Architecture design review
Software Design/Development

Top down

Software

Bottom up

Sudoku: Requirement models (e.g. core user stories), domain diagram, MVC

Sudoku: Tutorials, built “proofs of concepts” (e.g. view, buttons)
Software Design/Development

Top down

Game project:
Requirement models (e.g. core user stories), domain diagram, MVC Sudoku

You should have started on both of these already

Game project:
Tutorials, built "proofs of concepts" (e.g. view, buttons) Sudoku
Software Design/Development

Top down

Game project:
Requirement models (e.g. core user stories), domain diagram, MVC
Top down design

• Define “core” game
• Develop architectural model
What is core game?

• Essential behavior
• Resolves high risks
• Illustrates gameplay mechanics
• Can be implemented for alpha
Defining core game

• User stories, use cases, flow chart, story board
• Must concretely describe several minutes of gameplay
Architecture

Architecture Ex: system structure

For example:

Client (browser) — Internet — Server (application)
Architecture

Architecture Ex: system structure

- Views from different levels

For example:

Client (browser)  Internet  Server (application)

Web server  Application  Database
Architecture

Architecture: system structure

• Views from different levels

And:

Client (browser)  Internet  Server (application)

Web server  Application  Database

Model  View  Controller
Architecture

Architecture: system structure
- Views from different levels
- Style (perspective): means of abstracting a system into independent components that communicate through defined protocols/interfaces

Pipes and Filters (GCC)

For example data flow:
Architecture

Architecture: system structure

- Views from different levels
- Style perspective: means of abstracting a system into independent components that communicate through defined protocols/interfaces
Architecture

Architecture: system structure
- Views from different levels
- Style: means of abstracting a system into independent components that communicate through defined protocols/interfaces

Or responsibilities: Object-oriented

Model ↔ View ↔ Controller
Architecture

Architecture: system structure

• Views from different levels

• Style: means of abstracting a system into independent components that communicate through defined protocols/interfaces

Systems can conform to several architectural styles.
Cocos2d-x Architecture

Games

Graphics
- Cocos2D
- Platform Layer
  - Adaptor
  - 3rd libs

Audio
- CocosDenshion
- Platform Layer

Physics
- Box2D
- Chipmunk

Scripting
- Lua
- SpiderMonkey

iOS

Android

WP8

OSX

Windows

Ubuntu
Your architectural model

• What is your Domain Diagram
• What are your main classes?
• What are their (single) responsibilities?
• How do they collaborate to realize your use cases/user stories?
Developing your architectural model

- Define core Uses Cases, User Stories
- Define Domain Model
- Organize according to MVC
- Act out Use Cases
- Lather, rinse, repeat....
Architectural Design Review
(next week Lab)

Design Review Package:
due next week Wednesday night

Design Review Package Evaluation
due end of class Thursday

Design Review
happening Lab Friday

For “core” game
Review?

• What:
  Have skilled people assess your work

• Why:
  Find problems early, also limitations
Types of SE Reviews

- Requirements review
- Architectural review – we will do
- UI design review – we will do
- Component design review
- Test plan review – we will do
- Code review – we will do
Architectural review

The Questions

• Is it capable of meeting the requirements?
• Is it practical to build, test, and support?
• Does it satisfy good design principles?
• Is it documented such that someone can review it?
Preparation

Author
(from Team being reviewed)

Moderator/Team
(from team doing review)

Reviewers
(team doing review)

prepare work products

examine work products

ready for review?

Y

prepare review package

N

schedule review

send out packages

read package

study work products

prepare comments
Work products

• Requirements models (use cases, user stories, etc.)
• Domain model
• Class diagram
• Class responsibilities
• Sequence diagrams
• **Self-critique/assessment**
• Etc.
Architecture Design Review Package

• Table of Contents
• Introduction/Background
• Agenda/Goals for review
• Recommendations for reviewer assignments
• Roadmap to Arch documents
• Self Critique
• Additional Info...
Review Package - Introduction

• Background
  – What project are we discussing
  – What do reviewers need to know about it
    • history, key problems, important decisions, etc.
  – Where can reviewers find more info
    • requirements, designs, analysis

• Goals for review
  – specific work products to be reviewed
  – scope (what is in/out of bounds)
  – what approval means
Self-critique/assessment
Architectural Design

• Overall assessment:

• Rationale:

• Weaknesses:

• Feasibility:

• Extensibility:

• Testability:
Preparation

Author/Team (from Team being reviewed)

prepare work products

Moderator/Team (from team doing review)

examine work products

ready for review?

Y

prepare review package

N

read package

study work products

prepare comments

Reviewers (team doing review)

Class on Thursday

Process completed before 11:59pm Wednesday night.

schedule review

send out packages

28
Architecture Design Review Package

• Table of Contents
• Introduction
• Agenda for review
• Recommendations for reviewer assignments
• Roadmap to Arch documents
• Self Critique
• Additional Info…
Review package - Intro

• Background
  – What project are we discussing
  – What do reviewers need to know about it
    • history, key problems, important decisions, etc.
  – Where can reviewers find more info
    • requirements, designs, analysis

• Goals for review
  – specific work products to be reviewed
  – scope (what is in/out of bounds)
  – what approval means
Review package (hyperlinked)

- Intro
- Agenda
- Criteria
- Work products
- Reviewer assignments

Agenda: Order materials will be reviewed.
Review package (hyperlinked)

• Intro  
• Agenda  
  Criteria: You should augment any rubric for specifics of your project; i.e. what is important for reviewers to look at  
• Criteria  
• Work products  
• Reviewer assignments
Review - Possible Criteria

Are the Uses Cases appropriate?
• Is the architecture able to realize the Use Cases?
• Is the design feasible? Are underlying models clear?
• Does the design follow good Design Principles?
• Does it use Patterns effectively

The author and moderator should amend as appropriate.
Review package (hyperlinked)

• Intro
• Agenda
• Criteria
• Work products: Use Cases, UML Diagrams, etc.
• Reviewer assignments (possible)

What each reviewer is responsible for;
e.g. give each reviewer responsibility for a subset of the
classes or subset of use cases.
Do not divide by work product!
Preparation

Author

- prepare work products

Moderator/Team

- examine work products
- ready for review?
  - Y: prepare review package
  - N: send out packages

Reviewers

- read package
- study work products
- prepare comments

Y
N
Preparation

Author
- prepare work products
- examine work products
- prepare review package
- schedule review
- send out packages

Moderator
- examine work products
- ready for review?
  - Y
    - prepare review package
  - N
    - study work products
    - prepare comments

Reviewers
- read package

Class and Hwk

Chart:
- prepare work products
- examine work products
- ready for review?
  - Y
    - prepare review package
  - N
    - study work products
    - prepare comments

Legend:
- Y
- N
Preparation

Author
- prepare work products
  - prepare review package
    - schedule review
      - send out packages

Moderator/Team
- examine work products
  - ready for review?
    - Y
      - read package
        - study work products
          - prepare comments
    - N

Reviewers
- read package
  - study work products
  - prepare comments

DESIGN REVIEW
In LAB
Review process

• Moderator – Reviewing team
  – keeps review moving
  – ensures all voices are heard and key points covered
  – ensures decisions are made: accepted, major/minor revisions, further review

• Recorder – Reviewing Team
  – takes notes, records all issues raised and decisions reached, all questions, suggestions, and action items
  – publishes a report of the review
Review process cont.

• Reviewers
  – Discuss project as detailed in agenda
  – Provide expertise on assigned aspect of project
  – Raise questions, concerns

• Author and design team (Reviewed)
  – Answers questions but is otherwise silent
Review cont.

• Stick to specified level
• Avoid re-specifying/designing system
• Avoid getting sidetracked
Reviewers Responsibilities

• Participate!
• Participate!
• Participate!

No one reviewer will thoroughly understand everything. But collectively you should. During the discussion you should contribute based on your expertise so that the group can, collectively, assess the work.
THE END
Review package (hyperlinked)

- Intro
- Agenda
- Criteria
- Work products
  - Use Cases, Class Diagrams,
  - Sequence Diagrams, etc.

- Reviewer assignments --- Reviewing Team
Review package (hyperlinked)

- Intro
- Agenda
- Criteria
- Work products
- Reviewer assignments

  What each reviewer is responsible for; e.g. give each reviewer responsibility for a subset of the material, e.g., share Use Cases, UML
Preparation

Author

prepare work products

Moderator

prepare review
package

read package

Reviewers

Y

N

study work products

prepare comments

schedule review
send out packages

In Friday

prepare work
products

examine work
products

ready for review?
Architecture

Architecture: system structure

• Views from different levels

• Style (perspective): means of abstracting a system into independent components that communicate through defined protocols/interfaces