

CSCI 053

Introduction to Software Development

http://www.seas.gwu.edu/~sibert/GW/John_Sibert.html

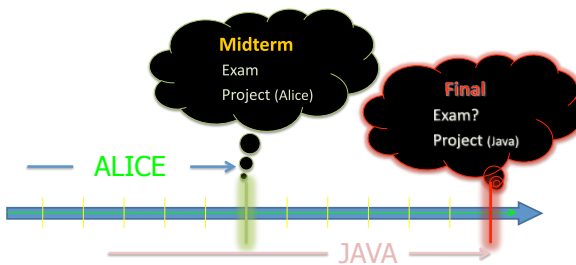
John Sibert

Department of Computer Science
The George Washington University
Fall, 2010

Information

- Class
CS 053
- Location
Tompkins Hall 211
- Schedule
Tue-Thu
Lab: Tue
- Office
- Phone
+1-202-460-2901
- Email
sibert@gwu.edu
- Book (required)
**Exploring Wonderland Java Programming
Using Alice and Media Computation**
ISBN-13
9780136001591
Authors
Dann, Cooper & Ericson
- Materials
Lab notebook
- Software
Alice www.alice.org
Java java.sun.com
Eclipse www.eclipse.org

Timeline



Classroom Expectations

Student Behavior

- Be prompt
Arrive in class on time, and turn in work on time.
- Be prepared
Have all class materials with you and know the due dates.
- Be a positive participant
Speak up when you have a question or a comment.
- Be a problem solver
Correct problems promptly before they escalate.
Ask for help when you are stuck.
- Be a team player
Give help when asked, and ask for help when needed.
Cite assistance of outside sources.

Classroom Expectations

Show Respect

- Value yourself
- Treat others with politeness and respect
- Honor the ideas and opinions of others
- Be responsible with property and belongings

Classroom Expectations

Student Behavior

Responsibility for Coursework

- Bring yourself, notebook/computer, textbook, planner, and appropriate writing tools to class.
- Know due dates, and submit all coursework on time.
- All assignments are posted on Blackboard and on the class Web site.
- <http://home.gwu.edu/~astefano/cs53>

Classroom Expectations

Student Behavior

Responsibility for Coursework

Policies

- Make sure you have read and understood the policies described on the course website.
- Food and beverages are not allowed in the classroom.
- Please be in class before the class has started.
- Homework assignments may be submitted late, but the grader will deduct 10% per day for every day it is late.

Classroom Expectations

Student Behavior

Responsibility for Coursework

Policies

Grading

• Quizzes	(weekly)	20%
• Homework	(weekly)	40%
•	Lab assignments	30%
•	Lab reports	10%
• Midterm Alice Project		15%
• Midterm Exam		10%
• Final Java Project		15%

Classroom Expectations

Student Behavior

Responsibility for Coursework


Policies


Grading


Our Pledge to Students


- We will respect you and work with you to solve problems
- We will promptly correct and offer feedback on your work
- We will work with you to meet your learning goals
- We will offer extra help if you ask

Course Objectives

What do you want the computer to do?
Creativity and problem solving 



Decide how to make it happen...
Design 

Make it happen!
Programming 




Critique the result
Testing, presenting, analysis and maintenance 

Creativity and Problem Solving

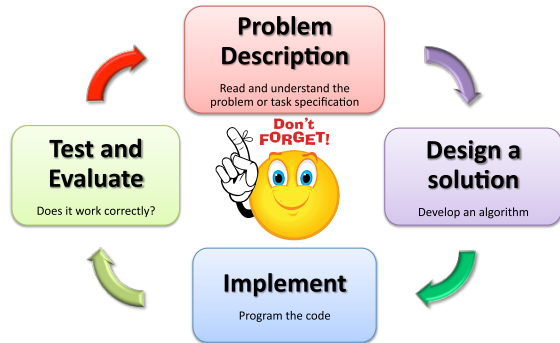
Can I teach these? Can you learn these?

Confucius' Wisdom

- "Tell me and I'll forget." 
- Show me and I'll remember. 
- Involve me and I'll understand!" 

Traditional CS Problem Solving



An Example Problem ...

- ... in need of a (creative?) solution:

The number 6 is perfect because
its proper divisors 1, 2, and 3
add up to 6:
 $1 + 2 + 3 = 6$

- What is the next perfect number?
- Two approaches:
 - brute force vs elegant solutions

Good computer science



- ... is elegant and efficient brute force!



First few weeks

- Problems will be “tell the following story”
 - Using computers as simulators
- Design a solution using Alice objects
 - Program the solution in Alice
 - Test the solution
- We will closely follow the textbook.

Week 1

• Objectives

- Introduce Alice
- Design an Alice program
- Build an Alice program
 - Use doInOrder and doTogether controls
- Create objects in Alice
 - Change an object's properties from within a program
 - Position objects relative to one another
- Create a method in Alice
 - Invoke methods on objects in Alice
 - Pass a parameter to a method in Alice
-

First things first

- Introduction to Alice
- Free development environment <http://www.alice.org>
 - under “Downloads” pick Alice 2.2
 - follow directions (ask your lab instructor if you get any problems)
- Used to create
 - 3D movies, 3D games
- Used to teach programming
 - Without the frustration of syntax errors
- When you start Alice
 - a screen offers you four tutorials
 - work through each and every one **ASAP**
 - run the tutorial in Alice as you read



Program Design

- Can be compared to movie making especially when programming Alice
- and we know that movies require:
 - lots of planning
 - forethought
 - sketching
 - tries
 - evaluation
 - retries
 - again and again
- Better design – less testing and retesting!



Film Making Terms



- Treatment: short prose version of the movie (*story*)
- Screenplay: full text, including dialog
- Scene: a part of the movie *usually a single location*
- Shot: part of scene *usually one camera location*
- Viewpoint: perspective of camera for a shot
- Storyboard: drawing or blueprint for shot

Programs

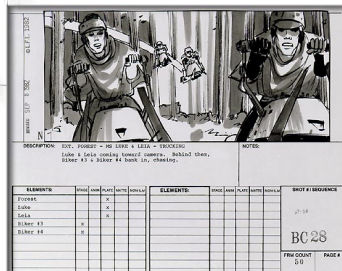
- We will treat them like movies, need a basic user story
 - a prose version of what the program does
 - it is a foundation of good software
 - it is like a screen play
 - it exists before any camera or acting is done

User Story

- A basic description of all events that make up the entire saga
 - Look for **nouns**:
 - these are your **objects** (characters and props)
 - objects can be visible or invisible, ex. animal vs wind*
 - Look for verbs:
 - these are your actions
 - Chronological flow of actions
- Algorithm!!!**

Designing a Solution in Alice

- To create a design, we borrow the idea of **storyboards** from professional animators at Disney, Pixar, etc.



Components of a storyboard-sketch

- Object's position (location)
- Object's pose (pos. of limbs, if it has any)
- Object's orientation (direction it is facing)
- Camera's viewpoint

Multiple sketches

- Can be joined with links into a sequence...



- ... forming a transition diagram

Example: Alice and the Rabbit

- User story:
 - Scene: Alice and White Rabbit on a grassy field, Alice to the left of Rabbit
 - Alice turns head towards user
 - Alice greets user
 - Rabbit turns to face user
 - Rabbit greets user
 - Alice introduces herself
 - Rabbit introduces himself
 - Simultaneously Alice and Rabbit say 'Welcome'

Storyboard

Detailed drawing of a shot

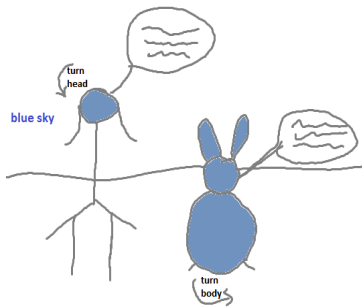
Arrows show movements of characters or camera

Collection of storyboards helps actors visualize film

Storyboard-sketch: storyboard for application screen

Example: Alice and the Rabbit

A storyboard-sketch



Example: Alice and the Rabbit

- Set the scene:
 - Select the grassy background
 - in mine, this is default
 - otherwise create new world and select grass
 - Click “Add objects” button
 - Pick Alice Liddell from People
 - Pick White Rabbit from Animals
 - Drag to move White Rabbit to the right

Things to Observe

- In the Alice window:
 - Menus
 - Buttons:
 - Object tree: with parts of objects
 - Details area:
 - The Alice world window
 - The editing area
 - Events area

Things to figure out

- How do you add objects?
 - How do you position objects?
- How do you access object sub-parts?
- What is that box drawn around a selected object?
 - notice how the details area adapts as you select objects and object parts

