



Java2D Graphics

3/8/2007





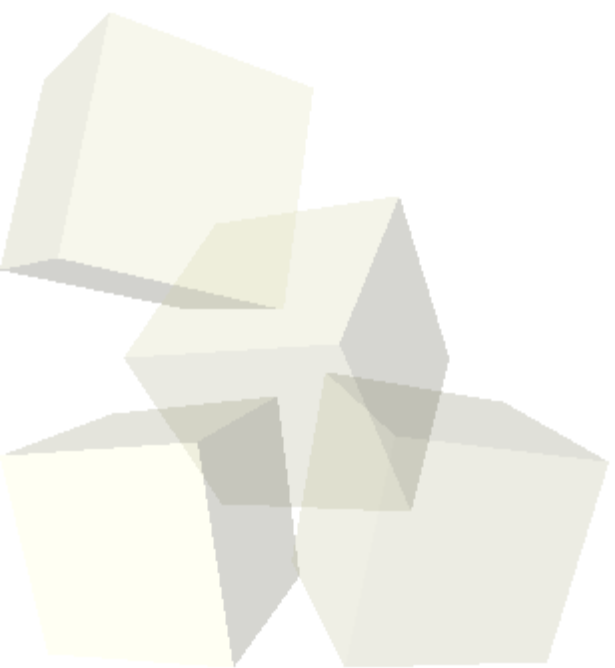
Opening Discussion

- Do you have any questions about the assignment?
- Do you have any questions about the reading?
- How do you do custom drawings in Java?
- What are some of the capabilities of Java2D?
How do you use those capabilities?





Midterm Results





- I want you to spend a few minutes writing a little program with a main in it that draws a little custom picture. The picture should include both shapes and text.
- Now make it so that the picture rotates. You can make the rotation happen either because of user input or based on a timer.





- In C if a function ran into a problem it either had to return an invalid value, or it had to set a flag. Both of these options leave the burden on the programmer to check things and often lead to propagation of errors, making them harder to track down.
- In Java, problems in code should be reported with exceptions. We have already seen the use of try/catch blocks when we call code that might throw an exception.

```
try {  
} catch(Exception1 e) {  
} catch(Exception2 e) {  
} finally { // This isn't required  
}
```



Types of Exceptions

- There are basically two types of exceptions in Java, checked and unchecked. (There are also errors, but when those happen it typically means you are sunk and can't do anything about it.)
- With a checked exception you are forced to either have it in a try/catch or your function method must say it throws that exception by using the throws syntax.
- Unchecked exceptions are any exceptions that inherit from RuntimeException, and they do not have to be dealt with. Without these your code would become unwieldy because almost every line can throw a NullPointerException.



Throwing Your Own

- If you write a function that might not work, but can't handle the problem itself, then your function should throw an exception.
- We throw an exception with the throw operator. It is followed by an object of type Throwable (normally an exception of some type).
- The method should have a throws clause listing the different exceptions it can throw. This is required for checked exceptions.





Self-documenting Exceptions

- Good exceptions tell you why they happened and give you sufficient information to help you debug.
- For example, something like an `IndexOutOfBoundsException` should tell you what index was asked for and the range that was valid so that you don't have to rerun the program and print that information before the point where it crashes.





- The process of refactoring is fairly new in the software world, at least it has only been recently formalized. You refactor your code when you change the structure without changing the function. A proper refactoring will not change what a program does in any way at all.
- You normally refactor code when you see something that can be improved. A simple example is changing the name of a method or variable so it better fits the usage.
- More complex refactorings include break up methods, pulling code out into methods, pulling code out into classes, etc.



- What are you going to be doing for Spring Break?
- I'm going to be adding two more sections to "From C to Java". I'll send out an e-mail when they are ready for you to read.
- Try not to forget too much over the break.
Remember that this is a creative venture and you need to exercise your abilities to grow them and also so they don't atrophy.

