Slide 1

Administrivia

- Reminder: Quiz 2 Thursday. Likely topics are inheritance and interfaces.
- Reminder: Homework 2 design due today at 11:59pm. Okay to turn in tomorrow without penalty. Discussion today.

Slide 2

Administrivia

- *Please do not* reboot the machines in HAS 340! People use these remotely, and you may cause someone's program to crash. If you think a reboot is needed, ask a faculty member.
- Some words about space usage on Sol (your home directory):

  Disk space on Sol is finite, so space per user is limited by a quota. Can be increased if necessary, but try not to. Bad/strange things happen when you go over quota.

  `quota` to check. `du --maxdepth=1` to check which directories are using a lot of space.

  If you don't know what something is, ask before deleting it.

Slide 3

Homework 2 — General Comments

- Design phase is meant to be about defining classes and interfaces. For every class (or interface) and every method, I want comments (can be be brief). For classes, these should describe (to the best of your understanding) how they fit into your game (e.g., "class for wall blocks").
- In order to generate the HTML documentation ("javadoc"), probably have to have something minimally compilable. As suggested in assignment — create skeleton/stub versions of methods, and fill in real code in code phase.
- Be sure to get the updated JAR file (should have name `PAD2F05Assn2.jar` — link from Web page was wrong until yesterday afternoon). With every assignment there will be a new JAR file, as you replace various parts of the starter code with your code.
- Define your own package rather than putting your classes in `edu.trinity.cs.gamecore`.

Slide 4

Homework 2 — Design

- Interfaces `YourBlock`, `YourEntity`: In project API, referred to as "general block type" and "general entity type". You will use these as replacements for `BasicBlock` and `BasicEntity`, and everywhere else you use one of the framework's generic classes.
- Player and game setup classes. Copy code from `BasicPlayer` and `BasicGameSetup` and edit (change `package` line, block and entity types). May want to change game setup more during code phase. Also edit your main class from the first assignment.

  Don't worry about player for now — you will start writing your own in the next assignment.

Slide 5

Homework 2 — Design Continued

- Block class(es). These are blocks that make the playing field for your game. Should have one class for each kind of block (floor, walls, ladders, anything that doesn't move). Try to define as many as you can. Copy code from `BasicBlock`.
- Screen class (class implementing `Screen` interface). This is the most work in this assignment. Eclipse can make stub methods for you. Copy and paste comments from API.

Slide 6

Homework 2 — Code

- Go back through copied code and look for things you know you want to change. E.g., you should be able to figure out how to draw something okay-for-now for your game's blocks by changing `getImage`.
- For your screen class, decide what variables you need to implement the methods in `Screen`. (You'll probably want a two-dimensional grid (array) of blocks (your block type) and a list of entities (your entity type).)

Slide 7

"Good Style" Tips

- Make methods public if needed by code that uses your class, private otherwise.
- Make variables private unless there's a good reason not to — prevents unwanted/inconsistent access.
- Use named constants (static final variables) rather than hard-coded values.

  E.g., `private static final screenWidth = 20;`

  Also remember that you can get the size of an array from its `length` field (variable).
- Follow Java conventions — class names start with a capital latter, method and variable names with lower case.

Slide 8

Multidimensional Arrays

- "Arrays of arrays", e.g.,

  ```
  int[][] x = new int[10][100];
  ```
- For 2D arrays, first index is row, second is column.

  (Note, though, that this is not the "graphics convention" used in the game.)

Slide 9

Minute Essay

- Write code to define an array of four `String`s and fill it with data of your choice.
- Write code to define a two-by-three array of `int` and set each element to the sum of its row and column.

Slide 10

Minute Essay Answer

- One solution (array of `String`s):

```
String[] s = new String[4];
s[0] = "hello";
/* other three lines similar */
```

- One solution (array of `int`s):

```
int[][] a = new int[2][3];
for (int row = 0; row < a.length; ++row)
    for (int col = 0; col < a[0].length; ++col)
        a[row][col] = row + col;
```