CSSE 220

Collision Handling without InstanceOf

Checkout *DoubleDispatch* project from SVN

InstanceOf

- If you do inheritance correctly, you shouldn’t need instanceOf…
  - Centipede game doesn’t make this easy
  - Why?
    - Because interactions between monsters depend on the type of each object
- How do we do this?
  - Double Dispatch!

Let’s say you have this class …

```
public abstract class Monster {

        public abstract void collide(Monster m);

        public abstract void collide(Mushroom m);

        public abstract void collide(Centipede m);

        public abstract void collide(Scorpion m);
}
```

Late-Binding with Params? Uh oh…

- So this code:

  ```
  Monster m = getCollidedMonster();
  //We’ll say getCollidedMonster() returned a Mushroom
  this.collide(m);
  ```

- What method is called?
  - collide(Monster monster)
    - NOT collide(Mushroom mushroom), even though the instantiation type of m was Mushroom
- Late-binding only works for the implicit argument (what becomes **this**), it doesn’t apply to parameter types.

This would work, but ew….

```
public abstract class Monster {

   public void collide(Monster m) {
        if (m instanceof Mushroom) {
               this.collide((Mushroom)m); return;
        }
        if (m instanceof Centipede) {
               this.collide((Centipede)m);return;
        }
        if (m instanceof Scorpion) {
               this.collide((Scorpion)m); return;
        }
   }
   public abstract void collide(Mushroom m);

   public abstract void collide(Centipede m);

   public abstract void collide(Scorpion m);
}
```

Ew means
Don’t Do This!

Let’s try Double Dispatch…

```
public interface Monster {

      void collide(Monster m);

      void collideWithMushroom(Mushroom m);

      void collideWithCentipede(Centipede m);

      void collideWithScorpion(Scorpion m);
}
```

Double Dispatch

The key: You know your own type, so let’s say we’re in the Mushroom class, and Monster is of type Centipede:

```
public class Mushroom implements Monster {
        public void collide(Monster m) {
                m.collideWithMushroom(this);
        }

        public void collideWithMushroom(Mushroom m) {
                //do specific action
        }
}
```

This will call the collideWithMushroom method on the Centipede class.

Then in the Centipede’s collideWithMushroom class, add code for what should happen when a Centipede collides with a Mushroom.

Work time

Be sure everyone is getting a chance to drive.

TEAM PROJECT