Comp 790-063
Course Overview

Instructor: Prasun Dewan (FB 150, help401@cs.unc.edu)

Course page:
http://www.cs.unc.edu/~dewan/290/f14/index.html

Course Page

Comp 790-063: Distributed Collaborative Systems

Course Overview

Distributed collaborative systems allow users from different computers to interact with each other. Thus, they are layered on top of two base technologies: user-interface and distributed abstractions. They include shared desktop/window systems, instant messaging, shared editors and spreadsheets, distributed presentations systems, and of course, games. In this course, we will look at issues in the design, implementation, testing, and evaluation of these systems.

At the end of the course, you will have a basic understanding of how state of the art collaboration software available works, the potential uses of this software, the design space of collaborative applications and infrastructures, and the two base technologies: user-interface and distributed abstractions.

The topics covered will include collaboration architectures, consistency of replicated objects, collaborative user-interfaces, application and system taxonomies, session-based application-level multicast, performance, causality, operation transformation, and concurrency and access

Linked from my home page (google my name to find it) and UNC course pages

Class Material

Tentative Schedule

	Unit (Start Date)	Slides	Chapters	Assignment
	Introduction			
	Application-level Multicast and Architecture of a Basic Collaborative Application			Implement IM and Editing User Interface
	Awareness			Implement basic IM/Editing Using Application-Level Multicast
	Jitter Recovery			Implement Replicated AWT/Swing Windows support in project
	Response Times			Implement telepointers with jitter recovery

No book

Each unit will have PPT slides (with recordings), sometimes Word document, and assignment

COLLABORATION

COMPUTER SUPPORTED COLLABORATION

TECHNIQUES AND ABSTRACTIONS FOR SUPPORTING COLLABORATIVE APPLICATIONS

Abstractions

Reusable hardware or software component that can be used by some large set (domain) of applications

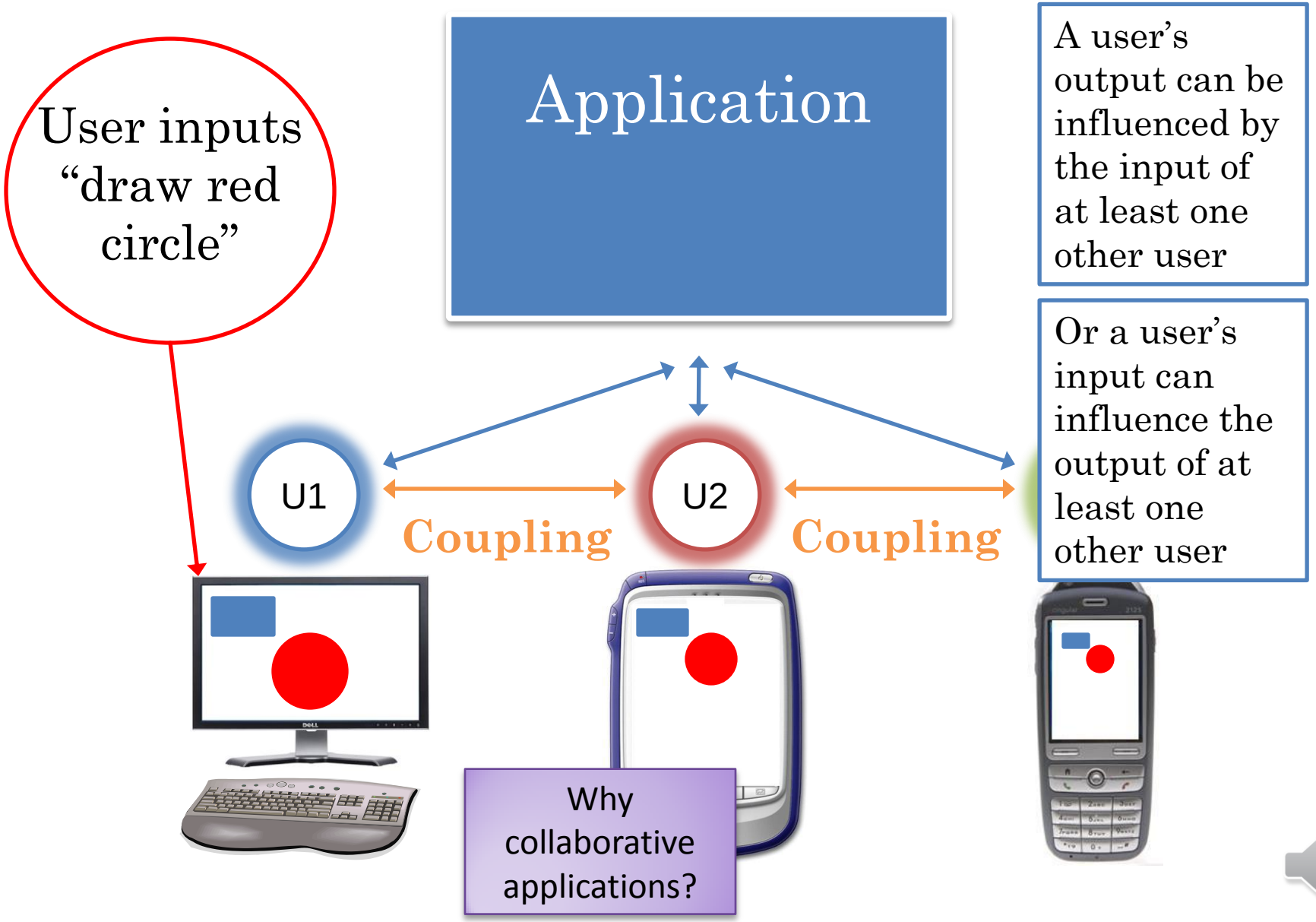
Array, class, file, database, textfield

Techniques

Concept applicable to a large set of applications that may or may not be converted to an abstraction

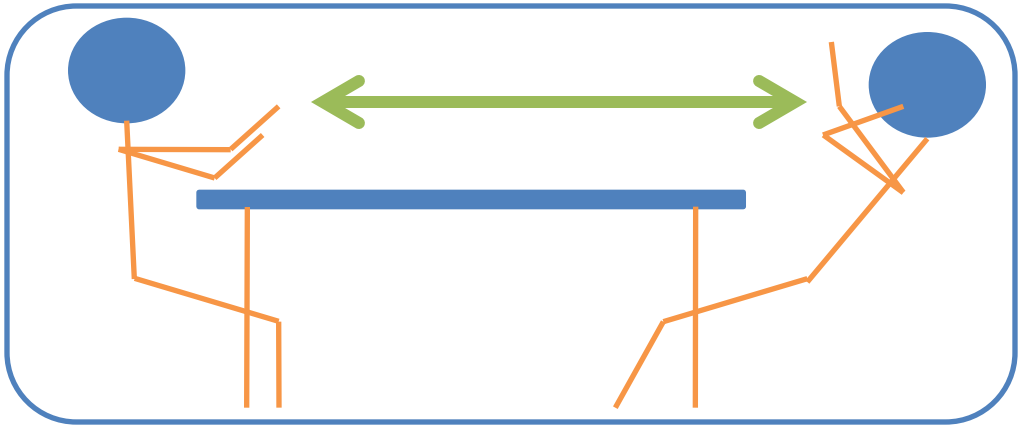
User-interface (graphical, …); algorithm (bubble sort, ..), design pattern (observer, ..); architecture (client-server, p2p, …)

COLLABORATIVE APPLICATION

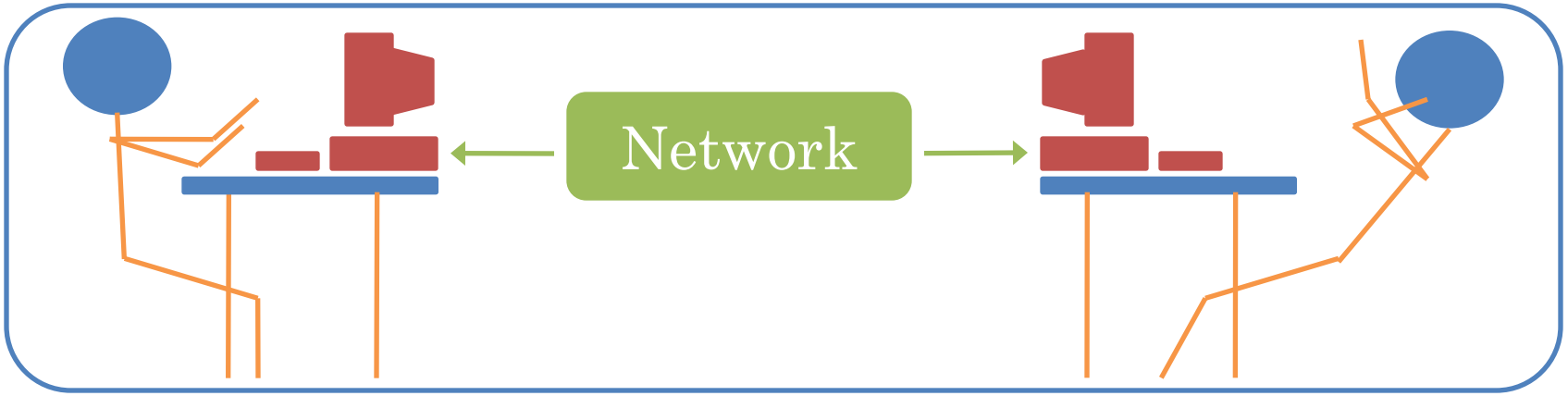


TOWARDS BEING THERE

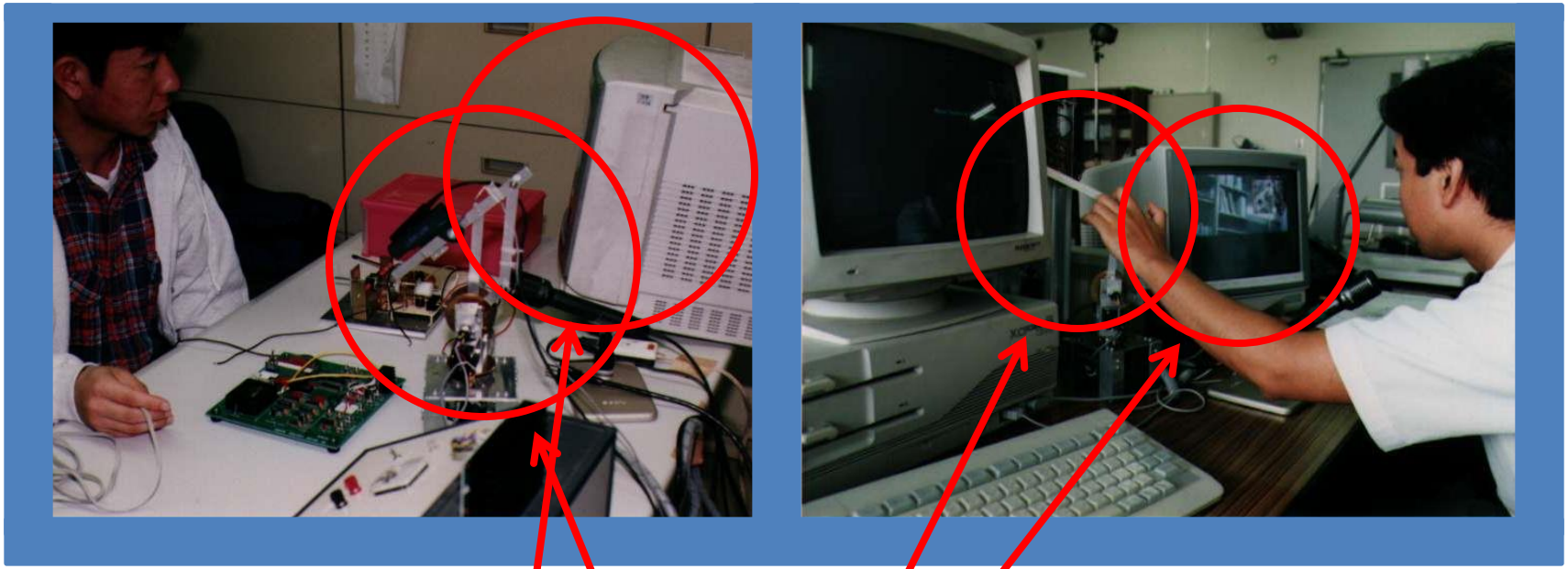
Face-to-face Interaction



Computer Supported Interaction



GESTURECAM: REMOTE SURROGATE



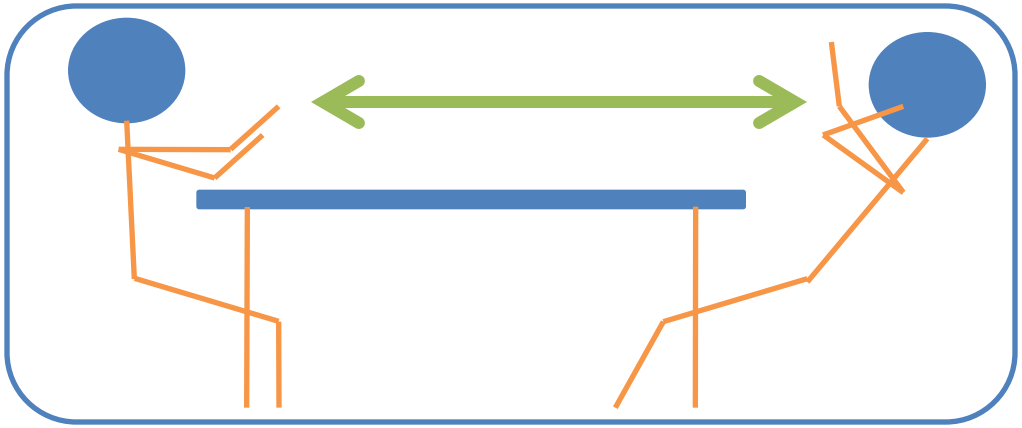
Control GestureCam in another location

Both users see what the GestureCam is "looking" at on TV screens

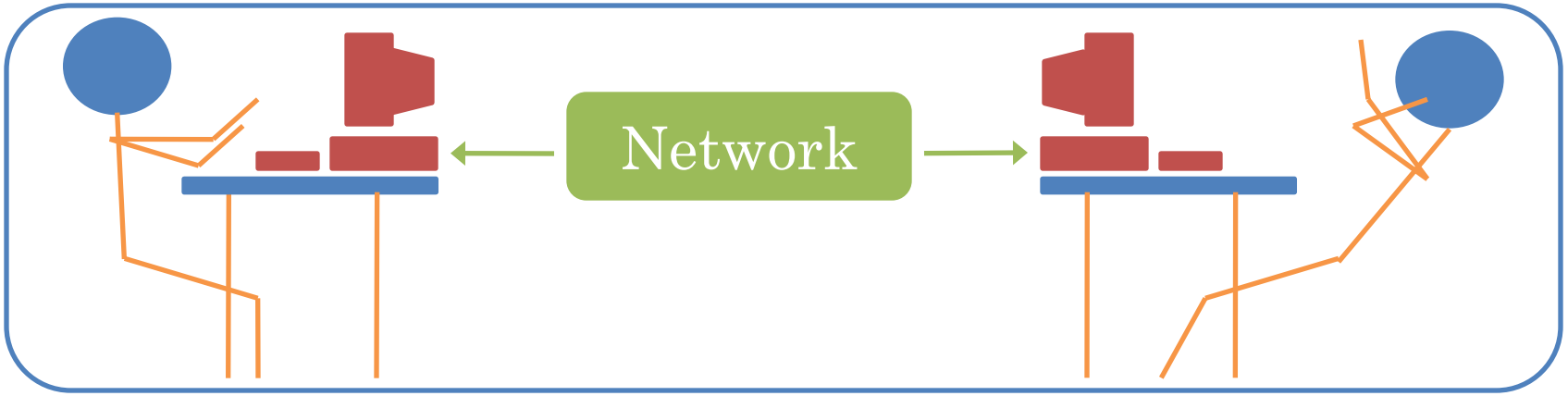
Controller can "zoom in" visually and physically to point at an item

TOWARDS BEING THERE

Face-to-face Interaction



Computer Supported Interaction

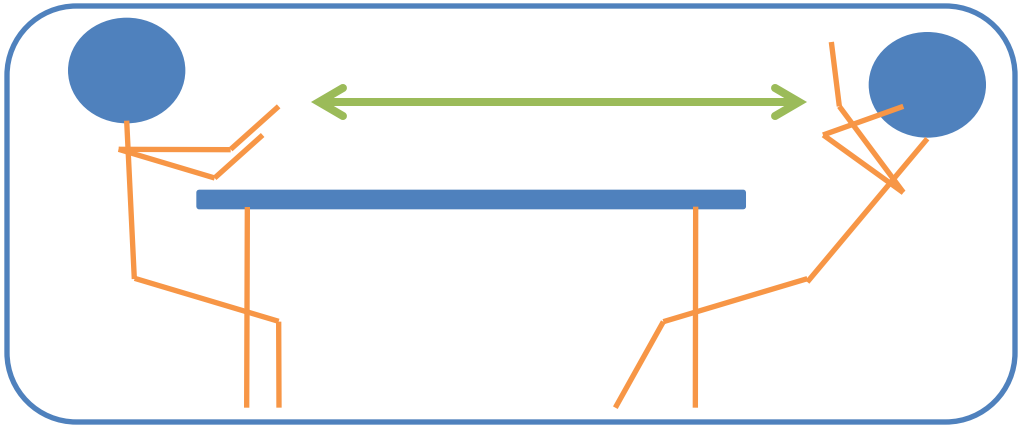


Computer Supported < Face to Face

Second best – used to overcome time and cost constraints

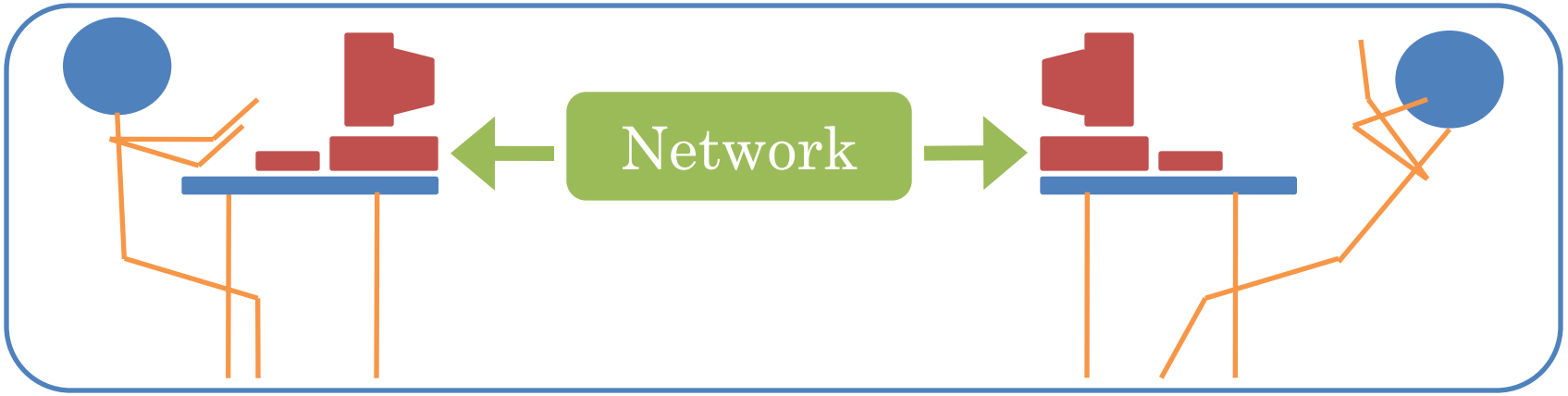
BENEFIT: BEYOND BEING THERE

Face-to-face Interaction



Hollan & Stornetta

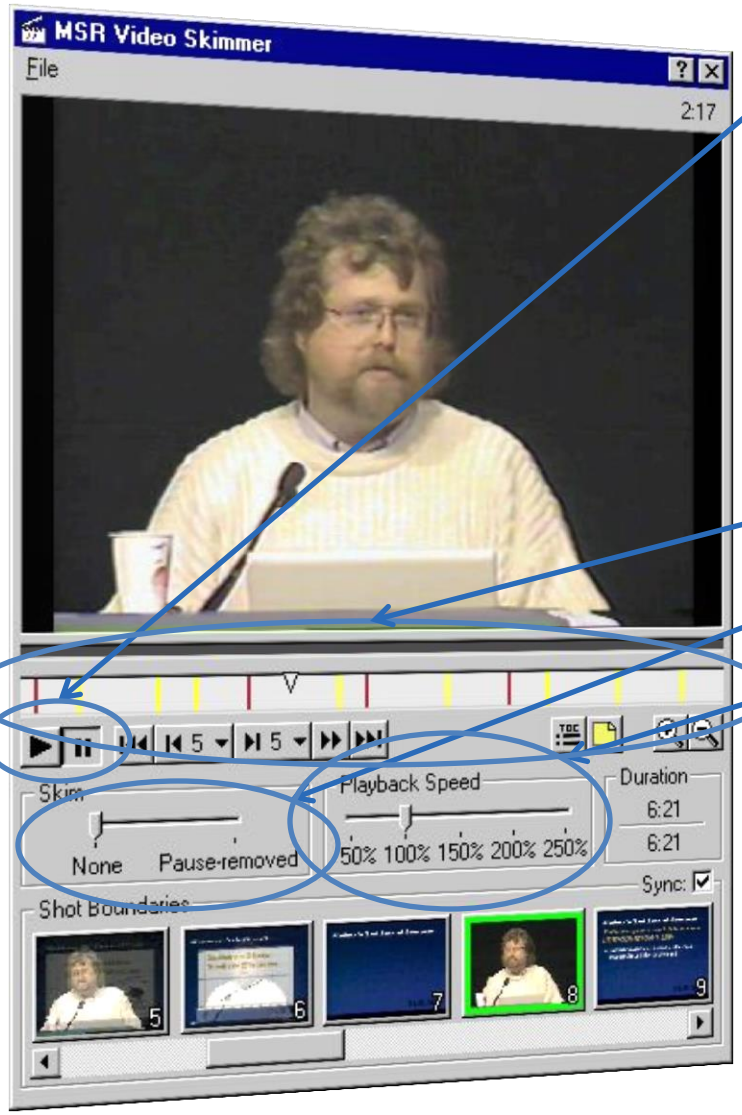
Computer Supported Interaction



Computer Supported > Face to Face

Use it even when face to face option available

MEETING VIDEO BROWSING



Basic Video Controls:
Play, Pause, etc.

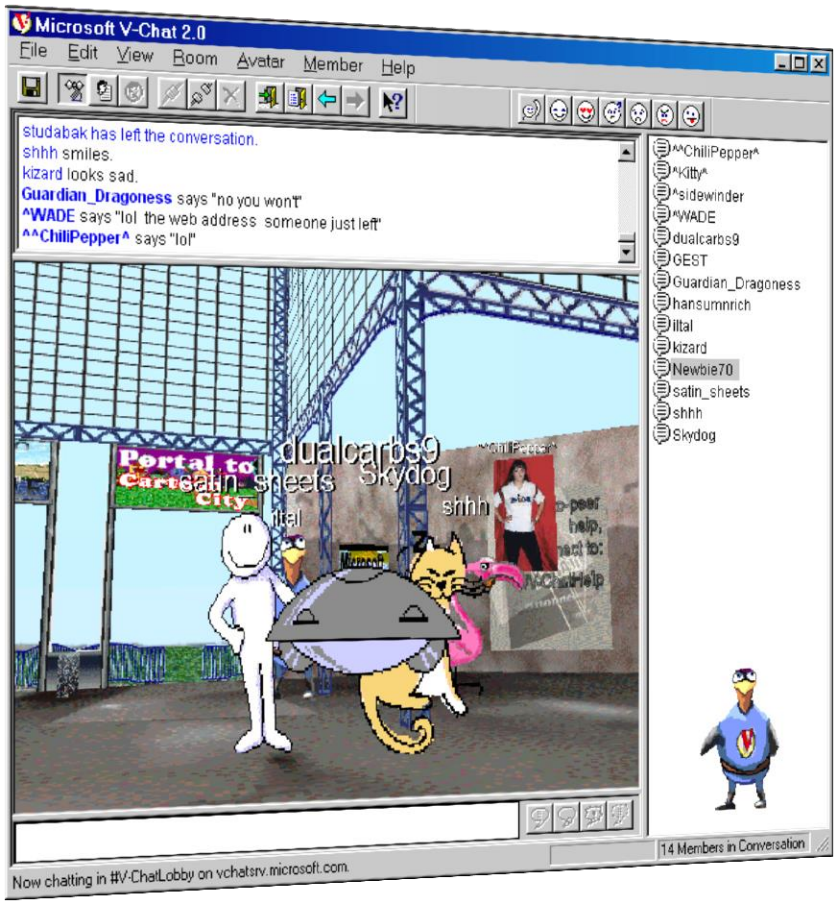
Beyond Being There

Video Processing

- Bookmark Video
- Pause Removal
- Time Compress Without Changing Pitch
- Automatic TOC Generation
- Automatic Slide Summaries

COMBINING BOTH GOALS

Chat Rooms



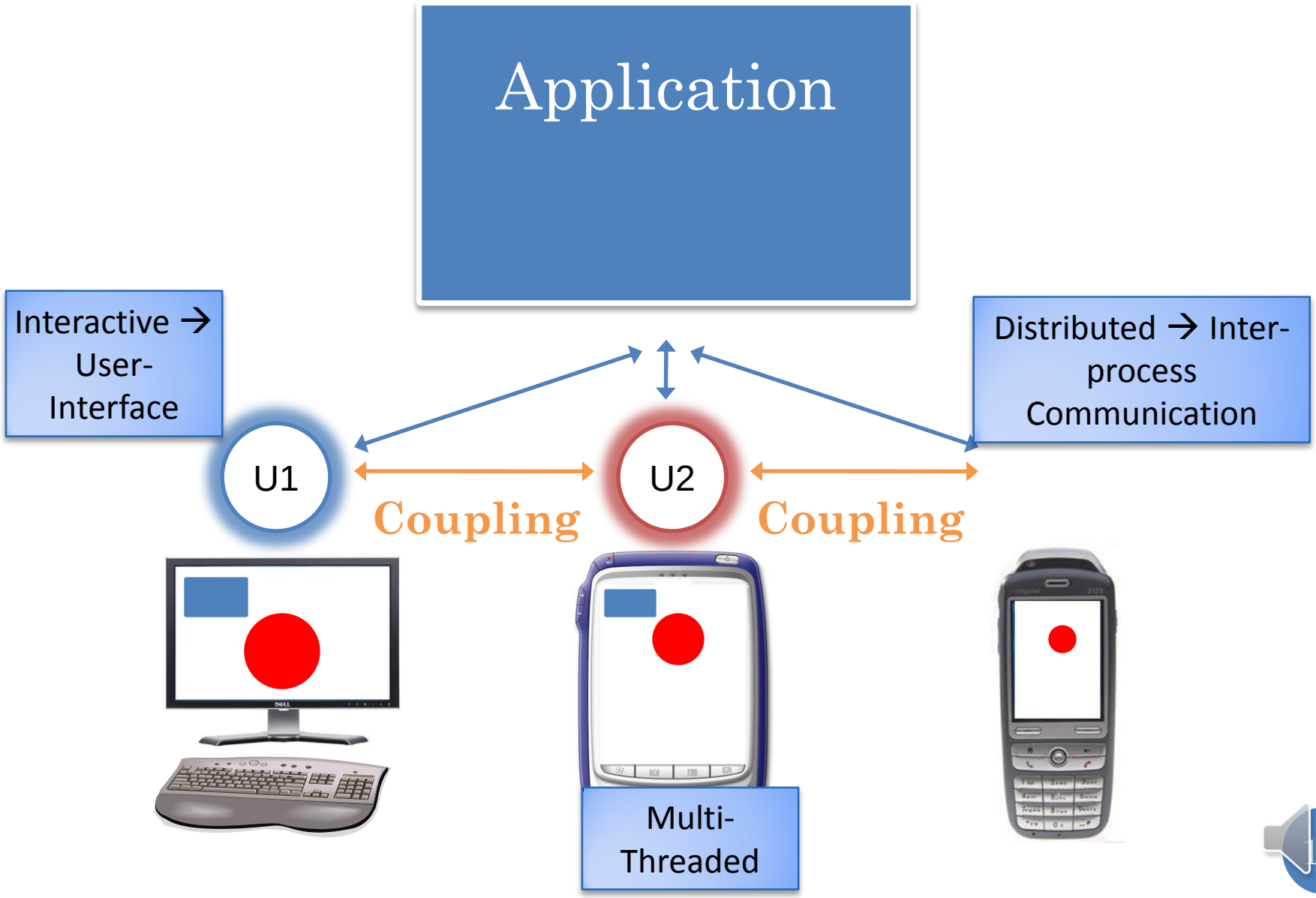
Toward Being There

- Feel Transported to a Common Virtual World

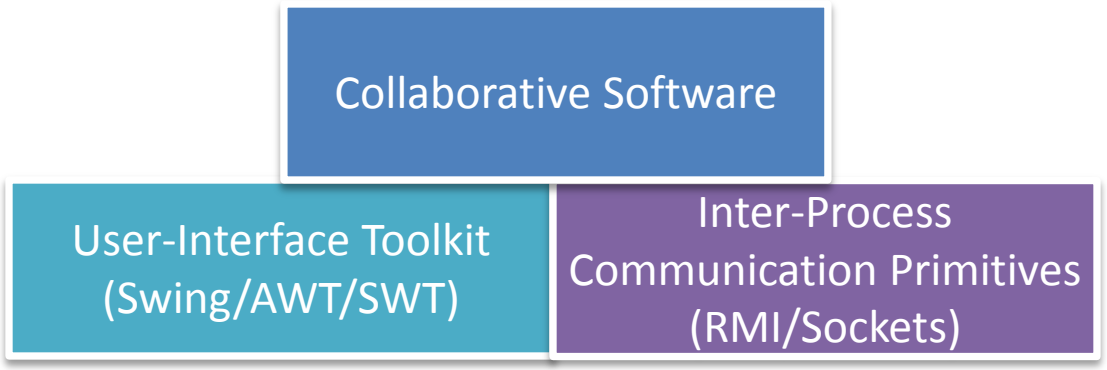
Beyond Being There

- Avatar Selection
- Anonymity

Base Technologies



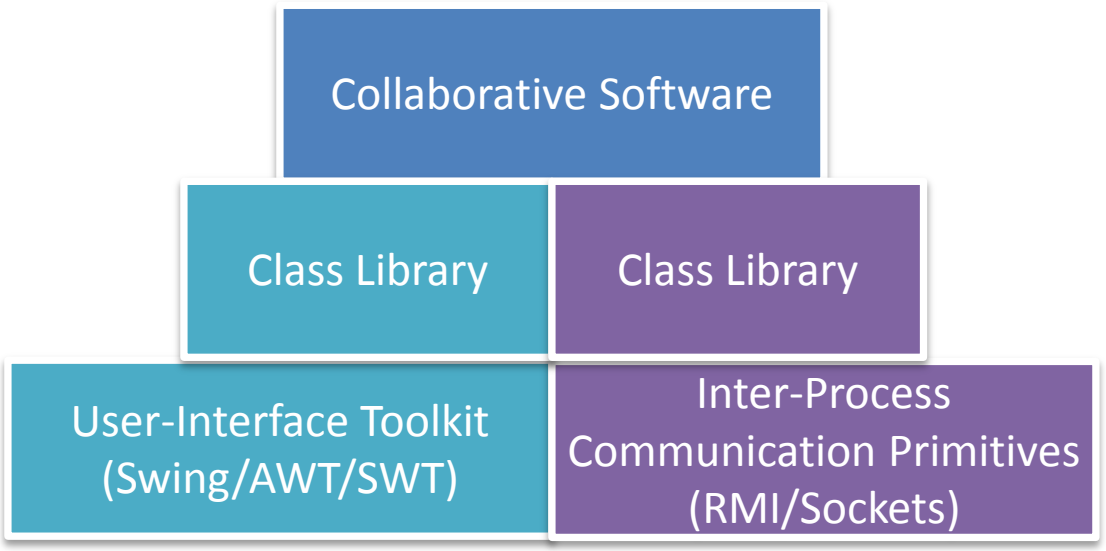
Base Technologies



Will learn to apply these technologies and not learn in depth the principles behind them

These technologies are not easy to use!

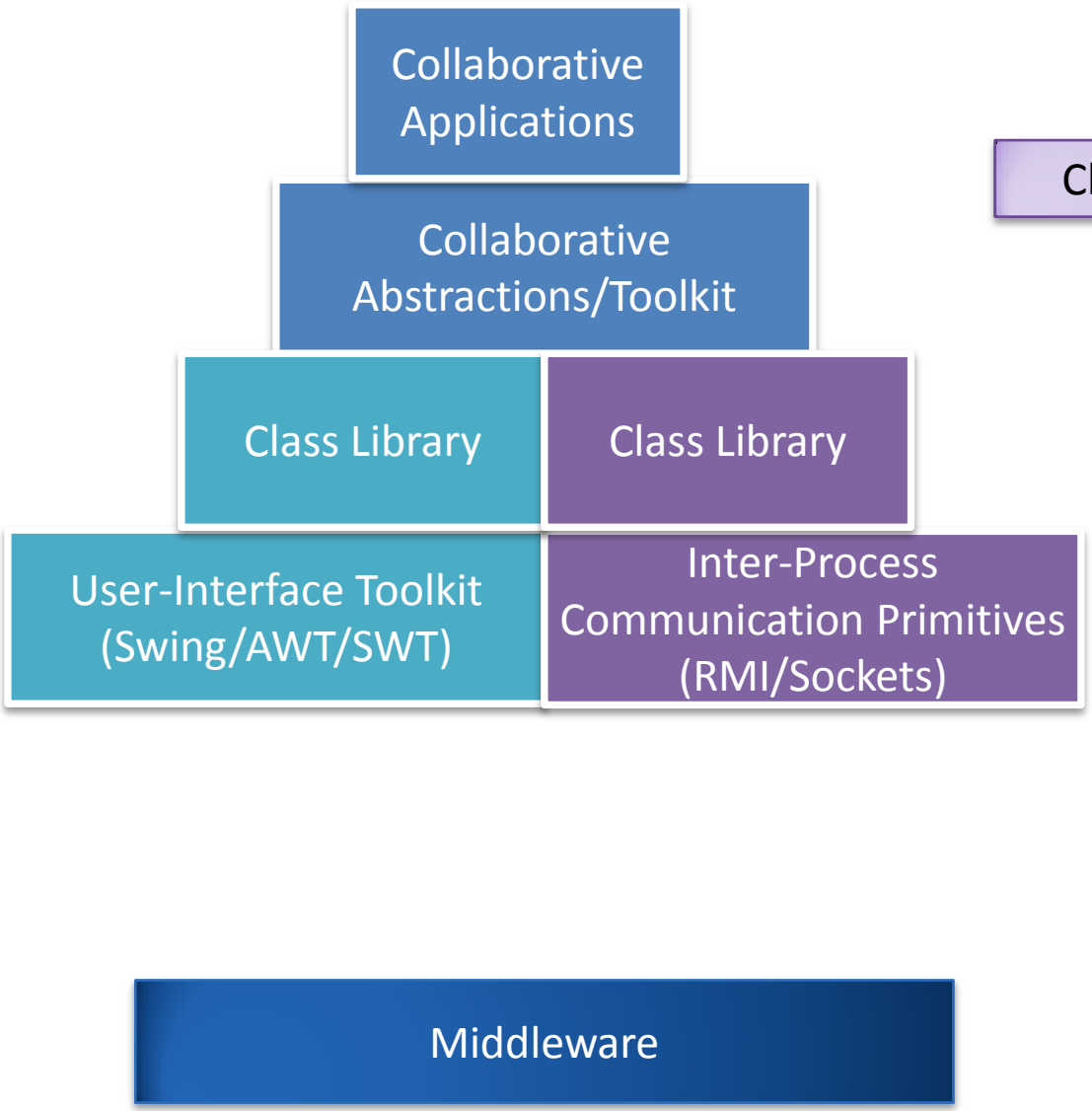
Class Abstractions



Will provide Java libraries to allow you to focus on the **what** rather than the **how**

Java/Object-Oriented Programming!

Collaborative Abstractions vs Applications



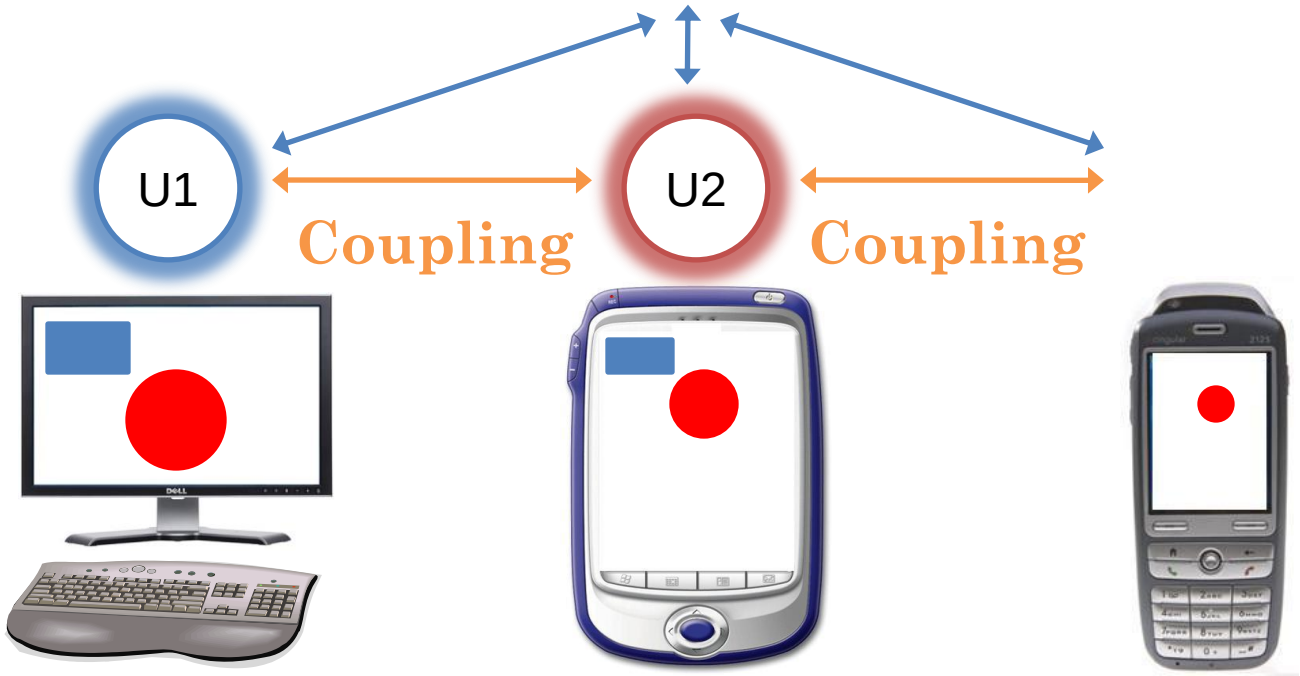
Challenges?

Base Technologies

Design of coupling UI, algorithms, design patterns, architectures and abstractions

Application

Other functions?



ISSUES: SEMANTIC

Collab. Function	Description
Session Management	How do distributed users create, destroy, join, and leave collaborative sessions?
Coupling	What is the remote feedback of a user command and when is it given?
Awareness	How are users made aware of uncoupled commands of their collaborators, that is, “out of band” commands?

ISSUES: SEMANTIC

Collab. Function	Description
Access Control	How do we ensure that users do not execute unauthorized commands?
Concurrency Control	How do we ensure that concurrent users do not enter inconsistent authorized commands?

ISSUES: SEMANTIC

Collab. Function	Description
Merging	How do we merge conflicting commands entered by different users?
Undo/Redo	What are the semantics of undo/redo operations in a collaboration session?

Many more!

ISSUES: IMPLEMENTATION

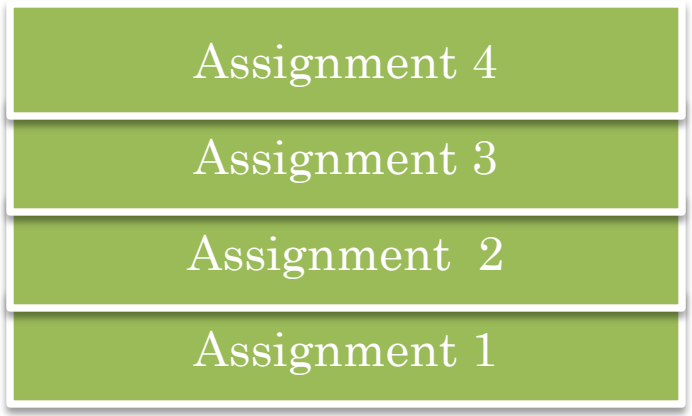
Issue	Description
Objects	What kind of objects are composed in a collaborative application?
Collaboration Awareness	Which of these objects are collaboration aware? How are these objects integrated with existing collaboration-unaware objects?
Concurrency	How is the application decomposed into concurrent threads?
Distribution	How are the application objects placed on different hosts?

ISSUES: IMPLEMENTATION

Issue	Description
Replication/Migration	Which of these objects are centralized (in a server) and which are replicated (on peers)?
Performance	What kind of services are provided to overcome jitter and latency?

Many more!

Layered Assignments = Project



Assignments will build on each other to create a semester project

Due dates normally separated by two weeks (holidays, exams can cause more separation)

Getting Help

- Can discuss solutions with each other at a high level
- Not at the code level
- Sharing of code is honor code violation
- Can help each other with debugging as long as it does not lead to code sharing
- Will give algorithms for assignments

Piazza for Getting Help and Class Discussion

Getting Help and Class Discussion

We will be using Piazza for class discussion and getting help. The system is highly catered to getting you help fast and efficiently from classmates, the TA, and myself. Rather than emailing questions to the teaching staff, I encourage you to post your questions on Piazza. If you do not get a response within a day or two on Piazza, please send mail to help401@cs.unc.edu. But try Piazza first. Do not send mail to an individual instructor, as that can overwhelm him - such mail will be ignored.

Before posing a question, please check if this question has been asked before. This will reduce post clutter and reduce our burden. Repeat questions will be ignored by the instructors.

Piazza allows anyone to respond. So if you see a question that you think you can respond to, please do so, as that will reduce our burden and help you "teach" your fellow students.

This will be a form of class participation that will be noted when I allocate my fudge points!

Hope it works well

If you have any problems or feedback for the developers, email team@piazza.com.

Signup Link:piazza.com/unc/fall2014/comp790063

Class page: piazza.com/unc/fall2014/comp790063/home

Grade Distribution

Exams (Two midterms, no final)	40%
Assignments (Home work)	60%
Fudge Factor (Class/piazza participation, other distinguishing factors), particularly useful for borderline cases	10%

Project Heavy but No final!

Projects

Demo of Projects