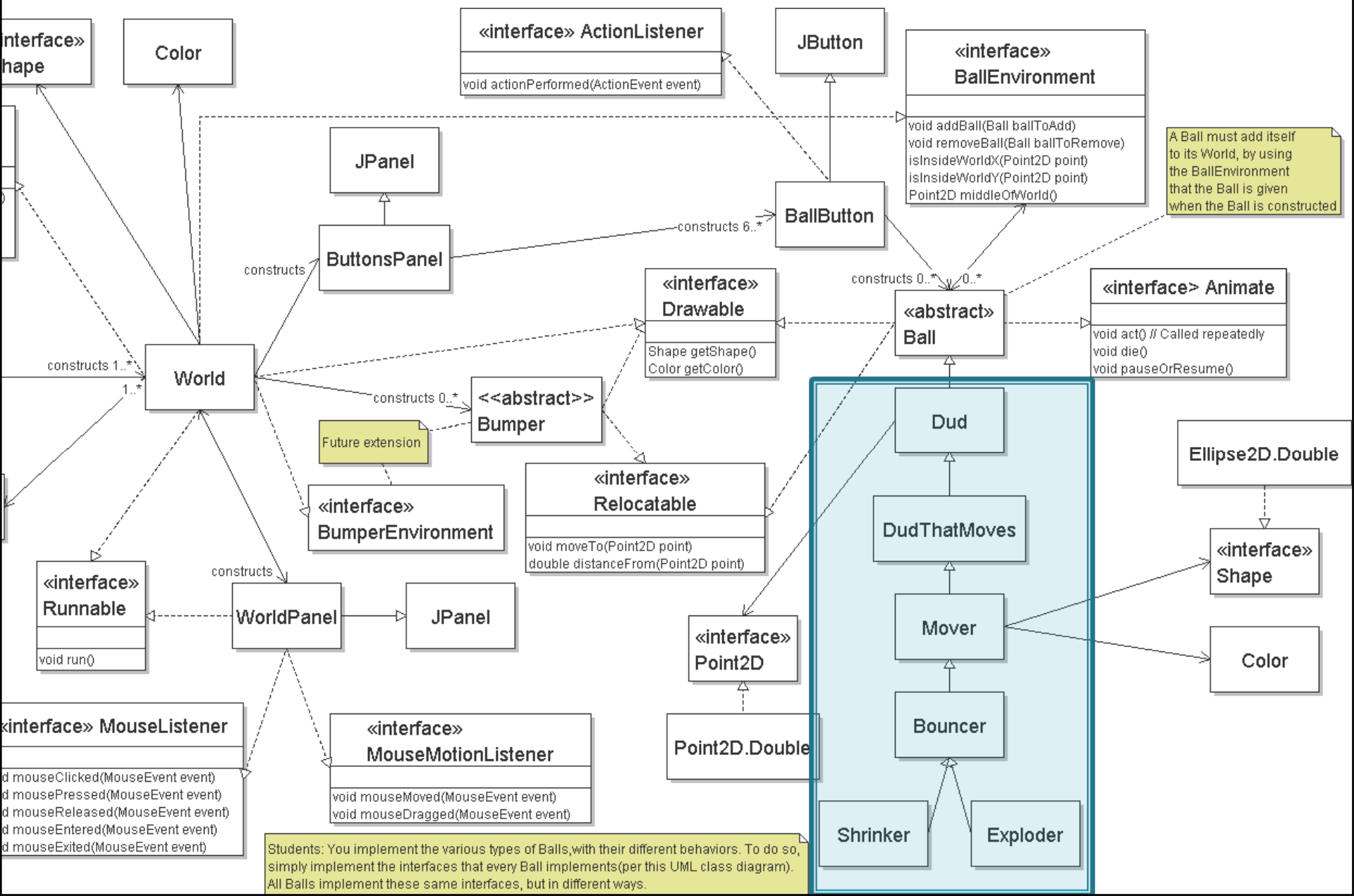
UML Class Diagram for BallWorlds



More details on Part of Diagram



Focus on the Part You Will Implement



Ball Class

- Abstract
- Implements which interfaces?
- What data might be needed for every kind of Ball?

- Let's do a little bit of code exploration.
- Then write Dud together.