CS3505/5020
Software Practice II

Quiz #1 in class today
Compose your teams
Activity diagrams
Project #4 guidance
Compose your teams

- In class you will have some time to form your team
  - For those of you not in class, I am still working on getting the Wiki posted.
    » Security has been the issue
    » I hope students will help make the Wiki a nice place to form teams and share assignment ideas
    » The Wiki is your space, not mine – use it wisely
Activity Diagrams (Chapter 11)

- Used to model sequential and parallel work flows
  - Often useful to model what really happens in a business to help understand that work flow to help you understand how to write the code

- Much like the flowcharts of old

- Primary components:
**Example**

Browsing Site

-New Customer

New Customer Registration

-Log Into Site

Wish List Items

Shop for Items

Checkout
Advanced Activity Concepts

- Subactivities
Partition (Swim lanes)

Note – Petri Net model of computation
Other Features

- See Chapter 11 for a description of the many other features in activity diagrams such as:

  - Send Signal
  - Receive Signal
  - Time Event Signal
  - Pin
  - Pin indicates a parameter to an Action
  - Expansion
  - Handles multiple Items.
  - Process
  - <<concurrent>>
  - [accept]
  - [reject]
  - Final Flow
State machines v. Activity diagrams

- State machines emphasize condition

- Activity diagrams emphasize action
  - High level of support for concurrency

- Generally, activity diagrams imply state between actions, state diagrams imply actions between states.
  - Lots of crossover
Required reading

- This week / weekend:
  - Chapter 10 – state machine diagrams
  - Chapter 11 – activity diagrams

- Next week
  - Chapter 4 – sequence diagrams

- This book is easy reading! Don’t put it off, and feel free to look at other modeling chapters that may help.
  - Chapter 3, 11, 15, etc.
Required reading

- The third edition of the book is significantly different than the second.
  - If you have the second edition, seek web resources to cover features not in the book
Required reading

Quiz #2 (in two weeks) will cover these modeling techniques and software engineering
  - Make sure you know how to interpret each type of diagram
Project #4 summary

- Design a protocol for handling lobby and in-game communications for a multiplayer game that:
  - Provides no appearance of lag wherever possible
  - Keeps players’ game states in sync (whether in the lobby or in the game)
  - Allows players to join / drop in the lobby or drop in the game without serious side effects

- Use UML diagrams to clearly specify your solution
  - Your solution must not have ambiguities
Project #4 guidance

Consider the scope of the project:
- Things to worry about:
  - Low-level: What data will be sent around?
    - (XNA helps with lobby connections, but not game sync data)
  - Mid-level: How will network communications control the state of the game or lobby?
    - Choose appropriate representation
    - Use appropriate diagrams
  - High-level: What states will the game or lobby be in?
    - How are transitions initiated?
    - What about errors and unexpected events?
    - Use appropriate diagrams
Project #4 guidance

● Consider the scope of the project:
  – Things to NOT worry about:
    » Low-level: What calls will we make?
      ● Leave coding details for the coder.
    » Mid-level: What classes will we need?
      ● Leave code design details to the coder.
    » High-level: What will the game be?
      ● The look-and-feel of the lobby and game are not your concern (yet).
      ● Leave the game to me. You can assume that it will be real-time and interactive using the keyboard and mouse.
      ● Design changes are my prerogative, and should be expected. ;)
Project #4 guidance

- My game will have a well-defined notion of state.
  - Assume the coder will be able to make a single call to advance the game to the next state.
    - The coder will supply keyboard and mouse states to this method
    - I will guarantee that my game will be entirely deterministic and repeatable. For identical inputs and state, the game will always transition to the same next state.
  - The coder will be able to set the game loop update rate once before beginning.
  - Network communications will be adjusted to have a latency of ~200 ms and a reliability rate of 90%.
    - XNA directly supports this debugging mode.
Consider this rudimentary networked game loop

- Find our mouse/key inputs
- Send inputs to other players
- Receive inputs from other players
- Update the game state with all inputs
Fill in the missing details

● The previous diagram ignores:
  – Possible errors
  – What data is actually sent
  – The fact that the lobby also uses the update/draw methods
    ● The game may not be running, etc.
  – Other

● Fill in these missing details
Fill in the missing details

- The game will have state. Will your network communications also have state?

- How will you recover from errors and avoid lag?
A few key ideas in multiplayer gaming:

- You can have a client / server model
  - Clients send data to a server (usually just one particular game process)
  - The server then sends responses to the clients
    - Activity is only coordinated with the server

- You can have a peer-to-peer model
  - Game processes communicate directly with other game processes
    - Activity is coordinated between peers
A few key ideas in multiplayer gaming:

- In a peer-to-peer model:
  - Each peer needs to be able to predict:
    - The other peers’ game states
    - The other peers’ network states
  - If a peer does not know what another peer is doing, communication is needed.
    - This is required for user input, of course.
    - Only required of game/network state if communications is lost. Recover.
To succeed with project #4:

- **Start early!!!** (Did I say ‘!!!’?)
  - Be thinking about how game processes will stay synchronized with each other
  - Missing pieces of the design will only become apparent when you have time to reflect

- **Use diagrams to coalesce your thinking.**
  - An idea is worthless until you write it down
  - Your partner needs to be able to critique the design