

CS 349

User Interfaces

Byron Weber Becker
Spring 2009

Slides mostly by Michael Terry

Welcome!

Instructor	Byron Weber Becker 12:30-1:00 MW in DC2017 (before class)	bwbecker@cs.uwaterloo.ca DC3105
TA	Matthew Kay TBA	mjskay@cs.uwaterloo.ca DC3591
TA	Jaime Ruiz TBA	jgruiz@cs.uwaterloo.ca DC3591

User Interfaces

How much of an interactive application's source code is dedicated to user interface concerns?

User Interfaces

- Significant portion of code in an interactive application dedicated to user interface and *interaction*

User Interfaces

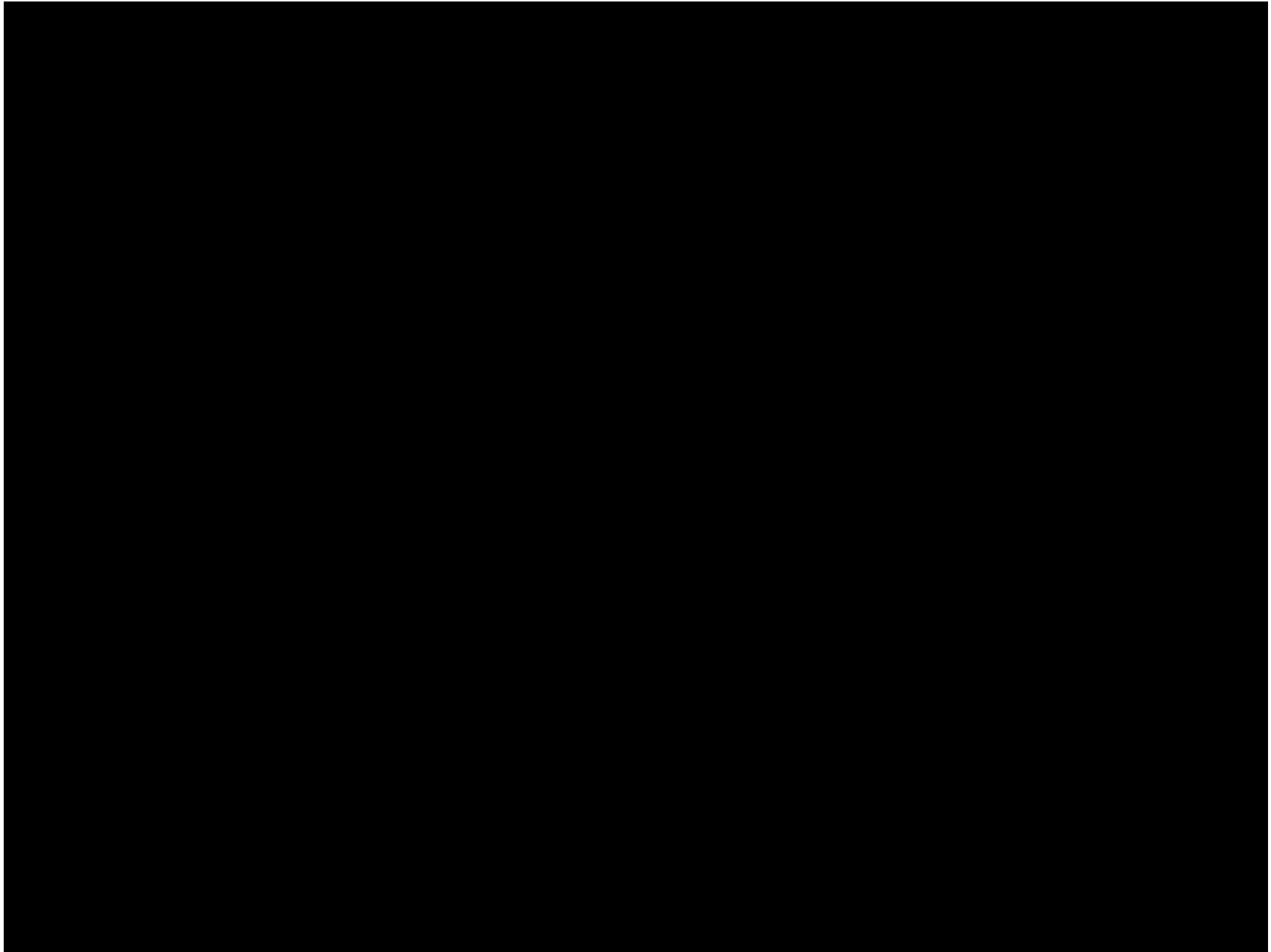
- What is a user interface?
- What isn't a user interface?

Definition: User Interface

- A user interface is the method by which an individual communicates intention to an artifact, and the artifact responds to that action.

Interfaces

- A jet fighter? (Avro Arrow, next slide)
- Does a microwave have an interface?
- A refrigerator?
- A hammer?



Interface vs. Interaction

- What is the difference between an interface and interaction?
- In common usage, *interface* refers to the external presentation to the user
 - Controls (*what* you can manipulate to communicate intent)
 - Visual, physical, auditory presentation (what the program uses to communicate its response)
- *Interaction* is used to connote *behavior*: The actions the user must invoke to perform a task and the corresponding responses
 - Interaction is *action*
 - Unfolds over time

Interface / Interaction Design

- What makes a good interface?
- What is the best interface you have ever used?
 - What makes it so good?
- Why is interaction design so hard?

Interaction Design

- Challenging because of variability in users and tasks
 - Varying levels of expertise
 - Range of tasks performed with the same tool
 - Example: Excel
- No one right way to design an interface, interfaces can always be improved

Empowering Users

- Well designed interfaces *empower* users to do things they couldn't otherwise do
 - Desktop publishing, grassroots journalism (blogs), movie production, music production, image editing, assistive technologies...

Empowering Users 2

- A well designed tool can literally change the world
 - The web browser, Linux, original Napster, the spreadsheet, email, instant messaging...
- This class will help you on your path to changing the world through computation

Course Goals

- Primary focus is on how to *construct* user interfaces
 - Provide foundation for you to be able to build highly interactive, usable applications
 - Expose you to underlying architecture of modern GUI toolkits
 - Teach a set of strategies applicable across a range of interface problem types
- Give *basic* exposure to design process. How to design interfaces is covered in CS489 (offered in the Fall and perhaps Winter)

Course Topics

- History, basic parts of user interface
- Output and input devices
- Windowing systems
- Event handling
- User interface software architectures
- Design patterns
- Undo
- Custom components
- Constraints and layout managers
- Design process
- Visual design
- Timers and threading in interface design...

See the syllabus on the web site!

Assignments

- Assignments meant to provide meaningful, engaging experiences in constructing interfaces...
- ...while giving you the opportunity to create applications you will want to share with others
- Lots of room for creativity in assignments
- Assignments require *significant* time coding
 - Do not underestimate the time it takes to code interactive applications that are intuitive and easy-to-use

Course Marking

- 4 assignments, 40% of mark
 - 1 bonus assignment worth up to 3% of your course grade.
 - Assignments are important. Failing the assignments will be grounds for failing the course.
- 2 midterms, 20% of mark
 - In class, Friday, May 29
 - In class, Friday, July 3
- 1 final, 40% of mark

Course Policies / Info

- Course website:
 - <http://www.student.cs.uwaterloo.ca/~cs349/s09/>
- *No late assignments accepted*
 - *Do not email assignments to instructor/TAs. They will be deleted*
- Assignments must all compile and execute in undergraduate environment or a (provided) virtual machine
- About newsgroups...

Course Wiki

- A wiki for CS349 has been set up at <http://crackle.cs.uwaterloo.ca/cs349-s09/>
- To create an account:
 - Click the “Special:SpecialPages” link in the Accounts section
 - Log in using...
 - On the Special Pages page, click the “Log in / Create Account” link
 - Do the obvious stuff...
- Help each other... Keep the thing organized... Use the “Recent changes” page
- I have a family and a life outside of working hours...

Course Textbook

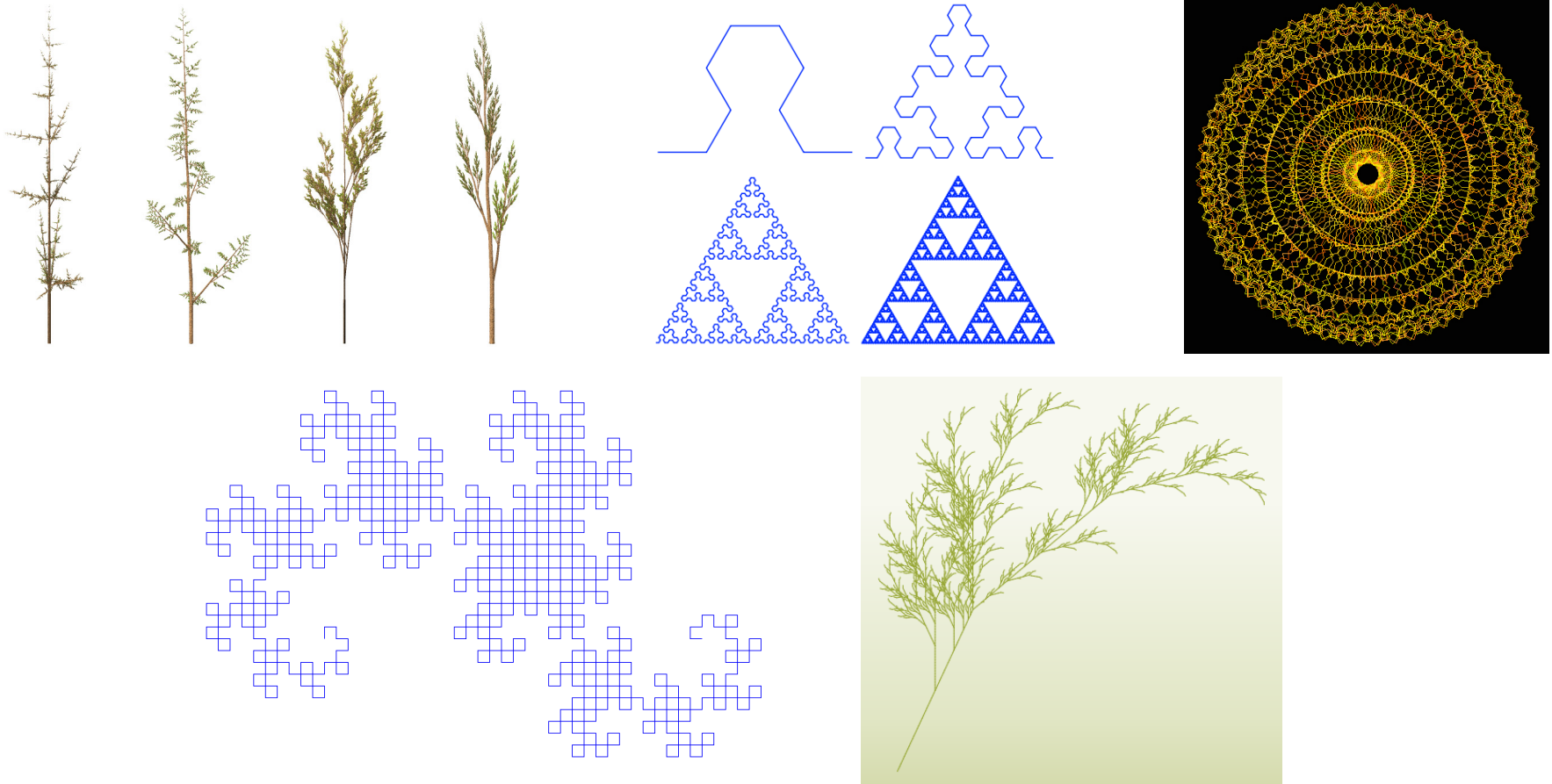
- Dan Olsen's *Developing User Interfaces*
- Course notes by Bill Cowan are available in Pixel Planet

Responsibilities

- Mine...
- Yours...

Assignment 1

- L-Systems/iterated systems interactive viewer



* All images from Wikipedia. Your viewer will be black and white. We will provide algorithms to generate the systems.

Preparing for First Assignment

- X Windows assignment
- You will implement a rudimentary widget toolkit, from scratch, using X lib, in C++
- We will provide:
 - Example code, including a basic test harness
 - A virtual machine with Ubuntu and tools/X installed if you prefer to work on your own machine
- You will need to flesh out code, maintaining basic API
 - No guarantees the code is complete or bug free

Virtual Machine

- VMWare image
 - 2GB download
 - 4GB uncompressed, virtual disk can grow to 10GB, max
- *For the adventurous only* (although it was successful last term)
 - Otherwise, simply use undergrad environment as normal
- *Your machine should have at least 1GB of RAM*
- Drag and drop possible (tested on Windows and Mac), as is networking
- If you use the VM, note this fact and we will test it on the VM. Otherwise, it will be tested in the student environment.

Getting Started

- Decide on your development environment (UW machines, VM, your own). Get things set up.
- Get the provided code to compile. You will need to complete several classes first.
- Get an XWindow to show up when the program runs. This isn't hard, but you'll need to remember a couple of things from the X lecture (coming up).
- A Clock class is provided. Get it to show up in the window and update itself periodically to keep time.
- Display an L-System. Update it based on a Timer.
- You are now in a position to sit back and reflect on the rest of the job. Make a detailed list of what you need to do. Then do it!

Think Aloud: Instructing Subject

- Ask participant to say whatever comes to mind as they use the application
- Indicate that anything that may be confusing is not their fault, but the fault of the application's design
 - You want them to feel comfortable criticizing the application
 - Shift blame from internal to external attribution (participant to designer)
- Be prepared to prompt subject to remind them to continue thinking aloud
 - “What are you thinking about now?”

Think Aloud Demo...

- Application: Keynote
- Task: Create two slides with animation in text and fancy slide transitions