

Objectives

- Analysis and Design
- Introduction to final project

Nov 20, 2009

Sprenkle - CS209

1

ANALYSIS & DESIGN: FORMALIZED

Nov 20, 2009

Sprenkle - CS209

2

Design Heuristics

- Model real world whenever possible
- Avoid all-powerful (omnipotent) classes
- Distribute system intelligence among classes evenly
 - Top-level classes should share work uniformly
 - More easily understood system
 - More easily communicated design
- Minimize # of messages between class and helper
 - Reduce coupling btw class and helper

Nov 20, 2009

Sprenkle - CS209

3

Analysis Phase

"Dohickey"

- 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

Nov 20, 2009

Sprenkle - CS209

4

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



Nov 20, 2009

Sprenkle - CS209

5

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

Nov 20, 2009

Sprenkle - CS209

6

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?

Nov 20, 2009

Sprenkle - CS209

7

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

Nov 20, 2009

Sprenkle - CS209

8

Discussion

- What else can use cases be used for?

Nov 20, 2009

Sprenkle - CS209

9

Discussion

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

Nov 20, 2009

Sprenkle - CS209

10

FINAL TEAM PROJECT

Nov 20, 2009

Sprenkle - CS209

11

Project Deliverables Timeline

Deliverable	Who	Weight	Due Date
Preparation	Individual	8%	11/30
Preliminary Implementation	Team	37%	12/7
Final Implementation	Team	40%	You decide → latest 12/18
Analysis	Individual	15%	12/18

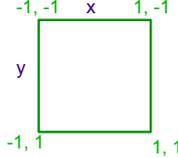
Nov 20, 2009

Sprenkle - CS209

12

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]

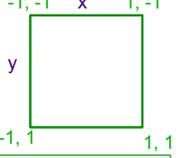


Nov 20, 2009 Sprenkle - CS209 13

Examples

```
For all x:
  For all y:
    pixels[x][y] = 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 ?

Nov 20, 2009 Sprenkle - CS209 14

Examples



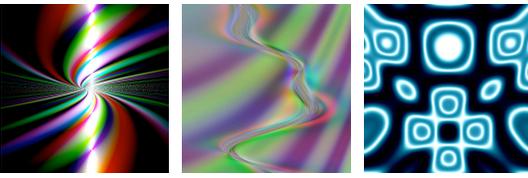
[-1, 1, -1] x x*y

```
For all x:
  For all y:
    pixels[x][y] = expression.evaluate(x, y)
```

Nov 20, 2009 Sprenkle - CS209 15

Specification

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



Nov 20, 2009 Sprenkle - CS209 16

Code Review

- Import an existing project:
 - /home/courses/cs209/handouts/picasso_init.tar
- What questions do you have about the code?
- What do you want to find out?

Nov 20, 2009 Sprenkle - CS209 17

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?

Nov 20, 2009 Sprenkle - CS209 18

Preparation Analysis

- What are the main parts/steps that need to be completed to complete the project?
 - How much work does each part require? Approximate in terms of time or relative to the other steps.
 - How many people should work on each part?
- How will your program handle the following use case: "The user starts the program, types 'x/y' in the expression window, and sees the image."?
 - From your description, it should be clear which classes/objects are responsible for completing each part of the task.
- What 3 extensions would you like to have in the final application?
- A plan for how you would tackle implementing the project.
 - What parts can be completed independently of the other parts?
 - What parts need to be completed before other parts?
- The parts of the project you're most interested in working on, in ranked order.
- Any questions about the given specification.

Nov 20, 2009

Sprenkle - CS209

19