

Project 1 Debriefing

- What are the main design concerns?
 - Breakdown into components
 - Communications among components
 - Crosscutting concerns
- Consideration of alternatives
- Informed, documented choices

Components

- A properly designed component should
 - Be cohesive
 - Be intentional
 - Have well-defined interfaces
 - Fill a well-defined role in the system
 - Prevent clients from seeing implementation
 - [Be flexible, reusable, general]
- What component in your solution best satisfies these criteria?

Components

- Grid
- Simulator
- Controller
- Model
- GUI
- Display
- [Logging]

Grid Responsibilities

- Holds a set of uninterpreted values
- Provides topological access from/to neighbors
- Computes one step
- Provides external access to computed values
- Provides formatted report
- Keeps track of its progress

Grid Issues

- Can't figure a way to design a general version that takes advantage of the efficiency of array indexing
- How to construct the various types of `Grids`
 - Constructors, reflection, factory method, generics?
- Even for `Object` version, it still has to provide an `<x, y>` access scheme for cells
- Two copies of `Grid` or two copies of each cell?

Grid Issues - 2

- Who is responsible for judging progress?
- How to compute amount of change?
 - Average over whole plate
 - Sample
 - Center cell
 - Analytic: accuracy increases (in number of bits of accuracy) as the iterations increase
- How to separate `Grid` from `Simulation`?
 - Have `run()`, `step()` and multiple `Grids` in `Simulation`
 - Have all in `Grid`
 - Split

Simulation Responsibilities

- Control simulation progress (step)
- Manage termination
- Communicate results to rest of program
 - Values
 - Steps
- Support (re-)configuration
- In place `Grid` update or separate `Grid` for each step?

Simulation **Issues**

- Should the `Simulation` be in its own thread?
- Push vs. pull communication with `Display`?
- Different reasons for quitting
 - No change
 - Step count
 - Pause
- Who is responsible for initial construction?

Controller Responsibilities

- Respond to user interface events
 - Prevents GUI from knowing about Model
- Translates from GUI events to Model updates and other effects
- Argument passing (type casting) to Model

Controller **Issues**

- How do other components access the Controller?
- Who is responsible for initial construction?

Model Responsibilities

- Isolate GUI from business logic

Model Issues

- How do other components access the Model?
 - Static versus passing in knowledge of Model to clients at the time of their constructions
- Who is responsible for initial construction?
- More complete encapsulation of domain concepts
 - How many dependencies can you name?

Sources of Domain Dependency

- Map between `Colors` and temperatures
- Labels on prompts to obtain initial conditions
- Ranges of allowed values on prompts
- Fick's law approximation method
- Map of `Grid` topology to `Display`

GUI Responsibilities

- Accept initial configuration data
- Accept user action requests
- Notify `Controller` of user events
- Present progress indicators
- Present log data and statistics

GUI Issues

- Threading
- Some display responsibilities

Display Responsibilities

- Visualize `Grid` temperatures
- Provide feedback on `Edge` settings
- Mapping of `Grid` values to `Colors`
- Depiction of `Grid` topology

Display Issues

- Access to actual `Grid` values
 - One at a time or aggregate?
- Separation from `GUI` even though both use `Swing` widgets
 - Initial condition feedback and logging data in `GUI` instead of `Display`
- Threading

Logging

- Multiple types of model-independent statistics
 - Time taken for a simulation
 - Memory used
 - Iteration count
- Issue: How to add this into `Simulation` in a cross-cutting way?
 - Interface
 - (Mixin)

Architecture Issues

- Architecture is not only classes
 - Functional breakdown for purposes of reducing complexity
 - The policy for dealing with the primary non-functional constraints
 - The collaboration policy among the components
- Push vs. pull between `Display` and `Simulation`
 - What happens if they have grossly different execution speeds?
- What is an event?
 - Completion of an iteration
 - Each cell update
- Does the event transmit the data, or is a separate request made?

Another Issue

- Orthogonality of primitive wrapping and topology
 - Multiple inheritance hierarchies for the eight possibilities
 - Policies