

## Lecture 1:

# Course Overview; Development Process

# Welcome to CS/INFO 3152

---

- Course is **tentatively full**
  - Had 197 applications for 96 (8x12) spots
  - Had to reject 1 out of every 2 students
  - This has been standard for past few years
- But drops are still happening
  - Had six spots open before classes started
  - If you are on waitlist, **talk to me after class**
- Want to stay at eight (8) person teams this year
  - Allowed a nine (9) person option last year
  - Turned out to be a bit unmanageable

# CS/INFO 3152: Game Design

---

- Single semester long game project
  - Interdisciplinary teams of 8 people
  - Design is **entirely** up to you
- First 3-4 weeks are spent preparing
  - **Labs** to develop basic game concepts
  - **Design** activities to solidify your ideas
  - **Group** activities to help you collaborate
- Remainder of class spent on project

# CS/INFO 3152: Game Design

---

- We provide a basic **milestone** schedule
  - **Today's focus**: the development process
  - Deliverables every two weeks (after week 4)
  - Details on course website:  
<http://www.cs.cornell.edu/courses/cs3152>
- Games demonstrated at **Showcase**
  - Once again, will open it up to the public
  - Public reaction is part of your grade
  - Submissions posted on the GDIAC website

# Course Structure

---

- **Lectures:** Mondays, Wednesdays, Fridays
  - Of general design and development interest
  - Will include group activities to use Upson 142
  - Lecture notes posted on website (but **incomplete!**)
- **ENGRC Labs:** Tuesdays (usually)
  - Create documents and presentations
  - Satisfies the technical writing requirement
  - See schedule for exact dates

# Course Structure

---

- **Game Labs:** First four Thursdays
  - Special labs for programming or design
  - Complete according to your project role
  - Only INFO has a choice; CS is programming only
- **Playtesting:** Thursdays for major milestones
  - Submit a **playable** prototype every two weeks
  - Others will playtest your prototype in class
  - We will critique each other's games

# This course is a lot of work!

---

- Expect at least **10 hours/week** outside of class
  - Once the project “starts” in four weeks
  - Typically bare minimum to finish game
  - But if you do this, guaranteed at least a B

## Includes

---

- Time working on game
- Time writing documents
- Time meeting with group

## Does Not Include

---

- 5 days/week in class
- Time spent on readings

# This course is a lot of work!

---

- Expect at least **10 hours/week** outside of class
  - Once the project “starts” in four weeks
  - Typically bare minimum to finish game
  - But if you do this, guaranteed at least a B

If this is a problem, let us know immediately

- Time spent on game
  - 5 days/week in class
- Time writing documents
  - Time spent on readings
- Time meeting with group



# Project Groups

---

- This is a **group-oriented** course
  - **8 person** teams of diverse talents
  - Most programmers have 3110 or 2112
  - At least two character designers/artists
  - One user interface specialist
- Groups have been assigned by the staff
  - We had to balance by talents, not by friends
  - Groups **must** be in the same section

# Game Requirements

---

- Must be unique with innovative **gameplay**
  - Avoid standard **point & click adventures**
  - But can take elements from other games
    - **Example:** platformer + something new
- Must be feasible in a semester
  - Avoid full-blown **RPGs** or **collectable card games**
  - But can have basic elements of these games
- Must have a single player mode

# Game Requirements

---

- Must develop in the game in **LibGDX**
  - Java-based cross platform engine
  - Unity is nice but it does “too much” for this class
  - Can use any IDE, but only IntelliJ is supported
- Must develop a game for a **desktop PC/Mac**
  - Designing gameplay for mobile input is hard!
  - Subject of the advanced class, 4152
- See website for help and resources

# Intellectual Property

---

- Your **group** retains all ownership
  - You can commercialize it later
  - You can make derivative works
  - Individual ownership is your responsibility
- But Cornell gets a non-exclusive license
  - Non-commercial use of final version submitted
  - We can post this version on our website
  - We claim no other rights to your game

# Grading Policy

---

- Mixture of **group** and **individual** grades
- Group grades are same for all group members
  - Group game grade (25%)
  - Document grades (20%)
  - Class presentations (5%)
- Individual grades distinguish group members
  - Individual game grade (20%)
  - Game Labs (20%)
  - Attendance (at demos) (10%)

# Game Grade

---

- Group grade reflects the game quality

Grade	Criteria
A	Bug-free, Fun-to-play
B	Complete and playable
C	Complete but unplayable
D/F	Serious delinquencies

- Individual grade represents contribution

Grade	Criteria
> Group	Visionary, group MVP
= Group	Good attitude, hard worker
< Group	Produce negative work
D/F	Abandon the group

# ENGRC Grading

---

- You **must** enroll in ENGR 3152 as well
  - No extra work; just what you do in discussion
  - Requirement by school of engineering
- All CS/INFO grades except the game and labs
  - Workflow & Group Reports (15%)
  - Course Documents (75%)
  - Attendance & Presentations (10%)
- Typically higher than course grade

# Software Development

---

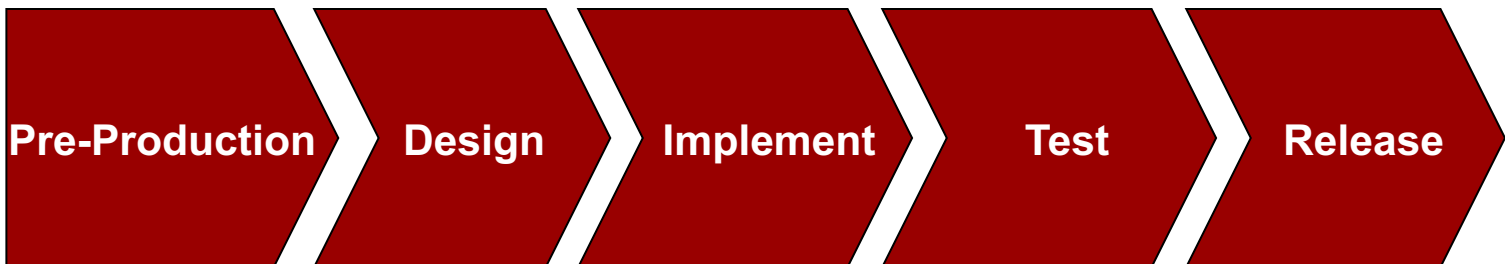
- **Design** process
  - Decide what game you want to make
  - Create a *specification* of your design
- **Development** process
  - Implement your specification
  - Test result to make sure it works
- **Release** (yeah!)



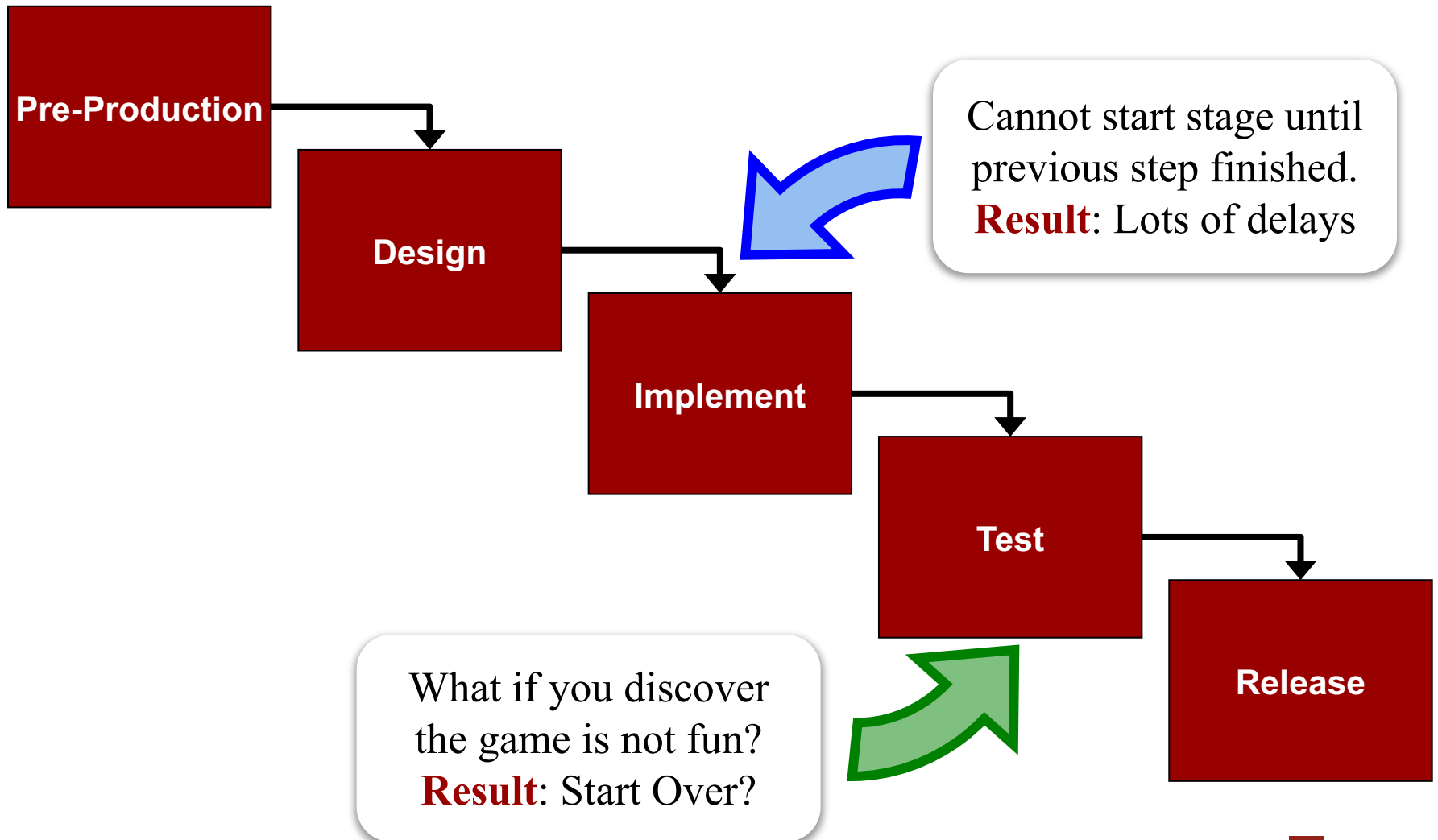
# The Traditional Model

---

- Document extensively; design to specification
  - Design and documents done before coding starts
  - Development follows a specified project timeline
- A general software engineering model
  - Often called the *waterfall* model

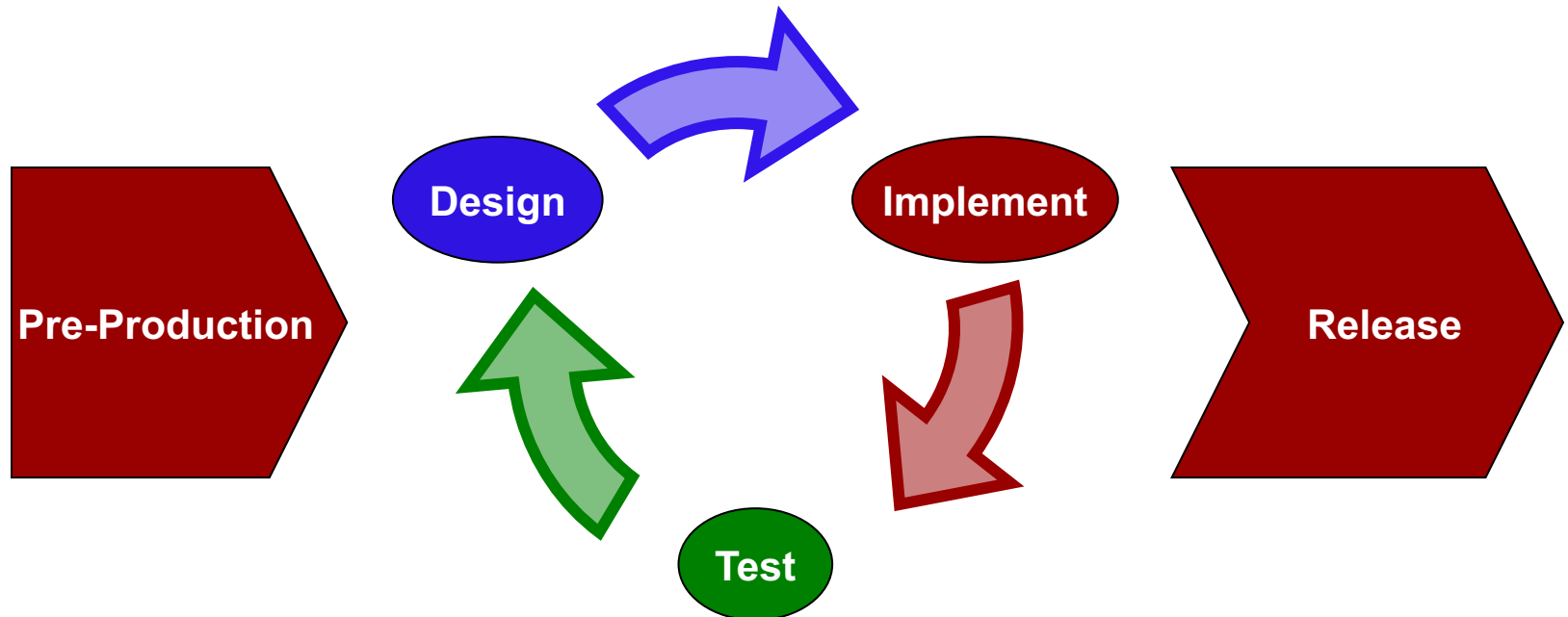


# Waterfall Model



# The Iterative Model

- Cannot evaluate game without playing it first
  - **Iterate**: Rethink design from intermediate results
- Should be playing 20% into development!
  - This requires *prototypes* (may be nondigital)

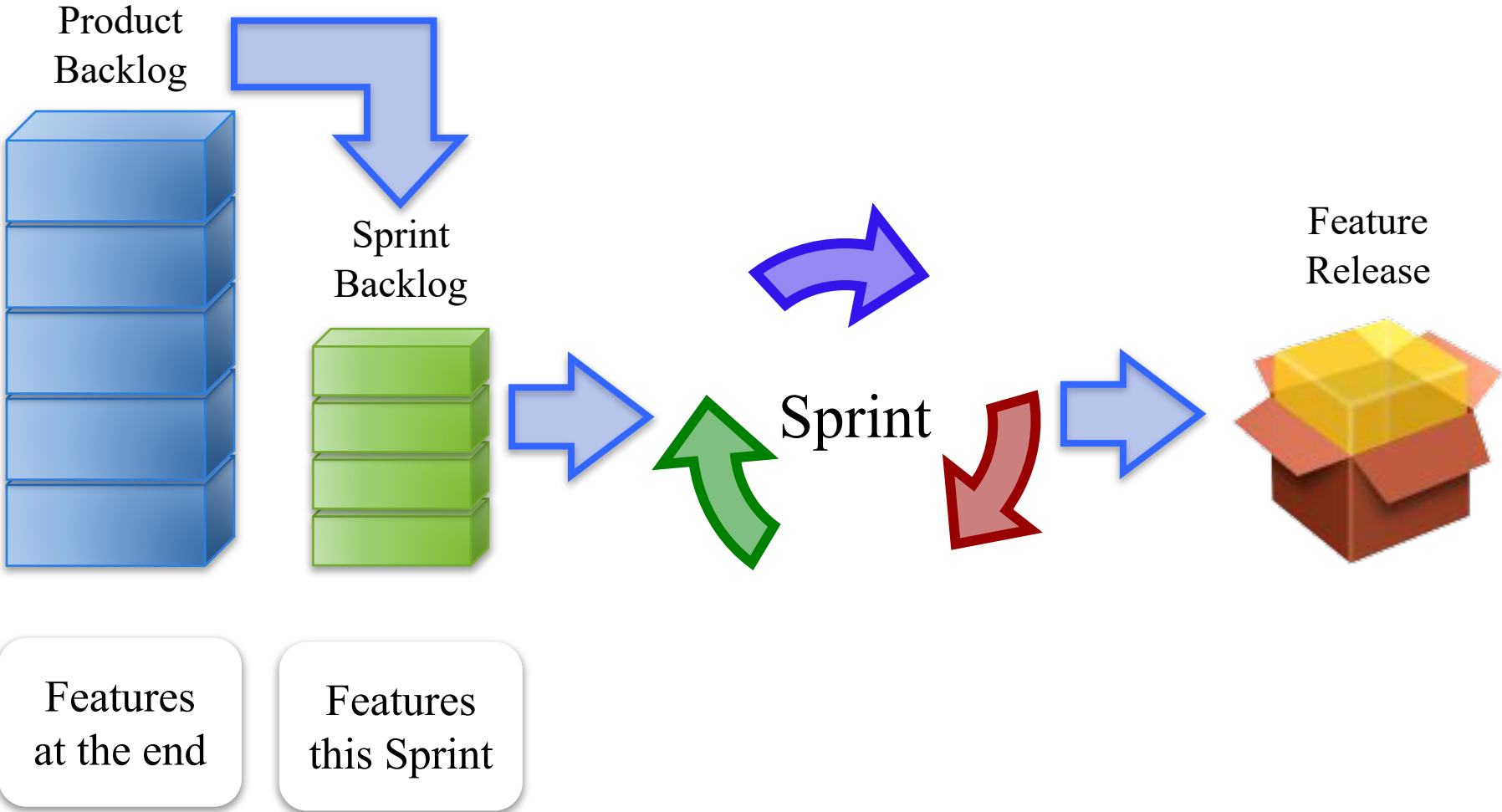


# SCRUM & Agile Development

---

- Iterative model is called **agile development**
  - The most popular agile method is **SCRUM**
- Key (but not only) idea: **SCRUM sprint**
  - Focus on a small, but testable deliverable
  - 3-4 weeks in industry; 2 weeks in this class
- **Sprint Backlog**: features left to implement
  - Chosen to implement for this sprint
  - Re-evaluate features at end of every sprint

# SCRUM Sprint



# Milestones

- *Suggestions* for your sprint backlog
  - Flexible enough to handle set-backs
  - Can renegotiate if you get seriously behind

<b>Week 5</b>	<b>Nondigital Prototype</b>	2/20
<b>Week 7</b>	<b>Gameplay Prototype</b>	3/6
<b>Week 9</b>	<b>Technical Prototype</b>	3/20
<b>Week 11</b>	<b>Alpha (Code Complete)</b>	4/10
<b>Week 13</b>	<b>Beta (Feature Complete)</b>	4/24
<b>Week 15</b>	<b>Golden Master (Installable)</b>	5/5
<b>Week 16</b>	<b>GDIAC Showcase</b>	5/20

# Documentation

---

- Major part of the development process
  - Why course counts for technical writing
  - Ensures group is always on “same page”
- At every point of development
  - **Pre-production**: concept document, gameplay
  - **Sprints**: reports, architectural specification
  - **Release**: game manual, post-mortem
- Challenge is understanding your *audience*

# Pre-Production Documentation

---

- **Concept Document**

- Describes the basic idea behind your game
- Communicate core vision without too many details
- Focus of ENGR Lab next week
- **Audience:** a game publisher (to get funding)

- **Gameplay Specification**

- Thorough overview of your gameplay
- Include formal design elements shown in class
- **May change as part of your sprints!**
- **Audience:** new team members (hired later)



# Sprint Documentation

---

- **Reports (every 2 weeks)**

- Outlines the upcoming sprint (who does what)
- Reflects on previous sprint (did you meet goals?)
- Written as a *group document*
- **Audience:** your game producer

- **CATME Reports**

- Online tool for reporting your progress
- Completed as *individuals*
- Allows us to look for problems in group

# Using CATME for Reports



## Report

[View Comments](#)[View Raw Data](#)[Return to Main Page](#)

Class Term Format Prof School  
am Review ME 316Fall 2015Lecture Leachman Washington State University

Enable pop-up texts  Show raw "Adjustment Factor"

[Re-Display](#)Search: 

<input type="checkbox"/>	Team ID	Contrib. to Team	Interact w/ Team	Keeping on Track	Expect Quality	Adj Factor (w/ Self)	Adj Factor (w/o Self)	Note
<input type="checkbox"/>	01	4.2	4.4	4.0	4.2	1.05	1.05	Under
<input type="checkbox"/>	01	3.6	4.2	4.0	3.4	1.00	1.00	
<input type="checkbox"/>	01	3.8	4.0	3.6	3.8	1.00	1.01	
<input type="checkbox"/>	01	3.0	4.2	3.6	3.4	0.91	0.87	
<input type="checkbox"/>	01	3.8	4.2	4.2	4.0	1.04	1.04	
<input type="checkbox"/>	02	3.8	4.2	3.8	4.0	1.00	1.00	
<input type="checkbox"/>	02	3.8	4.2	3.8	4.0	1.00	1.00	
<input type="checkbox"/>	02	4.5	4.2	3.8	4.2	1.04	1.02	
<input type="checkbox"/>	02	4.2	4.2	3.8	4.0	1.01	1.01	

<http://www.catme.org>

# Detailed Specifications

---

- **Architecture Specification**

- Outline of your software organization
- Used to distribute tasks to programmers
- **Audience:** team programmers

- **Design Specification**

- Outline of your design vision
- Also includes technical details of asset handling
- Used for to help designers work together
- **Audience:** team designers

# Detailed Specifications

---

- **Architecture Specification**

- Outline of your software architecture
- Use cases: **Directed by the Lead Programmer**
- **Audience:** team programmers

- **Design Specification**

- Outline of your design vision
- Also include design patterns and handling
- Use cases: **Directed by the Lead Designer**
- **Audience:** team designers

# Release Documentation

---

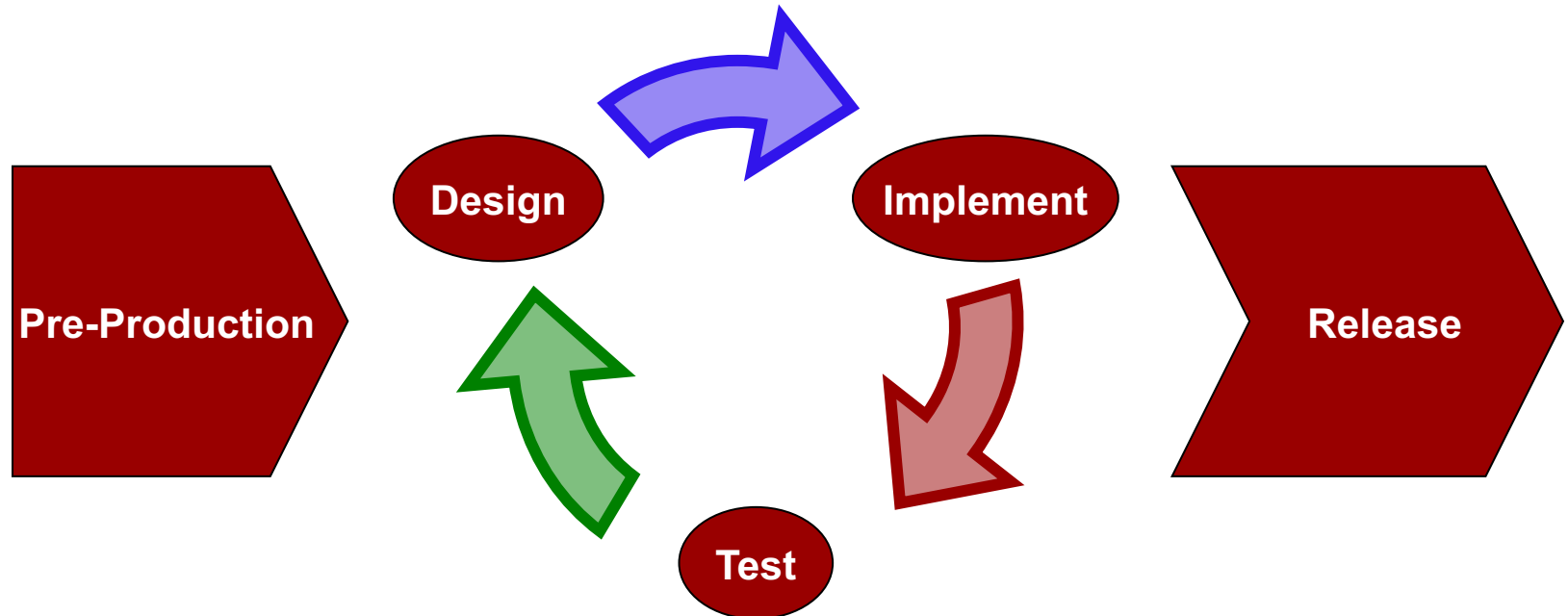
- **Game Manual**

- Concise description of gameplay
- Instructions on how to play the game
- Story, other material to improve the setting
- **Audience:** your players

- **Postmortem**

- *Honest* reassessment of what happened
- What went right; what went wrong
- **Audience:** yourself (for next time...)

# Development Process Review



- **Pre-production**

- Initial design
- **Concept Document**
- **Gameplay Spec**

- **Two-Week Sprints**

- Playable prototypes
- **Reports**
- **Arch/Design Spec**

- **Release**

- Public Showcase
- **Game Manual**
- **Postmortem**

# Semester Schedule

<b>Week 1</b>	Team Workflow	1/28	Pre-Production
<b>Week 2</b>	Initial Proposal	2/4	
<b>Week 3</b>	Concept Document	2/11	
<b>Week 4</b>	Concept Revision <b>(Project Kickoff)</b>	2/18	
<b>Week 5</b>	<b>Nondigital Prototype</b> Milestone Proposals	2/20 2/25	
<i>February Break</i>			
<b>Week 6</b>	Gameplay Specification	3/4	Development
<b>Week 7</b>	<b>Gameplay Prototype</b>	3/6	
<b>Week 8</b>	Detailed Specifications	3/18	
<b>Week 9</b>	<b>Technical Prototype</b>	3/20	

# Semester Schedule

<b>Week 10</b>	Document Revisions	4/1	Development
<i>Spring Break</i>			
<b>Week 11</b>	<b>Alpha Release (Code Complete)</b>	4/10	
<b>Week 12</b>	Game Manual (Draft based on Alpha)	4/22	
<b>Week 13</b>	<b>Beta Release (Feature Complete)</b>	4/24	
<b>Week 14</b>	Final Portfolio	5/6	
<b>Week 15</b>	<b>Golden Master (Installable)</b>	5/8	Release
<b>Week 16</b>	GDIAC Showcase	5/20	