Design Practice & Models
Today

• Decide on code standards

• Revisit/refine user stories
• Design quick & dirty UI
• Design Sudoku architecture
• Plan next iteration
User stories

- Feature: Display initial grid
  - User story: As a player I want to view the initial Sudoku grid in order to figure out my first move. The initial grid should have blanks and should have a unique solution.

- Feature: Enter number in blank cell
  - User story: As a player I want to enter a number in a blank cell in order to solve the puzzle.
New user story

- Feature: Generate grids
- User story: As a player I want different grids of varying difficulty in order to be entertained and challenged.
Feature: Enter number in blank cell
User story: As a player I want to enter a number in a blank cell in order to solve the puzzle.

Can I enter a 2 here?
Can I enter a 5 here? (What should it be.)
Inconsistent input
Incorrect input
Feature: Enter number in blank cell
User story: As a player I want to enter a number in a blank cell in order to solve the puzzle.

Can I change values?
- Can I change an initial value?
- Can I change a value I entered?
- Can I delete a value I entered?

Can I enter a 2 here?
Can I enter a 5 here? (It should be a 3.)
Can I change this to a 5?
Changing values
Inconsistent input

69/10/13
• Feature: Enter number in blank cell
• User story: As a player I want to enter a number in a blank cell in order to solve the puzzle.

Can I change values?
- Can I change an initial value?
- Can I change a value I entered?
- Can I delete a value I entered?

How will the I distinguish initial values from ones I entered??

Can I change this to a 5?

Changing values

Inconsistent input
Can I enter a 2 here?

Incorrect input
Can I enter a 5 here? (It should be a 3.)

9/10/13
Agile

• Choose simple rules to start
• Build flexibility into design provided it doesn’t cost too much

Feature: Enter number in cell
User story: As a player I want to enter a number in a cell in order to solve the puzzle.

Scenario 1: Cell is blank and new number is consistent.
Number appears in cell in a way distinguishable from initial grid values.

Scenario 2: Cell has number I previously entered and new number is consistent.
Number appears in cell in a way distinguishable from initial grid values.

In practice you’d do some initial software design to understand the implications of this kind of decision.
Note: we can augment user stories with scenarios

As a ______________________
I want to do this ________________
In order to achieve this goal ________________

Scenario 1: ...
Scenario 2: ...
Let’s generalize our user story to display the grid throughout the game

- **Feature:** Display initial grid
  - **User story:** As a player I want to view the initial Sudoku grid in order to figure out my first move. The initial grid should have blanks and should have a unique solution.

- **Feature:** Display grid
  - **User story:** As a player I want to view the Sudoku grid in order to figure out my next move. The grid should show but distinguish the initial values and my current entries. The initial grid should have blanks and a unique solution. The grid should always be consistent.
Current stories

• Feature: Display grid
  • User story: As a player I want to view the Sudoku grid in order to figure out my next move. The grid should show the initial values and my current entries. The initial grid should have blanks and a unique solution. The grid should always be consistent.

• Feature: Generate grids
  • User story: As a player I want different grids of varying difficulty in order to be entertained and challenged.

• Feature: Enter number in cell
  • User story: As a player I want to enter a number in a cell in order to solve the puzzle.

  • Scenario 1: Cell is blank and new number is consistent.
    Number appears in cell in a way distinguishable from initial grid values.

  • Scenario 2: Cell has number previously entered by player and new number is consistent.
    Number appears in cell in a way distinguishable from initial grid values.

Today

• Decide on code standards
• Revisit/refine user stories
• Design quick & dirty UI
• Design Sudoku architecture
• Plan next iteration
UI Design

• UI Design
  – User/Task models
  – Prototype (usually paper at first)
  – Usability test
  – Refine design & repeat

• “Growing” a UI Design is difficult

• We’ll do a quick & dirty design for our next iteration
Initial grid and number pad appear. Original entries are black. Current number “1” is highlighted in number pad.
User clicks on number pad cell.

Current number de-highlighted. New number is highlighted.
User clicks on cell. Cell is blank or has a value entered by user. Current number is consistent for that cell.

Current number appears in that cell in blue.
Today

• Decide on code standards
• Revisit/refine user stories
• Design quick & dirty UI
• Design Sudoku architecture
• Plan next iteration
Model-View-Controller (MVC) Design Pattern

Model and View do not interact directly

- **Model**
  - Data
  - Logic
- **Controller**
  - Logic
- **View**
  - UI elements
Model-View-Controller (MVC)

Data (pretty dumb)
- What program does
- Model
- No UI

UI elements (pretty dumb)
- Reusable elements
- View
- Puts Controller’s View onto screen

Logic
- In Charge

Controller

Keep it simple
Model-View-Controller (MVC)
Classic example
Data

Graph controller

Interactive Graph view

User can change data by manipulating graph

9/10/13
Adding new views is easy! (Don’t need to change existing code.)
Changing underlying data models does not affect UI.

Data

- Table controller
  - Table view

- Chart controller
  - Chart view

- Graph controller
  - Graph view

9/10/13
MVC Communication

- Controller Communicates directly to View and to Model
- Model and View DO NOT Communicate
- View communicates to Controller in CONTROLLED Ways
  - Target/Action
  - Delegation
Target Action


• Like Faults/Interrupts from CS 105 - Async action that has a specified function/method that responds to the “interrupt”
Delegation


• One object operates on behalf of another
• Delegating object has a reference to the delegate
Sudoku

GridModel

Know initial value, current value, solved value, answer questions about consistency, correctness

GridView

Displays current grid, has clickable cells

ViewController

Mediate between the Views and the GridModel

NumPadView

Displays current numPad, has clickable cells
Unified Modeling Language

UML is a standard language for visualizing, specifying, constructing, and documenting the artifacts of a software system.
SAMPLE UML Class Diagram

9/10/13
Agile strategy

Simple design models — Working software
Design Goals

1. Make it easy to build
2. Make it easy to test
3. Make it easy to maintain
4. Make it easy to change
Develop a “domain diagram”

Captures key objects in our domain and their relationships/interactions.
collegeSim domain model

Blocks represent nouns, links represent verbs

- Room
- Building
- Instructor
- Department
- College
- Meeting
- Course Offering
- Course
- Major
- Time Slot
- Textbook
- Class List
- Student
- Semester
- Author

- uses
- written by
Best Practices approach

1. Develop a “domain diagram”

Captures key objects in our domain and their relationships/interactions.

Initially defined by our core user stories

Nouns
Current stories

• Feature: Display grid
  • User story: As a player I want to view the Sudoku grid in order to figure out my next move. The grid should show the initial values and my current entries. The initial grid should have blanks and a unique solution. The grid should always be consistent.

• Feature: Generate grids
  • User story: As a player I want different grids of varying difficulty in order to be entertained and challenged.

• Feature: Enter number in cell
  • User story: As a player I want to enter a number in a cell in order to solve the puzzle.
  
  • Scenario 1: Cell is blank and new number is consistent. Number appears in cell in a way distinguishable from initial grid values.
  
  • Scenario 2: Cell has number previously entered by player and new number is consistent. Number appears in cell in a way distinguishable from initial grid values.
Nouns

- Player
- Grid
- Cell
- Value, entry, number
- Solution

Are there any implied nouns not listed?
Nouns

- Player
- Grid
- Cell
- Value, entry, number
- Solution
- Screen
- NumPad
- Grid generator

Are there any implied nouns not listed?
How about adjectives...

- Player
- **Grid**
  - Initial, current, solved, consistent
- **Cell**
  - Blank, filled
- Value, entry, number
- Solution
- Screen
- NumPad
- **Grid generator**
What are the relationships, how do they interact?

- Player
- Grid
  - Initial, current, solved, consistent
- Cell
  - Blank, filled
- Value, entry, number
- Solution
- Screen
- NumPad
- Grid generator
Let’s act out the user stories. I need 7 volunteers to start.

Current stories

• **Feature: Display grid**
  - **User story:** As a player I want to view the Sudoku grid in order to figure out my next move. The grid should show the initial values and my current entries. The initial grid should have blanks and a unique solution. The grid should always be consistent.

• **Feature: Generate grids**
  - **User story:** As a player I want different grids of varying difficulty in order to be entertained and challenged.

• **Feature: Enter number in cell**
  - **User story:** As a player I want to enter a number in a cell in order to solve the puzzle.
  - **Scenario 1:** Cell is blank and new number is consistent. Number appears in cell in a way distinguishable from initial grid values.
  - **Scenario 2:** Cell has number previously entered by player and new number is consistent. Number appears in cell in a way distinguishable from initial grid values.
Initial grid and number pad appear.
Original entries are black. Current number “1” is highlighted in number pad.

How should initial grid display work?
Grid generator generates setValue, setColor

Grid Cell row, column, value, color

Grid cells, initialValues displays Screen

Initial values getValue

Screen displays

Grid generator generates setValue, setColor

Initial values getValue

Grid Cell row, column, value, color

Grid cells, initialValues displays Screen

9/10/13
Key design principles

Single responsibility principle

Each class should have a single responsibility and that responsibility should be entirely encapsulated within the class.
Grid generator

Grid cells

Grid Cell
row, column,
value, color
setValue, setColor
generates

displays

Screen

setValue

Initial values
getValue

getValue

Grid Cell
row, column,
value, color
setValue, setColor
generates

displays

Screen

Grid cells, initialValues

Grid generator

generates

setValue

setColor
Initial grid and number pad appear.
Original entries are black. Current number “1” is highlighted in number pad.

How should initial number pad display work?
Who should create the number pad? We’ll come back to that.
User clicks on number pad cell. Current number de-highlighted. New number is highlighted.

Actors:
Player
NumberPad
NumPad cell “1”
NumPad cell “3”
User clicks on cell. Cell is blank or has user-entered value. Current number is consistent for that cell.

Current number appears in that cell in blue.

Actors:
Player
Grid
Grid cell row=6, col=7
NumPad
Key design principles

Single responsibility principle

Each class should have a single responsibility and that responsibility should be entirely encapsulated within the class.

What is responsibility of Grid?
What is responsibility of Grid Cell?

Maintain data, coordinate display
Hold data, UI: display/click
UML Diagrams

• Communication tools
• Include as much detail as needed for your current purpose – and no more
• Start with user story and domain diagrams then evolve your design using “responsibilities”

- Makes design simpler, more intuitive, and more flexible
- Ensures design realizes user stories
Assignment

- Revise our design to conform to model view controller (MVC) architecture (produce pdf)
- Refactor your code
- Support the following scenarios. If the user does anything else, there should be no change.

1. Initial grid and number pad appear. Original entries are black. Current number "3" is highlighted in number pad.
2. User clicks on number pad cell. Current number de-highlighted. New number is highlighted.
3. User clicks on cell. Cell is blank or has user-entered value. Current number is consistent for that cell.
4. Current number appears in that cell in blue.