Let's look at some answers to the interclass problem.

**Answers to questions on earlier minute essays.**
- Is the learning hands on? As much as I can make it.
- Is there required lab time? No. Open labs are optional.
- Does Alice ever get more complex than drag and drop? No. That's why we will move to Java.
- How will you use Java in life outside class? Unless you switch to a CS major you probably won't. What you will use is the ability to decompose problems and structure solutions. You might even find that writing a bit of code can occasionally be helpful.
- Are there labs with Alice that are open 24 hours? These labs are, unfortunately, not open 24 hours. Other labs won't have Alice.
• Can you create characters in Alice? Yes, but I don't expect you to. From what I've heard it is a fairly tricky process, but alice.org can help you if you are serious about doing it.
• What is programming? Writing a set of instructions for a computer to make it do something.
• Will I be playing noonball this semester? Yes.
• Will quizzes be announced? Yes. They are now on the schedule page.
• Will tests be cumulative? Not explicitly.
• Can I recommend other links or references to help on the projects? For Alice, alice.org is a good starting place. The links page has some Java stuff, but more might be added.
• Do you need the book for every class? No. You probably won't really NEED the book for any classes.
• When do we move to Java? October 15th.
• Do you need a lot of computer experience to be successful in this class? I sure hope not.
• Does living off campus make things harder? For this class it shouldn't as long as you mail yourself interclass problems or bring them on a thumb drive.
Let's go and add some more to the code that we started working on last time.
Your book has a section on software engineering that they present with a strong Alice twist.

- Write your story.
- Draw storyboards.
- Build scenes. Make an object for each noun in the story.
- Put together the code and put in method calls for all the verbs in the story.

This view of software engineering is a bit too simplistic. Here are the standard steps.

- Analysis – what is the problem?
- Design – how will you solve it?
- Implement – write the solution.
- Debug – fix what you messed up.
- Deploy and maintain
So far we have had all of our statements in Alice happen sequentially. This is the normal mode of working in programs.

Alice provides a simple way to make things happen at the same time. The “do together” block lets you specify a bunch of commands that should all happen at the same time.

Let's make use of this and see what happens.
When you are working on the project you will probably be picky about exactly how different figures are set up.

The quad view gives you a lot more information about the relative position of objects in your scene.
What makes do together potentially difficult to use? As it happens, most all things that can happen in parallel on computers can have interesting complications. This is a major area of growth in CS at the current time.

Interclass Problem – Do either problem 1.6, 1.7, or 1.8 from the text.