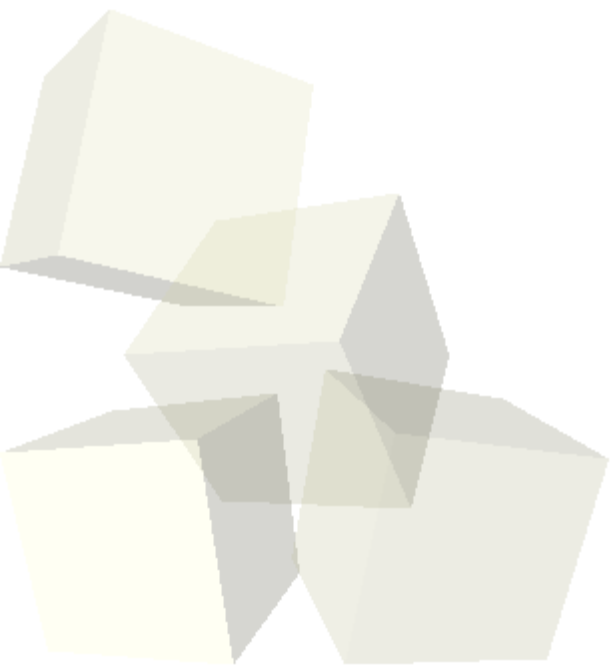




Use Case and Activity Diagrams

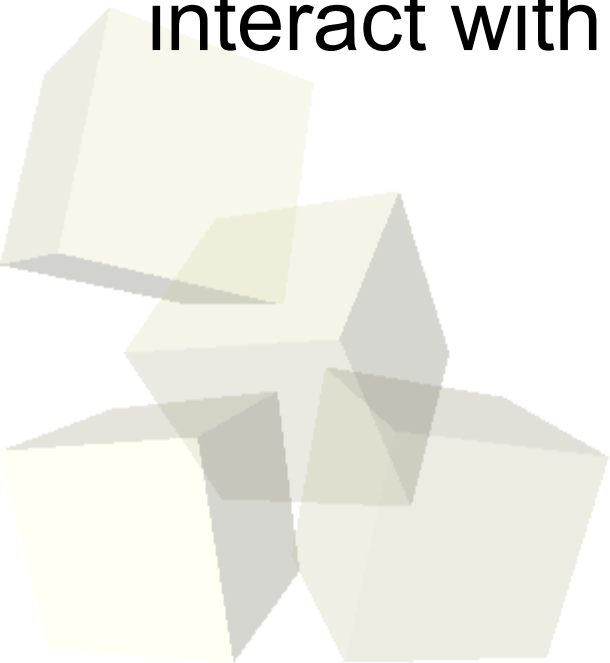
2/4/2008





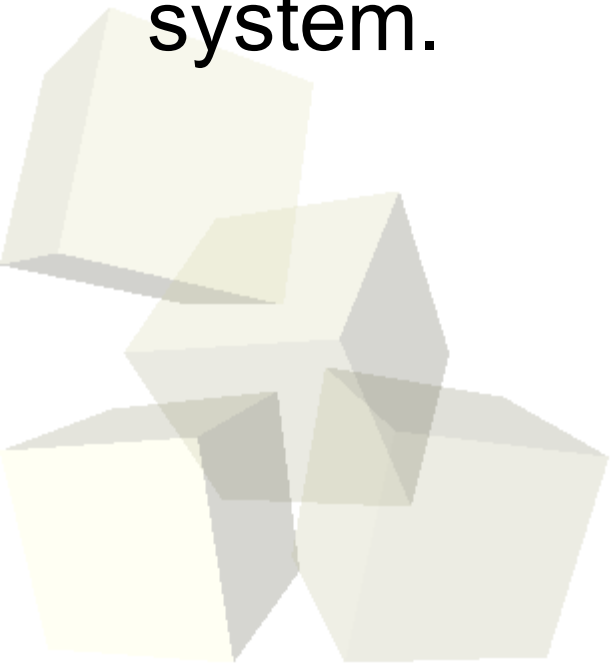
Use Case Diagrams

- These are diagrams that are used for the analysis portion of software development.
- This is the time when you are asking, “What should this software do?”
- When building use case diagrams you are basically determining all the different ways in which users will use your software, how they will interact with it.



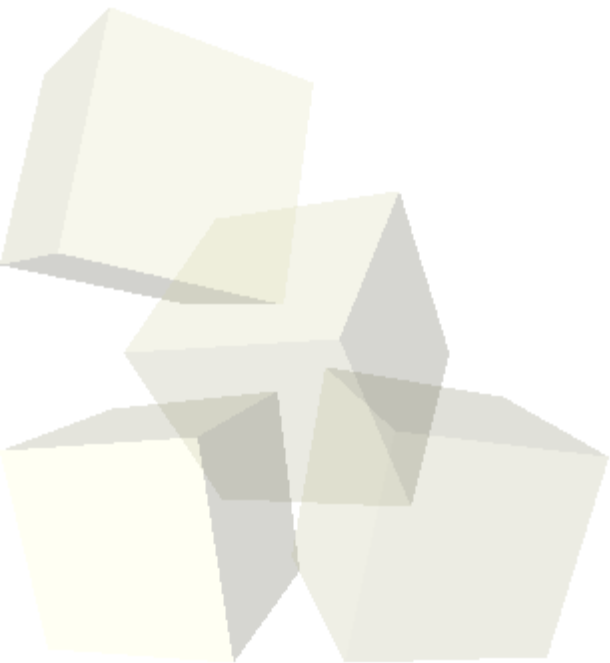


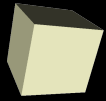
- The first element in a Use Case diagram is an actor.
- They are drawn as stick figures in your diagram.
- Often actors represent people.
- More generally, an actor represents something that interacts with your system, but that you don't have control over so it isn't really part of your system.



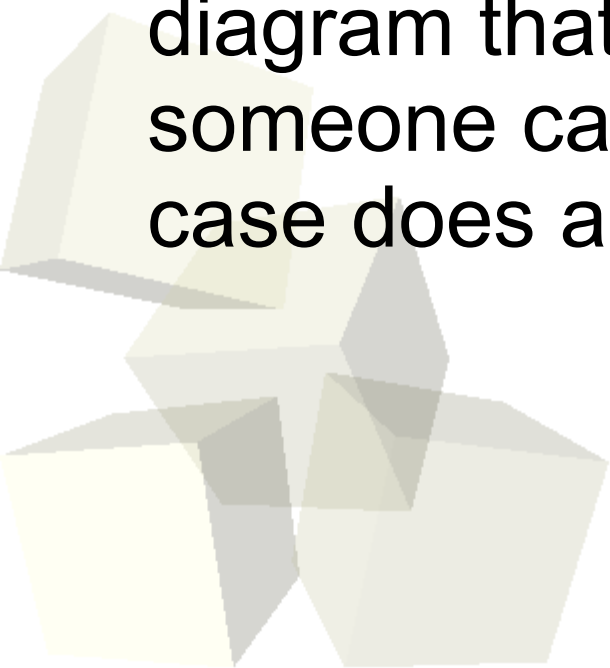


- The same type of arrow that gives you inheritance in class diagrams gives you generalization in Use Case diagrams.
- You can generalize actors to say that one actor is a subtype of another actor.
- You can also generalize use cases.



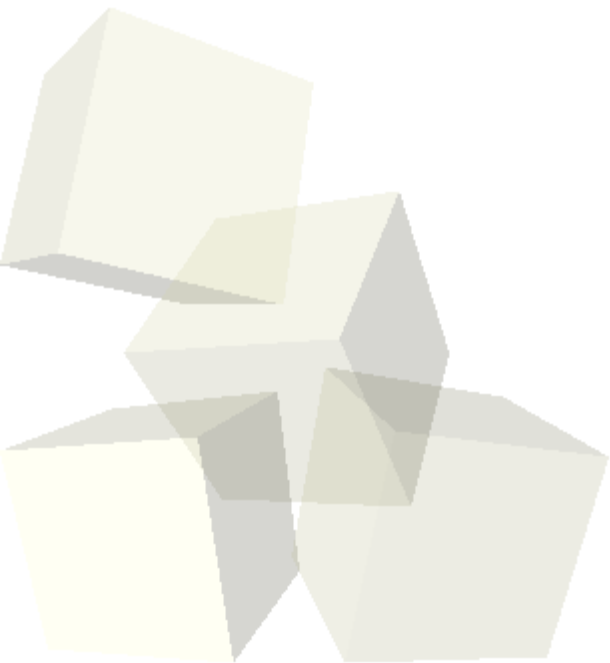


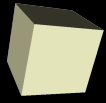
- These are the principle elements of a use case diagram.
- They are drawn as bubbles with a brief description in them.
- This simply represents something that your system does.
- You should include text associated with the diagram that provides enough description that someone can figure out in detail what the use case does and what requirements there are on it.





- Most UML tools allow you to draw boxes in Use Case diagrams that represents systems. Your use case bubbles generally go inside of a system.
- The tool in NetBeans doesn't appear to have this option.
- The actors should be outside of the system.





- An includes relationship between use cases tells you that one use case will always include the behavior of another use case.
- This is a dotted arrow with an open head annotated with << includes >>.





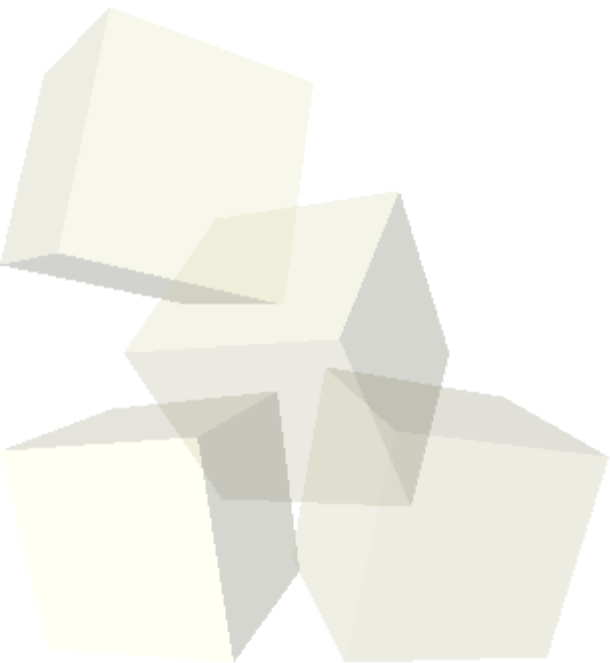
- There is also an extends relationship. This is like include only the arrow goes the other direction and here the use case doesn't always have to be part of the behavior of another use case.
- Extends is one of the lesser used features of use case diagrams. When you want to indicate that one use case is a more specific version of another, use generalization.





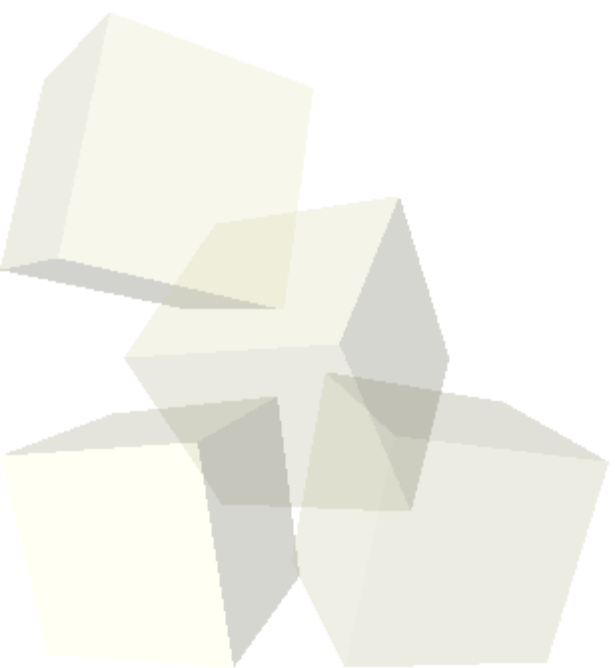
Activity Diagrams

- These diagrams communicate the work flow through a project.
- This is a design diagram. You use it after you are done with analysis and you know what you want to do. This diagram helps you figure out the steps that will be followed to get it done.



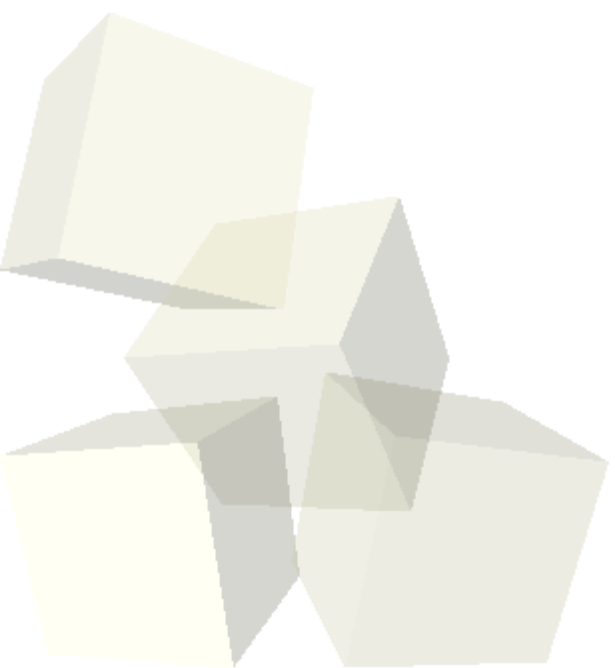


- All Activity Diagrams will have two nodes.
- The initial node is a solid dot while the final node has a circle around the solid dot.





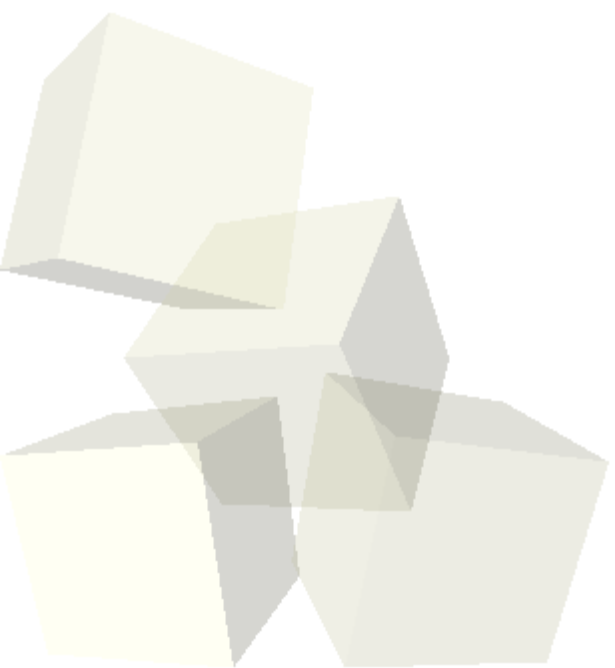
- See NetBeans

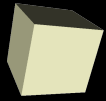




Decisions and Merges

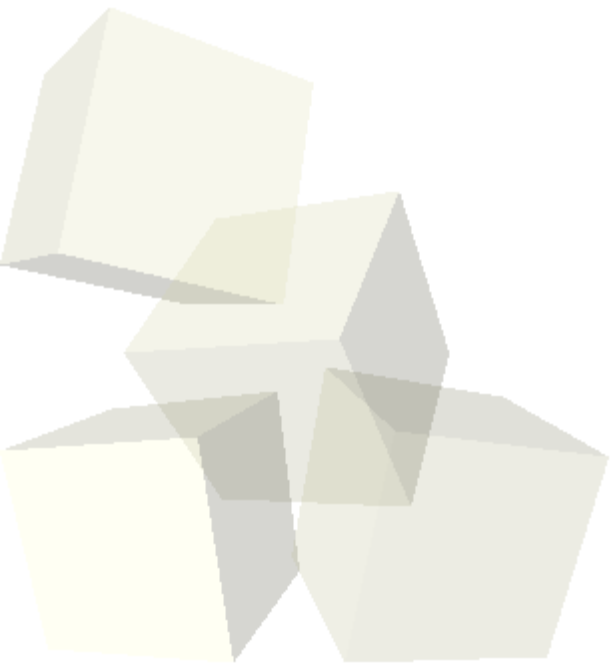
- See NetBeans





Multiple Threads

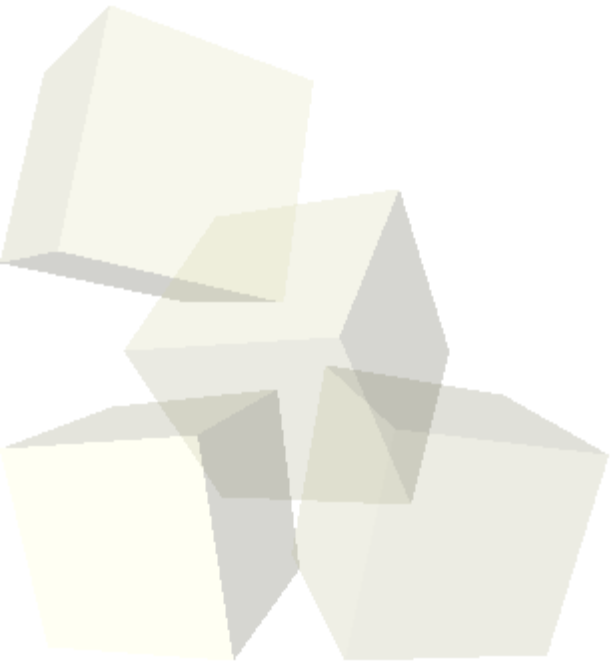
- See NetBeans





Calling Other Activities

- See NetBeans

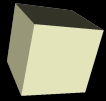




Action Inputs and Outputs

- See NetBeans





Object State Changes

- See NetBeans

