Objectives

• Analysis and Design
• Final Project

Review

• What is a design pattern?
• What design patterns did we discuss?
  ➢ What design principle(s) does it follow?
• What design pattern is used in the screen savers code?
• What is the dependency inversion principle?

Analysis Phase

• Create an abstract model in client’s vocabulary

Strategy:

1. Identify classes that model (shape) system as set of abstractions
2. Determine each class’s purpose or main responsibility
   • member functions
   • data members
3. Determine helper classes for each
   • Help complete responsibilities

Analysis Phase Discussion

• Expect to iterate
  ➢ Won’t find all classes at first
    • Especially helpers
  ➢ Won’t know all responsibilities
• Uncertainty in problem statement
  ➢ May be concerns that need to be settled
  ➢ Try to understand requested software system at level of those requesting software
• Rarely one true correct best design

Identification of Classes

• Potentially model the system
• Usually nouns from problem description or from domain knowledge
• Model real world whenever possible
  ➢ More understandable software
  ➢ Helps during maintenance when someone unfamiliar with system must update/fix code
Identifying Responsibilities

- Responsibilities convey purpose of class, its role in system
- Questions to Ask:
  - What are the other responsibilities needed to model the solution?
  - Which class should take on this particular responsibility?
  - What classes help another class fulfill its responsibility?

Have You Modeled Everything?

- Strategy: Role playing
- Act as different classes: can you do everything you want in various scenarios?
  - Fill in missing classes, responsibilities
  - Methods: parameters, what returned
  - Restructure as necessary
  - No code yet so not actually refactoring
- Example use cases/scenarios:
  - User borrows a video and returns it two days late
  - User tries to borrow book that is already checked out

Definition of Use Case?

- Description of steps or actions between a user and a software system towards some goal

Discussion

- What else can use cases be used for?
  - Test Cases

TEAM FINAL PROJECT
Project Deliverables Timeline

<table>
<thead>
<tr>
<th>Deliverable</th>
<th>Who</th>
<th>Weight</th>
<th>Due Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Team Name</td>
<td>Team</td>
<td>0%</td>
<td>Thur, 11/19</td>
</tr>
<tr>
<td>Preparation Analysis</td>
<td>Individual</td>
<td>8%</td>
<td>Mon, 11/30</td>
</tr>
<tr>
<td>Preliminary Implementation</td>
<td>Team</td>
<td>30%</td>
<td>Mon, 12/7</td>
</tr>
<tr>
<td>Final Implementation</td>
<td>Team</td>
<td>47%</td>
<td>You decide by latest 12/16</td>
</tr>
<tr>
<td>Analysis</td>
<td>Individual</td>
<td>15%</td>
<td>12/18</td>
</tr>
</tbody>
</table>

Teams

- Azmain, John B, Perry, Sima
- Asha, Brandon, John K, George, Alfred
- Julianne, Steve, Alexis, Mike
- Harrison, Lenny, Truth, Joe
- Abigail, Ted, Mina, Leigh
- Bryant, Connor, Rob, Chris
- Tamzid, Brian, Alec, Rachel

Teams, alphabetically by last name

Teams Work Best When They are Interdependent

- Are you allowing your team to truly be interdependent?
- Who might be you be ignoring?
- Who might be allowing themselves to feel inadequate?
- How do you show appreciation for each other and yourself?

Project Metrics

- >1700 lines of code
  - Even more by the time your team is done
- Good for gaining experience
  - Large (for a course) piece of existing code that you need to build on
- Good for job interviews
  - Know the number of lines of code (use one of the Metrics plugins)
**Picasso Specification**

- User can enter expressions
  - Interactively or from file
  - Language is defined in specification
- Many possible extensions

**Picasso Project Overview**

- Goal: Generate images from expressions
- Every pixel gets assigned a color, computed from its x and y coordinate and the given expression
- Colors are RGB values
  - Range [-1, 1]
  - Black is [-1,-1,-1]
  - Red is [1,-1,-1]
  - Yellow is [1,1,-1]

For all x:
For all y:
\[ \text{pixels}[x][y] = \text{expression.evaluate}(x, y) \]

**Examples**

For all x:
For all y:
\[ \text{pixels}[x][y] = \text{expression.evaluate}(x, y) \]

Consider x to be an RGB tuple, e.g., [x, x, x]

What is the resulting image if the expression is
- [-1, 1, -1]?
- x?
- x*y?

**Brainstorming**

- What do you need to do to complete the project?
- What do you “see” for the final project?
- What’s going to change?
- Where do you think you’ll run into problems?
- To focus your thinking, consider this use case: “The user starts the program, types ‘x/y’ in the expression window, and sees the image.”
- What are other use cases?

**Understanding Code**

- Run program
  - What does each button do?
- Start at Main.java
  - Follow calls to see where program goes
    - Breadth or depth-first search
TODO

- Assignment 9 due Friday
- Project Analysis due Monday after Thanksgiving