Off topic

- Barbie is becoming a computer engineer
- Google is visiting tomorrow (Wednesday)
- CS career faire Thursday
CMSC 433
Programming Language Paradigms and Technologies
William Pugh
Feb 16th, 2010
Snow Updates

• Project 2 due today
• Dropping Web security as a project
  • material will be covered in exercise/quiz
• Next project due Thursday, Feb 25th
Today

• Spend a little bit of time introducing you to the next project

• Give you stuff to get started working on for the next project

• 30+ minutes of tutorial videos online

• Leave some time for discussions for those still struggling with project 2
Next project

- Play with Picture package
- Learn and understand that your eyes such at blue
- Understand how to use reflection to construct objects
- Understand how to build grammars using ANTLR
Picture class

• Some code originally developed for CMSC 131

interface Picture {

    PictureColor getColor(int x, int y);
    int getWidth();
    int getHeight();
}


PictureColor

- Red, Green and Blue components
  - each 0.0 - 1.0

- Could have used standard Color class, but didn't want to have to explain the 0-255 range in CMSC131

- No need to switch to standard Color implementation for our purposes
Decorator design pattern

- Given an interface (or abstract class)
- A Decorator both implements that interface and is constructed around something else that implements that interface
Decorators delegate

- Generally, decorators delegate method calls to the wrapped object
- they may intercept/modify some or all of the calls
- and perhaps provide additional functionality
Decorators in Java

• Many decorators for InputStream, OutputStream, Reader and Writer

• Collections.unmodifiable\(X(X)\) provide decorators for all the collection classes

  • e.g., Set\(<T>\) unmodifiableSet(Set\(<?\) extends T\(>\) s)

• Also Collections.checked\(X(X, Class\ c)\)
Picture decorators

• For the picture class, we have many classes that are Picture decorators

• Slightly abusing the concept, we have some classes that decorate 2 or 3 wrapped Pictures

• e.g., CombineLeftRight(left, right) or Average(first, second)
Project 3
builder methods

• public static Picture build(String pictureClassName, Object... arguments) { … }

• pictureClassName is either the simple name of a class in the cmsc433.p3.picture class, or a full class name

• arguments is a sequence of Pictures, Integers and Strings

• find the appropriate constructor and invoke it
Finding a class

- Given name, Class.forName(name) returns a class matching that name, or throws an exception if none is found.
- Use Class.forName(name).asSubclass(Picture.class) to get a Class<? extends Picture>.
- No compiler warnings in construction code.
Finding a constructor

• We only consider constructors that take Picture, int and String arguments
• Use getConstructor method to get a constructor
  • Note: when looking for int arguments, use Integer.TYPE rather than Integer.class
• invoke the newInstance method on the constructor, passing the actual arguments
Picture interpreter

• public Picture buildPicture(String desc) throws RecognitionException { … }

• Given the String Green(Testudo())

• returns the same thing as new Green(new Testudo())
Picture memory

- Picture interpreter interprets statements
- A statement can either be an expression, or an assignment of a expression to an identifier
- for an assignment, update an internal Map<String,Picture>, and return null
An expression can be

• **ID** - a picture stored in the Map
• **ID '(' expList ')'** - construct an instance
• **INT** - an integer value
• **STRING** - a string value
Some implementation notes

• Throw some kind of exception if a string can't be parsed, if an appropriate constructor can't be found, or if an identifier isn't defined in the map.

• If you get stuck on parsing/interpreting variable length expList, we won't be testing more than 3 arguments. You can explicitly handle 0, 1, 2 or 3 arguments.
Parsing

• Recommend, but not require, use of ANTLR
• http://www.antlr.org/
• http://antlrv3ide.sourceforge.net/
• http://javadude.com/articles/antlr3xtut/
• watch concepts, prologues 1 & 2, parts 1-3
Project setup

- Check code out of projects/pictureParser in your CVS repository
- We will be using submit server and release tests for this project
  - there will be secret tests as well as release tests
New plugins / downloads

• New to update Course project manager
• Generated files aren't under CVS control
• previous version of course project manager could only submit files under CVS control
• Need to download ANTLR, additional plugins
• all discussed in video, prologue #1
Your Eyes Suck at Blue

- Original, Pixelated
- Pixelated Red, Green, Blue
Rods and Cones

- [http://hyperphysics.phy-astr.gsu.edu/hbase/vision/rodcone.html](http://hyperphysics.phy-astr.gsu.edu/hbase/vision/rodcone.html)

- You have 3 types of Cones: red receptors, green receptors, and blue receptors

- 6-7 million cones
  - 64% red
  - 32% green
  - 2% blue
Blue cones

- Highest sensitivity per cone
- Mostly found outside the fovea centralis, where we get our high detail vision
- Also, refractive index for blue is different enough that when red and green are in focus, blue is slightly out of focus
Fovea Centralis