Computers for Learning
CMSC 209

Prof. Diana Franklin

Overview of Course

- Learn about Learning
- Learn about the affordances technology brings
- Implement a learning application
  - Integrating what you've learned about learning
  - Designed for someone not just like you

Learning about Learning

- Readings on learning
- Writing reflections connecting readings to your own learning
- Discussing your reflections during class
- Engaging in some active learning activities in class

Why is sharing important?

- Others can learn from your learning experiences.
- Designing only for yourself is useless.
- Learning about what didn't work for others helps us design better.

What do YOU need for safe sharing?

What is required for sharing?

- A safe space
  - Only positive responses to sharing
  - Only positive statements about others
- Remember that struggles, challenges are all a normal part of the learning process
- Remember that other people's life experiences can be very, very different from yours
  - What is normal to you is not normal to others
- We must all respect, inside and outside the classroom, each other's experiences

Technical Content

- Game Engine implemented in Java
- Online tutorials
- Structured labs
- Short technical presentations in lecture
- Working in pairs (MUST work in pairs), must be in the same lab section.

Collaboration: Do's

https://www.youtube.com/watch?v=rG_U12uqRhE
Minute 3:59

Collaboration: Don'ts

Minute: 6:45

Implement a Learning Application

- Design
  - Profiling your own learning
  - Identifying a user (not just like you)
  - Designing a game
- In-class Activities
  - Initial design workshop
  - Final design flash talk
- Video demonstrating the use of your MVP app
- Paper describing the relationship between your design and the concepts learned in the readings

So many things to learn

- Java
- Object-oriented programming
- Event-based programming
- Adding to an existing codebase

Grading

- 4 Categories:
  - Reading Questions & Participation
    - You can miss 3 w/out penalty
    - Looking for depth of thought w/ participation
  - Midterm
    - Combo of Java + design + education
  - Labs
  - Final Project

  - people.cs.uchicago.edu/~dmfranklin - click on 209
  - https://www.classes.cs.uchicago.edu/archive/2019/fall/20900-1/

Java vs C

- Syntax
  - Arithmetic, variable declaration the same
  - Different:
    - Pointers
    - function calls
    - I/O (printing, reading from file, reading from user)
- Code organization
  - Strict rules about file names & code locations
- Abstract Data Type design
  - Package data together like with structs
  - Functions operating on data go with data
  - Do not touch data directly from outside code
- Object-Oriented programming
  - Relationships between different ADTs

Introduction to Java: Classes

- Structs: Group data
- Classes: Group data AND functions (methods)
  - ADT design methodology
  - All methods in Classname.java
  - Each Class implementation in separate file
  - Break up implementation into series of function calls
- Today: Just basic classes as a single unit

Puppy class

- Note all differences between C and Java implementation
- We will go through as many as possible today

Data – tied to functions

- variables: instance variables
- functions: instance methods
- Note how the functions are called

Printing to the screen

- system.out.println
- Note '+' signs to put things together
- Note you can even put in some variables directly!

public static void main

- void main
  - main is where all programs start
  - main returns nothing

public static void main

- public vs private:
  - Refers to either methods or variables
  - Determines scope (where it can be accessed from)
  - private means only accessible from within an instance method
  - public means accessible from outside of an instance method (via variable.method or variable.variable). Necessary since OS calls it.

public static void main

- static:
  - You don't need a variable in order to access the variable or call the method
  - static methods cannot access instance variables
    - Main needs to declare & allocate any variables it uses
  - Do not use static variables in these labs

Learning Starting Code

- Read the lab first to orient yourself to what you're learning and what you're looking for
- Play and observe
  - Do something (key, click)
  - What object responded?
  - What did it do?
- Find code: Think about the code structure
  - Organized by classname.java
- Predict which instructions did what actions
- Tinker
  - Replicate instruction, comment out original
  - Make changes and see how it affects execution