Programming Languages and Techniques (CIS120)

Welcome

Introductions

- Steve Zdancewic*
  - stevez@cis.upenn.edu
  - http://www.cis.upenn.edu/~stevez/
  - Office hours: Mondays 2:30 – 4:00pm (& by appointment)

- Swapneel Sheth
  - swapneel@cis.upenn.edu
  - http://www.cis.upenn.edu/~swapneel
  - Office hours: Tues. 10:30am – 12:30pm (& by appointment)

*Pronounced phonetically as: "zuh dans wick". I won't get upset if you mispronounce my name (really!). I will answer to anything remotely close, or, you can call me just Professor, or Professor Z. Whatever you feel comfortable with.

Head Teaching Assistants

Nicolas Corona

Gabrielle Hemlick

Hillary Aristotle

Will Goeller

What is CIS 120?

- CIS 120 is a course in **program design**
- Practical skills
  - ability to write larger (~1000 lines) programs
  - increased independence ("working without a recipe")
  - test-driven development, principled debugging
- Conceptual foundations
  - common data structures and algorithms
  - several different programming idioms
  - focus on modularity and compositionality
  - derived from first principles throughout
- It will be fun!

Prerequisites

- We assume you can already write small (10- to 100-line) programs in some imperative or object-oriented language
  - Java experience is *strongly recommended*
  - CIS 110 or AP CS is typical
  - You should be familiar with editing code and running programs in some language
- CIS 110 is an alternative to this course
  - See: https://advising.cis.upenn.edu/cis110
  - If you have doubts, ask us

CIS 120 Tools

- OCaml
  - Industrial-strength, statically-typed *functional* programming language
  - Lightweight, approachable setting for learning about program design
  - Web based development: codio.com
- Java
  - Industrial-strength, statically-typed *object-oriented* language
  - Many tools/libraries/resources available
  - Develop using Codio or Eclipse

Why two languages??

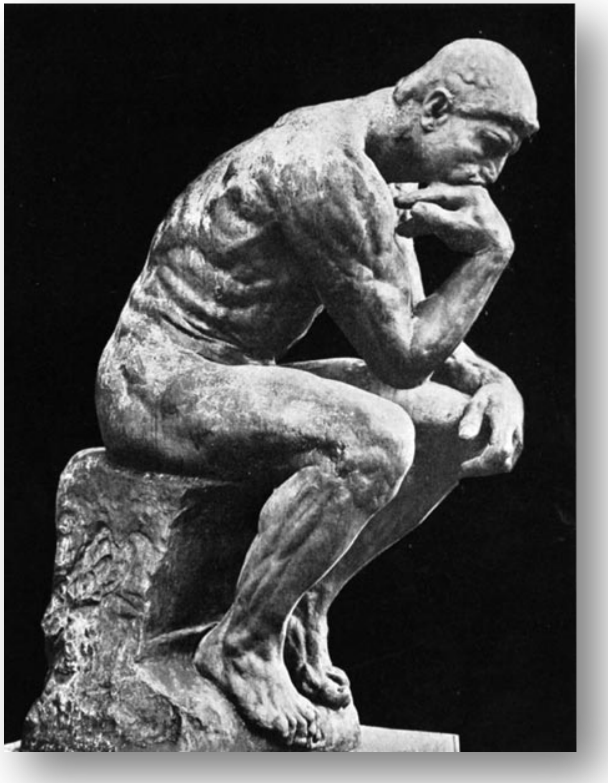
- Clean pedagogical progression
- Everyone starts at the same place
- Practice in learning new tools
- Different perspectives on programming

“[The OCaml part of the class] was very essential to getting fundamental ideas of comp sci across. Without the second language it is easy to fall into routine and syntax lock where you don't really understand the bigger picture.''
---Anonymous CIS 120 Student

“[OCaml] made me better understand features of Java that seemed innate to programming, which were merely abstractions and assumptions that Java made. It made me a better Java programmer.''
--- Anonymous CIS 120 Student

Philosophy

- Introductory computer science
  - Start with basic skills of "algorithmic thinking" (AP/110)
  - Develop systematic design and analysis skills in the context of larger and more challenging problems (120)
  - Practice with industrial-strength tools and design processes (120, 121, and beyond)
- Role of CIS120 and *program design*
  - Start with foundations of programming using the elegant design and precise semantics of the OCaml language
  - Transition (back) to Java *after* setting up the context needed to understand why Java and OO programming are useful tools
  - Give a taste of the breadth and depth of CS



Course Structure and Logistics

All course material is availble on the course website
Please read the syllabus today

http://www.seas.upenn.edu/~cis120/

Course Components

- Content Quizzes (2% of final grade)
  - Presentation of ideas and concepts, interactive demos, etc
  - Lecture notes & screencasts available
  - Grade based on participation in weekly quizzes
- Recitations / Labs (6% of final grade)
  - Practice and discussion in small group setting
  - Grade based on participation
- Homework (57% of final grade)
  - Practice with individual problem solving
  - Help available from course staff
  - Grade based on automated tests + style
- Exams (35% of final grade)
  - Test foundations of program design
  - Do you understand the terminology? Can you reason about programs? Can you synthesize solutions?
  - 2 midterms (10% each) and a final (15%)

Warning: This is a *challenging* and *time consuming* (and rewarding :-)) course!

Asynchronous Course Content

- Content videos
  - edited versions of Fall 2019 lecture recordings
  - "deep dive" videos created for Fall 2020
- Course lecture slides
  - Complete pdf of all slides presented in the videos
- Comprehensive lecture notes
  - ~400 pages CIS120-tailored material

Each Friday, we'll post the following week's course content.

You follow along and complete weekly content quizzes to mark your progress.
- 2% of overall grade
- graded on completion only (not correctness)
- link on Piazza
- first quiz due on Sept. 11th

(Optional) Synchronous Content

- No synchronous lectures
- Instead: interactive demos, Q&A, discussion
  - two zoom meetings MW 11am-noon, noon-1 ET
  - bring questions, ask "live" on Piazza
  - chat with faculty and other students
  - interact with the instructors

Come prepared to engage with the instructors and the course material!

- Will work best if you've watched (some of) the week's videos ahead of time
- These sessions will be recorded for later viewing
  - it takes 1-2 hours to process the videos for Canvas

- (most) Fridays: TA-led "watch parties"
  - synchronous, group chats, etc.
  - start on Fri. 9/11
  - this week: codio demo

Piazza

- We use Piazza for most communication in this course
  - from us to you
  - from you to us
  - from you to each other
- If you are already registered for the course, you should have been signed up automatically
  - If not, please sign up at piazza.com
- Piazza supports *anonymous* questions
- We'll use its "live" questions features during the synchronous interactive sessions

Look to Piazza for course content announcements, weekly "todo" lists, reminders, etc.

Recitations / Lab Sections

- Synchronous Recitations start *next week (Sept. 7th)*
  - Time scheduled by the registrar
  - Try Codio before the first meeting
- Goals of first recitation
  - Meet your TAs and classmates
  - Practice with OCaml before your first homework is due
- Asynchronous Option:
  - worksheet graded by effort / completion (not correctness)
  - due Thursday 23:59 AoE (Anywhere on Earth, UTC-12h)
  - send mail to your TAs

Homework

- 9 programming assignments
- Submit assignment on the course website
  - You'll get automated grade and style feedback
  - Each assignment will have limits on the number of submission attempts
- Due at midnight (23:59pm ET) on the date announced
- Late penalty
  -10 points if up to 24 hours late.
  -20 points if 24-48 hours late.
  no submissions accepted after that

Academic Integrity

- Submitted homework must be *your individual work*

- Not OK:
  - – Copying or otherwise looking at someone else’s code
  - – Sharing your code in any way (copy-paste, github, paper and pencil, …)
  - – Using code from a previous semester
- OK (and encouraged!):
  - – Discussions of concepts
  - – Discussion of debugging strategies
  - – Verbally sharing experience

Penn’s code of academic integrity:

http://www.upenn.edu/academicintegrity/ai_codeofacademicintegrity.html

Enforcement

- Course staff *will* check for copying
  - We use plagiarism detection tools on your code

Violations will be treated seriously!

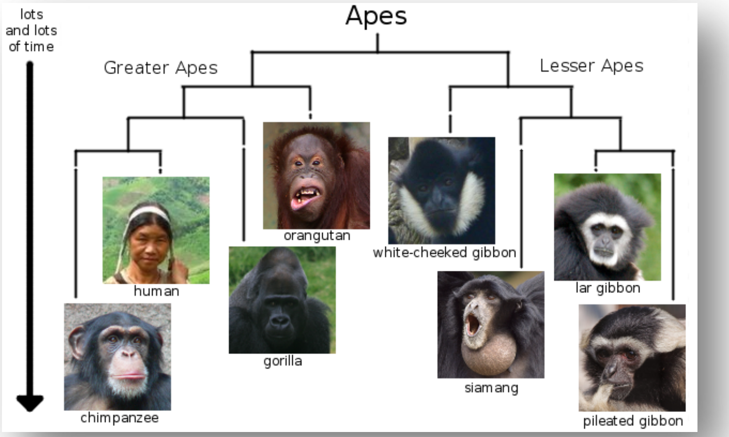
- *Questions? See the course FAQ. If in doubt, ask.*

Codio

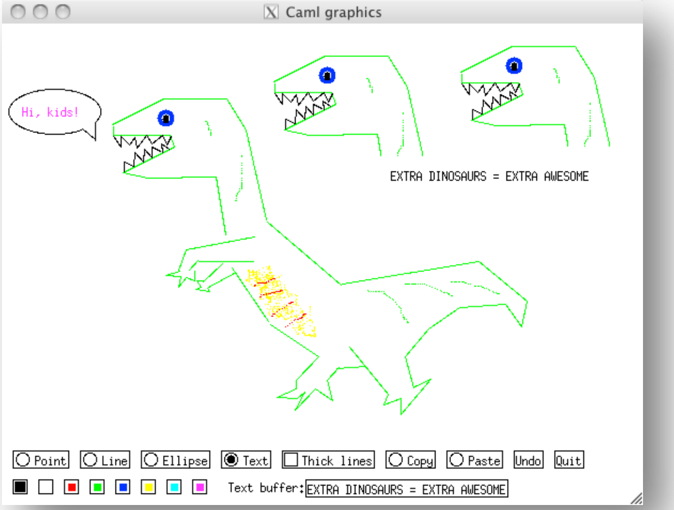
- For OCaml programming, we use Codio
- codio.com
  - web-based development environment
  - see Piazza for setup info
  - see video intro / interactive sessions for demos

- Under the hood:
  - linux virtual machine (Ubuntu)
  - pre-configured per project with everything you need

Some of the homework assignments…



Computing with DNA



EXTRA DINOSAURS = EXTRA AWESOME

Text buffer: EXTRA DINOSAURS = EXTRA AWESOME

Building a GUI Framework

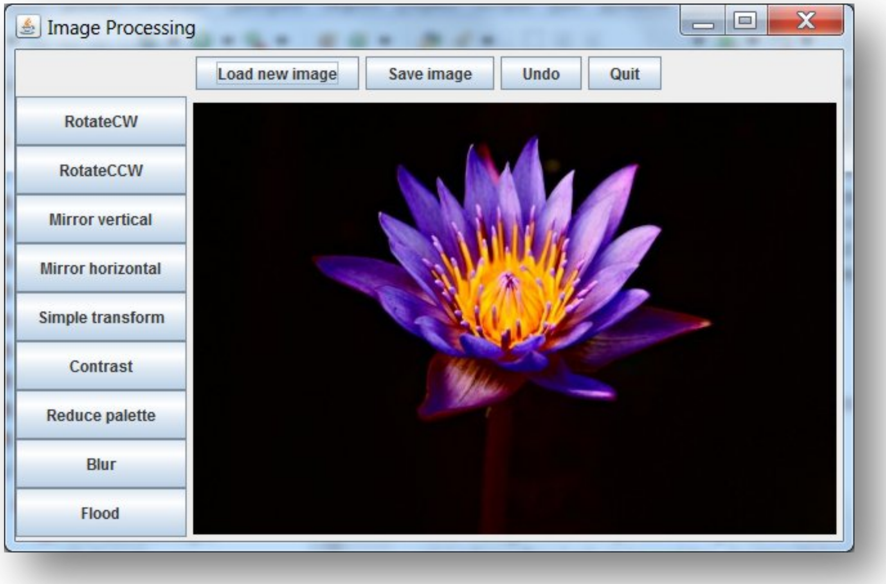
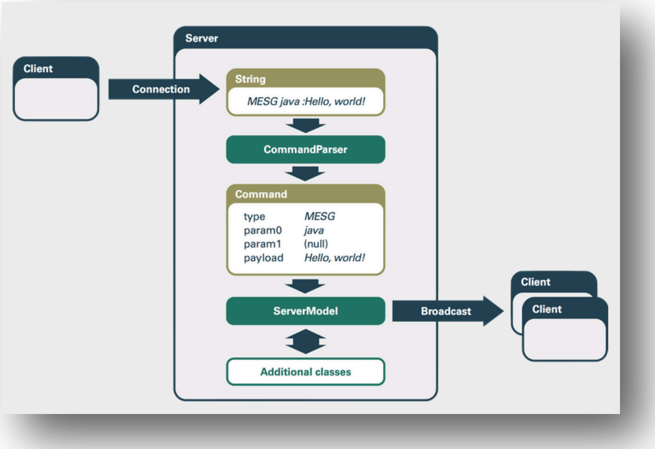
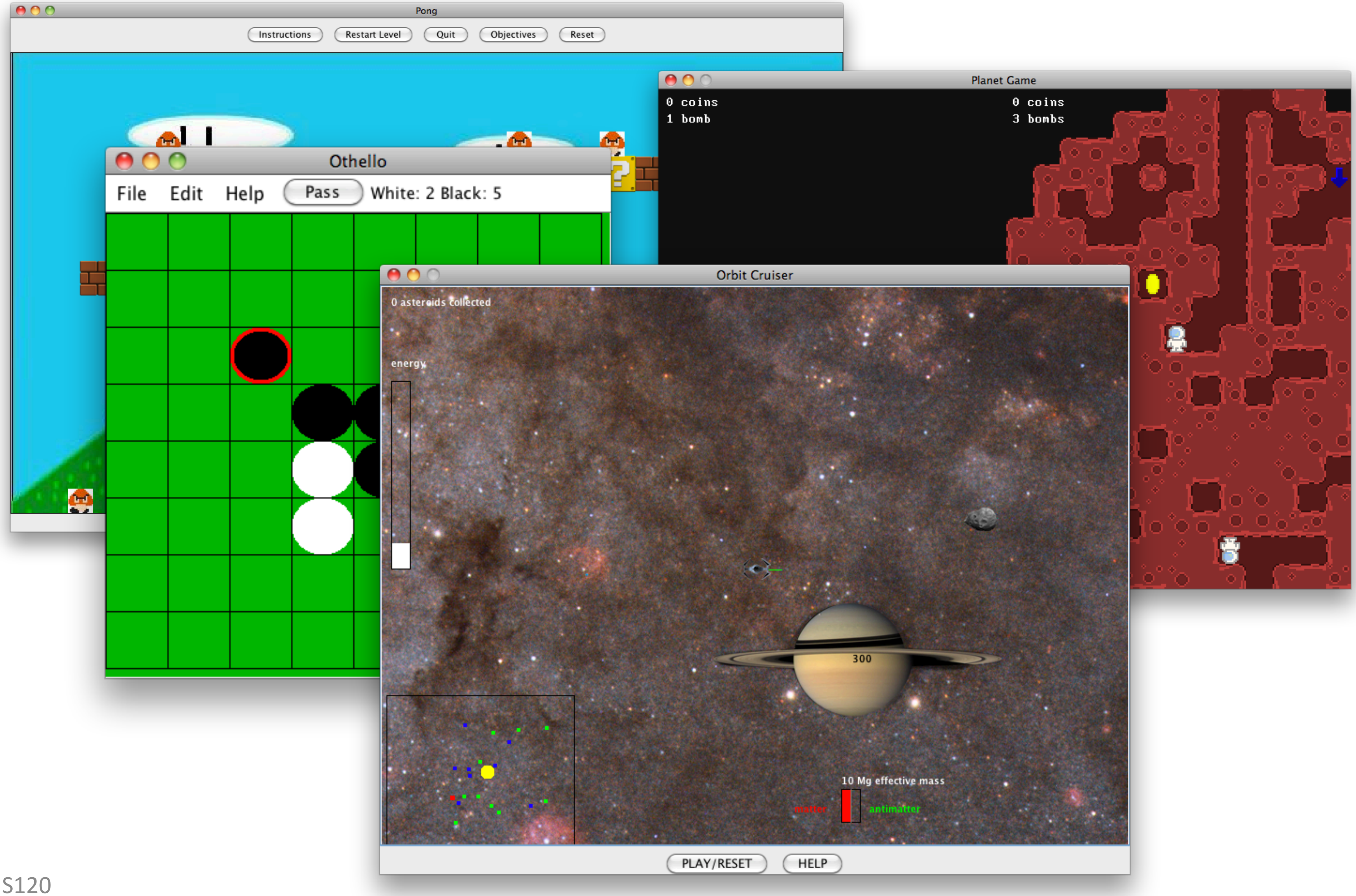


Image Processing

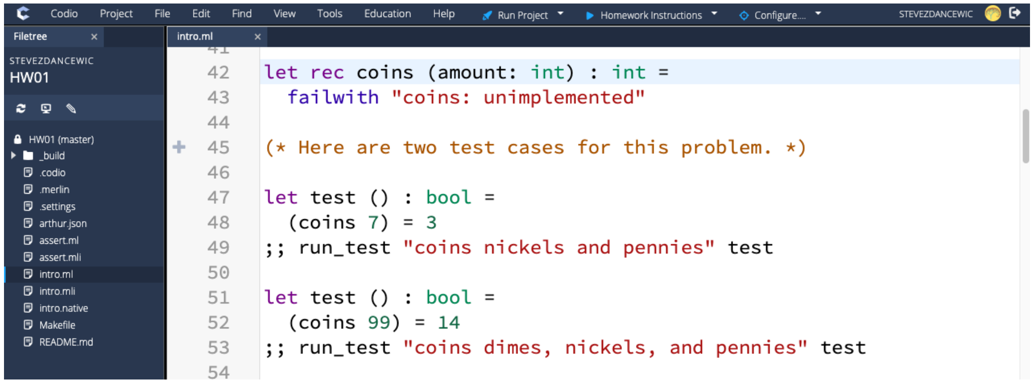


Chat Client/Server

Final project: Design a Game



HW01: OCaml Finger Exercises



```
let rec coins (amount: int) : int =
  failwith "coins: unimplemented"

(* Here are two test cases for this problem. *)

let test () : bool =
  (coins 7) = 3
;; run_test "coins nickels and pennies" test

let test () : bool =
  (coins 99) = 14
;; run_test "coins dimes, nickels, and pennies" test
```

- First homework assignment is already available
  - due: Tuesday September 15th at midnight
  - Sign yourself up to Codio to access it
- Introductory programming in OCaml
- Warm-up to using codio, etc.
- Some topics (lists, recursion) will be covered next week

Office Hours

- Faculty office hours are listed on the course web site.
  - See the links in Piazza

- TA Office hours listed on the web site calendar (under “Help/Office Hours” tab)
  - We use a tool called OhQ (Office Hours Queue)
  - TA office hours start next week
  - Schedule (on the web site) will be filled out soon

How to Succeed in CIS120

- Homework projects are key
  - The course videos, lecture notes, and slides explain the concepts, but *engaging with the material by writing code yourself is crucial.*
  - *START EARLY!*
  - *Seek help: Piazza, office hours, interactive sessions, etc.*

Even when in-person, much of the CIS120 learning experience is derived from the homework assignments.

- Our goal is to make learning the CIS 120 content as engaging and flexible as possible. You can use any combination of recordings, synchronous lectures, slides, and lecture notes that you prefer.

Feedback

- Teaching CIS120 in this way is new territory for us!
- We will occasionally ask you what's working and what isn't so that we can adapt.

- Feel free to voice suggestions and concerns by
  - posting (anonymously) on Piazza
  - catching us in office hours
  - sending mail to cis120@seas.upenn.edu

What’s Next?

- See the “CIS 120 – Week 1” Piazza post
- Watch the Week 1 videos
  - design process
  - value-oriented programming in OCaml

- Sign up for codio
- Jump in to HW01

- Friday’s Synchronous Sessions
  - demo Codio, jump-start HW1,
    Q&A about value-oriented programming in OCaml