Slide 1

Administrivia

- One purpose of the syllabus is to spell out policies, especially about:
  - Course requirements and grading.
  - Exam dates.
  - Late work.
  - Academic integrity.
- Most other information will be on the Web, either on my home page (here, office hours) or the course Web page (here).

  A request: If you spot something wrong with course material on the Web, please let me know!

Slide 2

More Administrivia

- Part of my job is to answer your questions outside class, so if you need help, please ask! in person or by e-mail or phone.
- Some of my office hours are designated as “open lab”. At those times I will be in one of the classrooms/labs ready to answer questions.

Slide 3

More Administrivia

- Probably the easiest (though not the only) option for doing the assignments is to use the Linux lab machines.
- You should have physical access (via your TigerCard) to four rooms containing such machines any time the building is open. You should have remote access to any that are booted into Linux.
- Returning students should already have accounts set up. (If you've forgotten your password, go to the ITS help desk and ask for it to be reset.) Accounts have been set up for new students. Username is the same as your Windows/ITS username; password has been mailed to you. To change it, open a terminal window and type `yppasswd`.

Slide 4

What Is This Course About?

- Improve programming skills.
- Understand "object-oriented" paradigm.
- Learn (more) basic concepts — data structures, etc.
- Along the way — learn Java, use IDE.

Slide 5

About the Readings

- The textbook (*Java Software Structures*) is intended for a second-semester programming course, for people who already know Java.
- Since we teach our first-semester course in C — Dr. Lewis has been writing a "From C to Java" document.
- You should probably skim all assigned reading, but those with Java background will find some parts review.

Slide 6

"Object Orientation"?

- A "programming paradigm" — contrast with procedural programming, functional programming, etc.
- No accepted-by-all definition, but most definitions mention encapsulation:
  - Data and functionality grouped together into "objects".
  - Some data/functionality is hidden.
- Origins in simulation/modeling, where the goal is to model complex systems consisting of many (real-world) objects.
- (More about this next week.)

Slide 7

The Course Programming Project

- Write an arcade-style game.
- Build on “game infrastructure” (a.k.a. “Lewis Magic Cloud”).
- Project goes a step at a time, with first steps fairly easy and a lot of flexibility.

Slide 8

Game Basics

- “Player” — human-controlled moving entity.
- “Screens” — two-dimensional grids, make up playing field, side view or top view, can be linked together.
- “Blocks” — components of “screen” grids.
- “Game entities” — program-controlled entities, stationary or moving.

Slide 9

To Do for Next Class

- Start reading the project description (on the Web, linked from "Useful links").
- Think about what kind of game you want to write.

Slide 10

Minute Essay

- Tell me about your background:

  If you took CSCI 1320 at Trinity, when and with what professor?

  If not, what programming classes have you taken (high school or other), and what language(s) did you use?

  Have you had any exposure to Java?

  Have you had any exposure to a Linux/UNIX command-line interface?

- What are your goals for this course? Anything else you want to tell me?