CSSE 374: Getting a Grasp on GRASP

Shawn Bohner
Office: Moench Room F212
Phone: (812) 877-8685
Email: bohner@rose-hulman.edu

Q1

+/∂ Feedback: Lectures

Pace

- 0 – much too fast
- 9 – somewhat too fast
- 7 – Somewhat too slow
- 0 – much too slow

Working well

- ☐ Class slides and material (5)
- ☐ Good class exercises (5)
- ☐ Diagram examples (4)
- ☐ Group activities (3)
- ☐ Daily Quizzes (3)
- ☐ Homeworks (2)
- ☐ Move to better classroom (1)
- ☐ Teaching style (1)

Improvements

- Not 1st hour (3) ☺
- On Target (3)
- Can't think of anything (2)
- More exercises (2)
- Show more idea solutions (1)
- More time on complex slides, less on simple ones (1)
- Smaller activities with More Depth (1)
- Pick up the pace (1)
- Modulate time on slides (1)

+/∂ Feedback: Quizzes

Quizzes

- 6 – Very helpful
- 8 – somewhat helpful
- 2 – somewhat unhelpful
- 0 – Very unhelpful

Working well

- Focuses lecture for me (5)
- Questions work well (4)
- Good study guide (4)
- Indicates high points (2)
- Enough time to answer (2)
- Good length/depth (1)

Improvements

- Quizzes are fine (3)
- Diagram questions more time or explanation (2)
- Sometimes hard to complete in time provided (2)
- Be more specific in answers (1)
- More heads-up on questions (1)

+/∂ Feedback: Reading and Homework

Reading	Homework Difficulty
1 – all of it	0 – much too difficult
4 – most of it	12 – a bit too difficult
7 – little of it	4 – a bit too easy
4 – none of it	0 – much too easy

+/∂ Feedback: Homework Helpfulness

Homework Helpfulness

- 8 – very helpful
- 8 – somewhat helpful
- 0 – somewhat unhelpful
- 0 – very unhelpful

Working well

- ☐ Re-enforces class material (6)
- ☐ Corresponds to Milestones (5)
- ☐ Good feedback (3)
- ☐ Provide sufficient info. (2)
- ☐ Frequency about right (2)
- ☐ Good and relevant (2)

Improvements

- More specific instructions (6)
- Working well (3)
- Make due at midnight (3)
- Easier homework (1)

⇕

- Challenging homework (1)
- Less open-ended (1)
- Rubric for homework (1)
- Homework seemed long (1)
- Due same time as milestones conflicts with priorities (1)
- Team-based homework (1)

ROSE-HULMAN INSTITUTE OF TECHNOLOGY

+/∂ Feedback: Workload

- Workload
  - 1 – much higher than average
  - 10 – somewhat higher than average
  - 5 – somewhat lower than average
  - 0 – much lower than average
- General Comments
  - Just about right (3)
  - Hope the group participation is a huge part of grade (1)
  - Consider CSSE 371 milestone pattern (1)
  - Encouragement (11), Neutral (5) Discouragement (0) ☺

Summary of +/∂ Actions

- Better clarify homework assignments
- More time to answer quiz questions
- Pace class better
- Homework at 11:55pm (yes or no?)

Mastering Object-Oriented Design

- A large set of *soft* principles
- It isn't magic. We learn it with:
  - Patterns (named, explained, and applied)
  - Examples
  - Practice

"The critical design tool for software development is a **mind well-educated in design principles.**"

Responsibility-Driven Design

- Responsibility Driven Design (RDD)
  - Pioneered by Wirfs-Brock in early 1990s
- Think of objects in terms of:
  - What they do
    or
    What they know

  …he human worker metaphor!

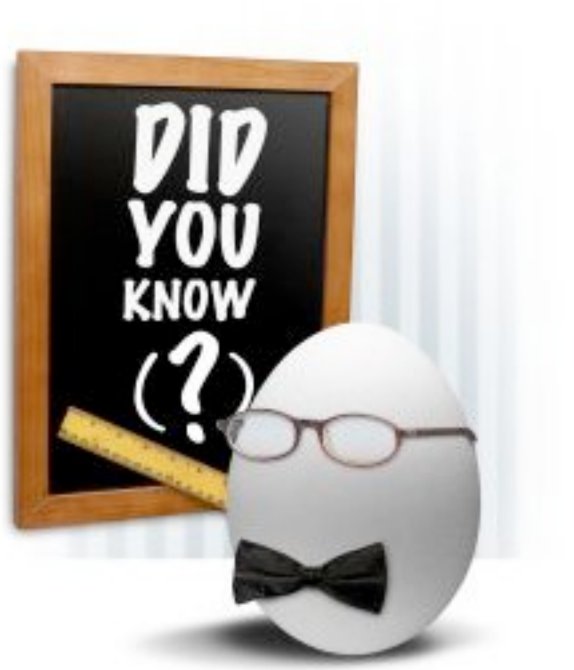
- An object's obligation or contract that it offers to other objects

Responsibilities for an Object

- Doing
  - a *Sale* is responsible for creating instances of *SalesLineItem*
- Knowing
  - a *Sale* is responsible for knowing its *total* cost

Knowing and Doing Responsibilities

- "Doing" Responsibilities
  - Create another object
  - Perform a calculation
  - Initiate an action in an object
  - Control/coordinate activities of objects
- "Knowing" Responsibilities
  - Knowing it's own encapsulated data
  - Knowing about other objects
  - Knowing things it can derive or calculate



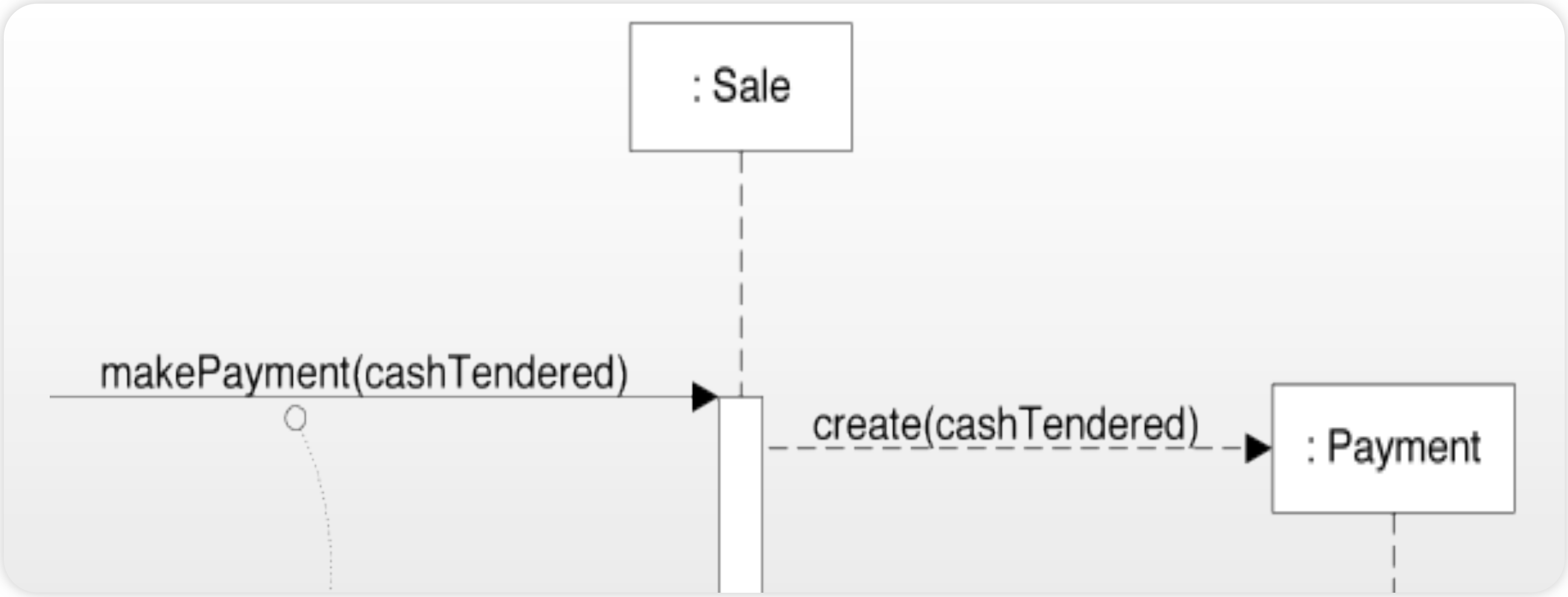
Responsibilities Come in All Sizes

- **BIG**: **provide access to a relational database**
- small: **create a Sale**

A **responsibility** is **not** the same thing as a **method**

When Do We Assign Responsibilities?

- While coding
- While modeling
  - UML is a low-cost modeling tool
  - Can assign responsibilities with minimal investment



General Responsibility Assignment Software Patterns (GRASP) 1/2

- General Responsibility Assignment Software Patterns (or Principles)
  - A set of patterns for assigning responsibilities to software objects
- What is a Pattern?
  - A pattern is a named and well-known problem-solution pair that can be applied in a new context

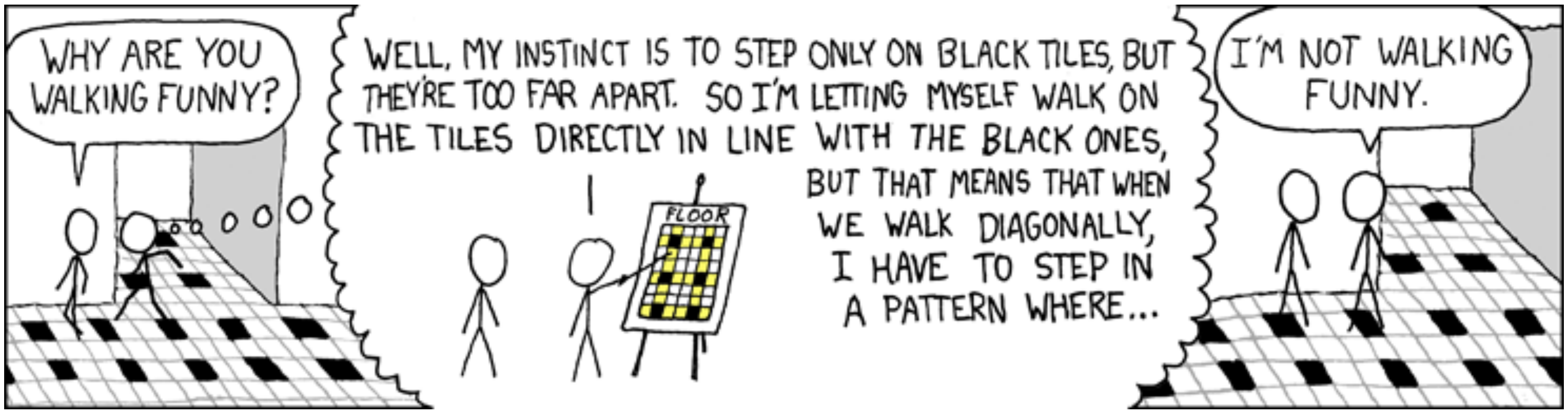
General Responsibility Assignment Software Patterns (GRASP) 2/2

- Five Covered In Chapter 17
  1. Creator
  2. Information Expert
  3. Controller
  4. Low Coupling
  5. High Cohesion

- Four Later In Chapter 25

Polymorphism	Pure Fabrication
Indirection	Protected Variations

Design: Floor Tiles


WHY ARE YOU WALKING FUNNY?
WELL, MY INSTINCT IS TO STEP ONLY ON BLACK TILES, BUT THEY'RE TOO FAR APART. SO I'M LETTING MYSELF WALK ON THE TILES DIRECTLY IN LINE WITH THE BLACK ONES, BUT THAT MEANS THAT WHEN WE WALK DIAGONALLY, I HAVE TO STEP IN A PATTERN WHERE...
FLOOR
I'M NOT WALKING FUNNY.

The worst part is when sidewalk cracks are out-of-sync with your natural stride.

Information Expert Pattern

Names Matter!

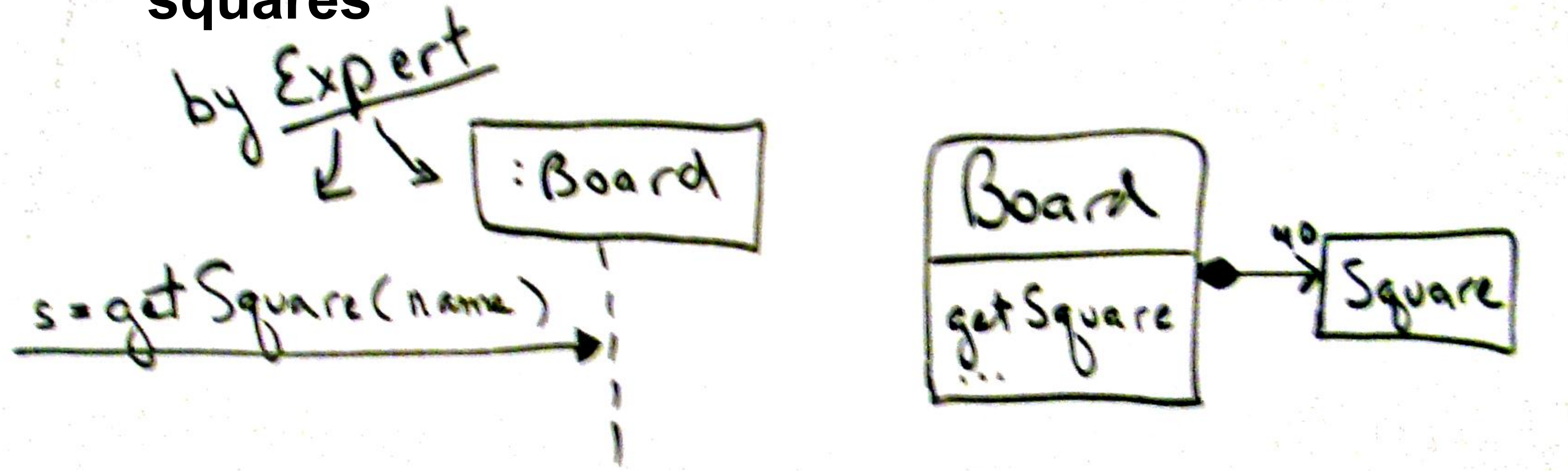
Pattern Name	***Information Expert***
Problem	What is a basic principle by which to assign responsibilities to objects?
Solution	Assign a responsibility to the class that has the information needed to fulfill it.

"New pattern" is an oxymoron!

Information Expert and Unique IDs

- Basic principle of RDD: Assign responsibility to the object that has the required information
  - "Tell the expert to do it!"

- Who should get a square given a unique ID?
  - Let the Board do it because it knows about the squares



Creator Pattern

- Who should create object A?

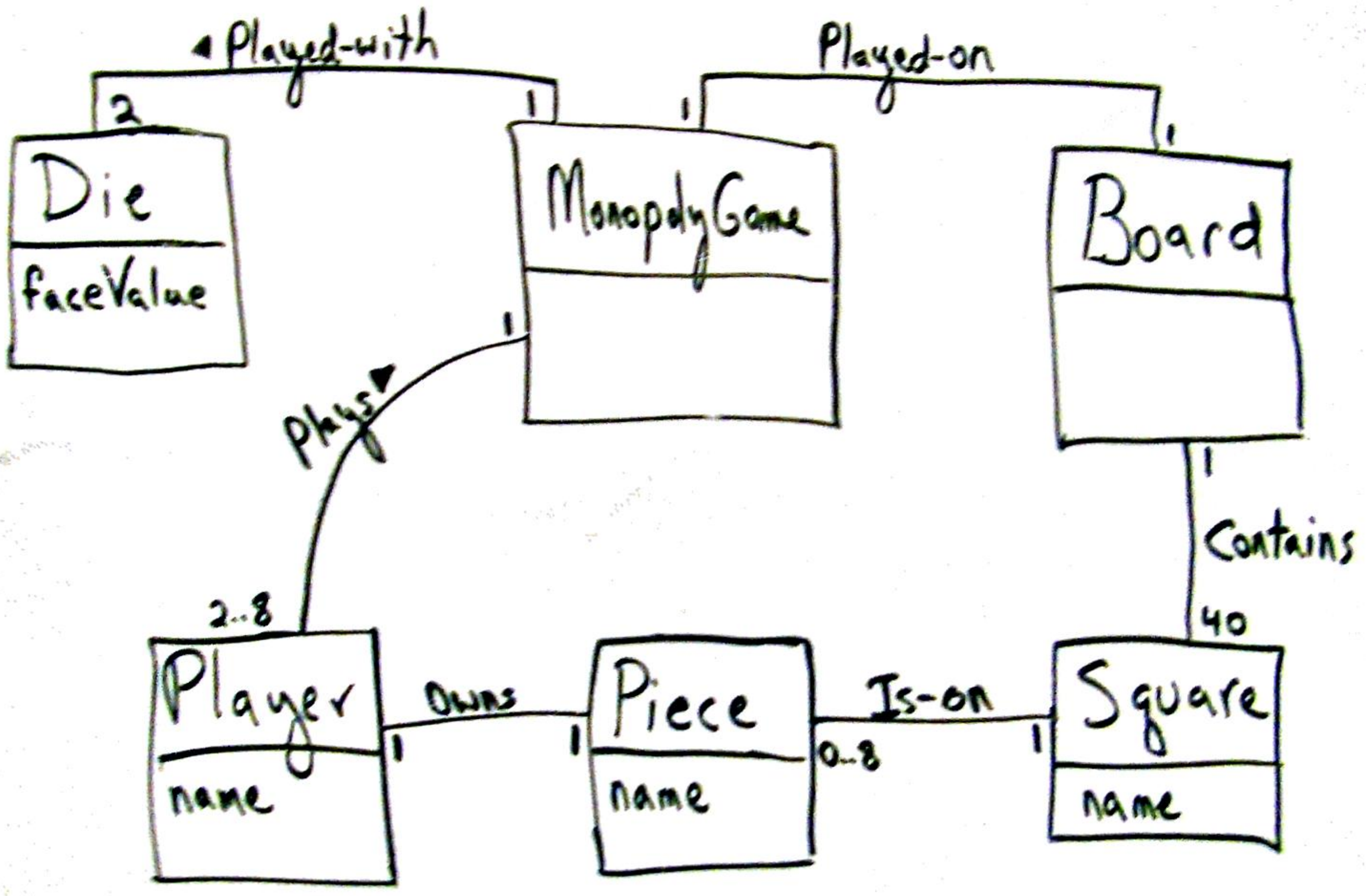
  Solution (advice):

  Let B do it if:

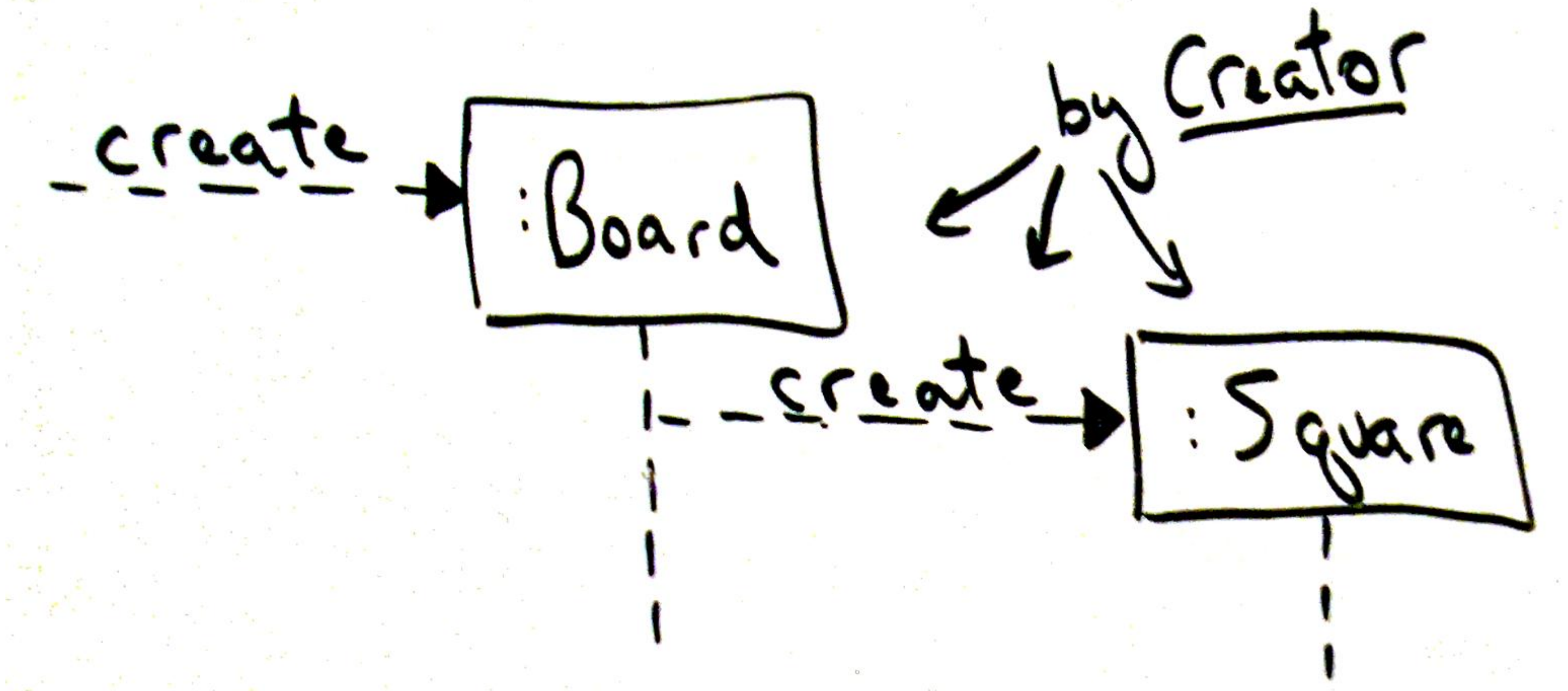
  - B contains or aggregates A
  - B records A
  - B closely uses A
  - B has the initializing data for A

- Monopoly Board Example
  - When you start a game, who creates the squares for the board?
  - Let Board create them since it *contains* the squares

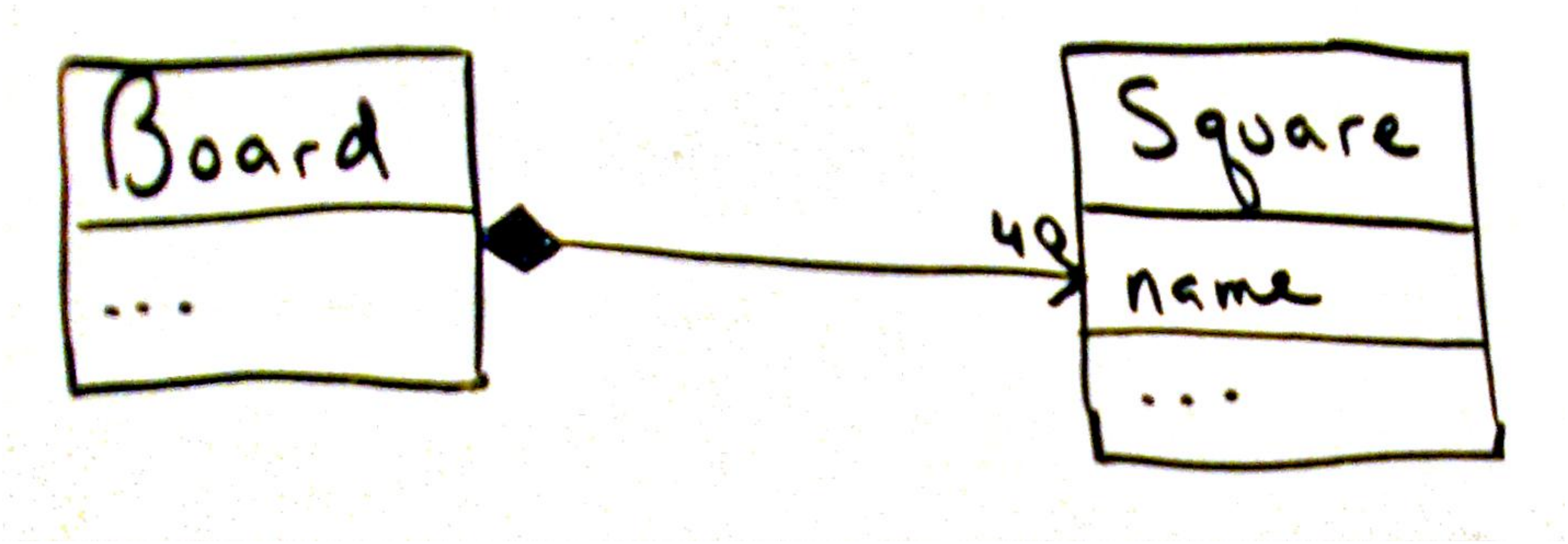
Monopoly Example



Create “in Action”

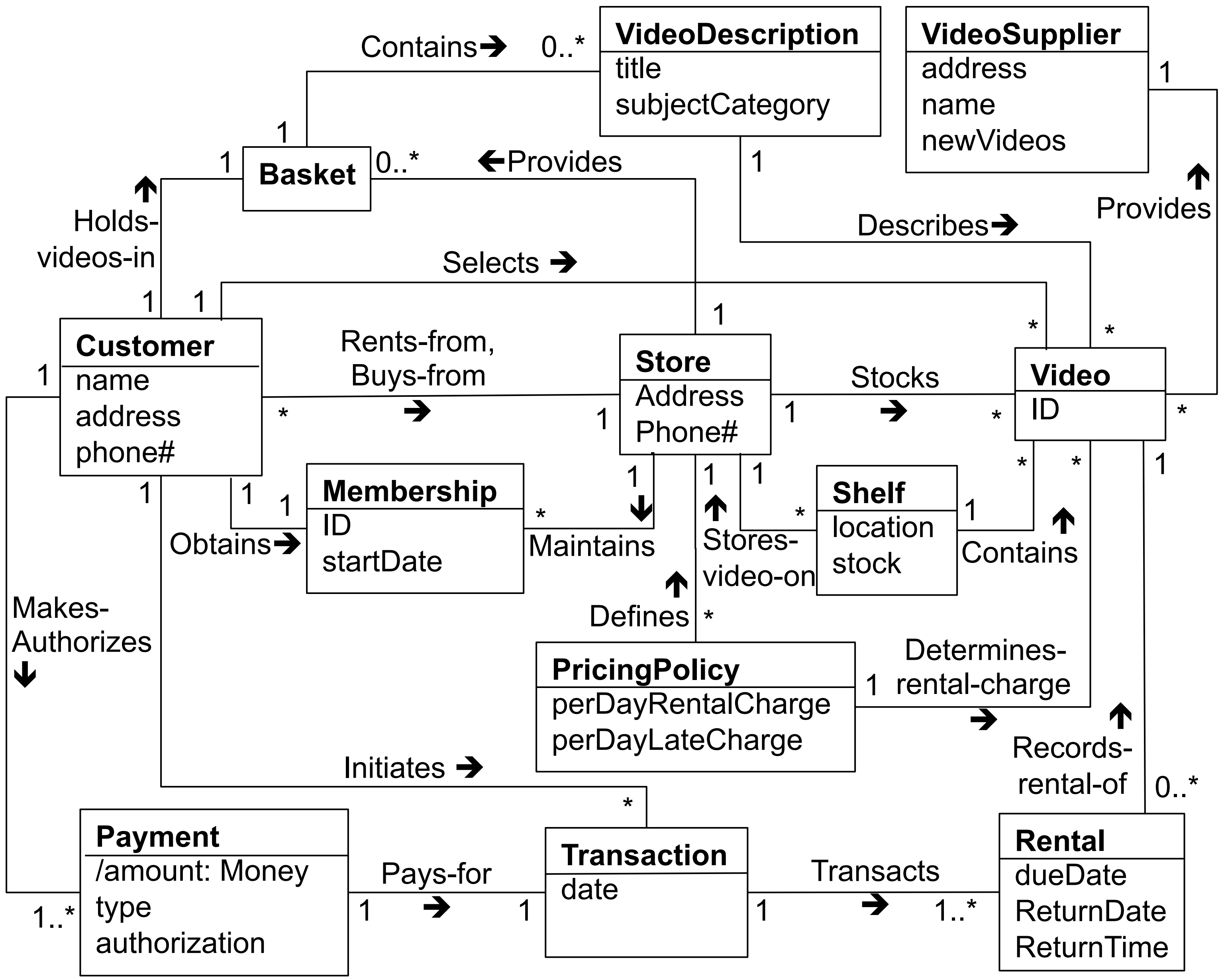


Composition



Exercise on Creator Examples

- Break up into your project teams
- Given the following:
  - Domain Model for BBVS
- Identify Creator pattern examples (hint)
  - B contains or aggregates A
  - B records A
  - B closely uses A
  - B has the initializing data for A



VideoDescription
title
subjectCategory
VideoSupplier
address
name
newVideos
Contains ➔
0..*
1
Basket
1
0..*
➔ Provides
1
1
Provides ➔
Describes ➔
Holds-videos-in ➔
Selects ➔
1
1
1
*
*
Customer
name
address
phone#
Rents-from, Buys-from ➔
*
1
Store
Address
Phone#
Stocks ➔
1
*
Video
ID
*
1
1
1
Membership
ID
startDate
*
1
Obtains ➔
Maintains ➔
1
1
1
Stores-video-on ➔
*
Shelf
location
stock
1
Contains ➔
*
*
1
Defines ➔
*
Makes-Authorizes ➔
PricingPolicy
perDayRentalCharge
perDayLateCharge
Determines-rental-charge ➔
1
Records-rental-of ➔
0..*
Initiates ➔
*
Payment
/amount: Money
type
authorization
1..*
Pays-for ➔
1
1
Transaction
date
Transacts ➔
1
1..*
Rental
dueDate
ReturnDate
ReturnTime

Homework and Milestone Reminders

- Finish Reading Chapter 17 on GRASP
- Homework 3 – BBVS Logical Architecture and Preliminary Design
  - Due by 5:00pm on Tuesday, January 4th, 2011
- Milestone 3 – Junior Project SSDs, OCs, and Logical Architecture
  - Due by 11:59pm on Friday, January 7th, 2011