CPS 108

● Reminder of PRO reasons to take the course (including)
  ➤ Programming, project, prolific, proficient, prodigious, process, procrastination, profound

● Some CON reasons
  ➤ Conceptual, condemnable, confusing, condensed, conscientious, contemptuous, contented

● In a nutshell
  ➤ Object oriented design and programming of medium-sized projects in groups using modern tools and practices in meaningful ways
Something new

● In the past, CPS 108 covered C++, then Java. In Spring 2001 the first project was based on using both at the same time
  ➤ In theory this was supposed to illustrate both how related and unrelated the languages are, allowing more time for both
  ➤ In practice C++ was slighted and students didn’t do Java initially (with no real penalty, so why worry). Once we started with Java, C++ wasn’t used

● Consequence: too long for Proofread because of Java issues, cut into subsequent projects, left little (no) time for smaller, mastery projects
  ➤ In the future: C++ first, then Java
Proofread

● Intended to be simple enough so that C++ and Java versions could be developed simultaneously
  ➤ Wasn’t big enough (intentionally) to require a big group
  ➤ Size of problem made single-person development optimal, lots of people didn’t need to learn Java

● Consequences
  ➤ Too long spent on this assignment
  ➤ No smaller, mastery assignments, no low-level C assignment
  ➤ Ramp up time in Java needed on subsequent assignment

● Don’t do C++ and Java together, at the beginning of the course
Architectural issues important, leverage a good system-level architecture into easier-to-develop and tune programs

- Ambitious groups could get away without an architecture, but a good architecture really helped ease development
- Lack of Java expertise (and other issues?) prolonged the assignment, what about on-time submissions?

Original puzzle code did more harm than good despite warnings that it wasn’t a completely good design

- Lack of networked joggle was lamentable
- Dots should have required a computer-player
- Card games would also be architecturally interesting
  - Solitaire? Other games?
Snews, final project

- Snews: a real program (albeit one that shouldn’t be written:-)
  - Requires reading RFCs, understanding a simple network protocol, using several standard but not universally used APIs
  - Opportunity for tool development, intergroup collaborations
  - Somehow interesting, but you’re the ultimate judge on this
- No need for lots of GUI stuff (one complicated widget?)

- Mandatory intra-group switching didn’t seem warranted
  - Some smaller groups disadvantaged
  - Still probably the right thing to do
  - Intra-group collaboration still possible
Slogo/Programming/Parsing

- Parsing and interpreting are important topics everyone should know about, CPS 140 isn’t required and doesn’t involve programming (typically)
  - Before and after (patterns/OO) Elan example useful?
  - Go over JavaCC for parser generation?

- Lots of bells-whistles in the language and the IDE, this is a good thing for projects
  - Allows less-experienced groups to accomplish a lot
  - Allows wizards to demonstrate wizardry

- More GUI issues than SNews
Post-mortem evaluation

- Need smaller programs first (Masteries are a good idea)
  - C++ inheritance, templates, low-level C, sockets
  - Java GUI, threads, web, network
- Success: OOGA, Final projects (design and implementation)
- Failure: Proofread and simultaneous C++/Java
- Success: Set your own standards, don’t meet my requirements
- Failure: Reading about software, aesthetics, ethics, …
- Failure: feedback
- Success: coding examples
- Failure: MVC for everyone
- Success: MVC for many
- Failure: Lower-level coding standards
- Overall: ______

CPS 108
Why is programming fun?

What delights may its practitioner expect as a reward?

First is the sheer joy of making things

Second is the pleasure of making things that are useful

Third is the fascination of fashioning complex puzzle-like objects of interlocking moving parts

Fourth is the joy of always learning

Finally, there is the delight of working in such a tractable medium. The programmer, like the poet, works only slightly removed from pure thought-stuff.

Fred Brooks