Announcements

Monday is the President’s Day holiday
• The first lab of next week is canceled because of the holiday, but the labs will be open on Tuesday for last-minute help on Proj2b
Next Friday is the second Midterm
A Survey is posted in “Announcements”
• Please answer it … it will improve the class

Whole Picture

Solving large problems is tough
-- but approach them logically
and you will succeed

Problem Solving

Large problems share many properties:
• They are daunting -- there’s so much to do!
• We don’t know where to begin
• Not sure we know all of the tasks that must be done to produce a solution
• Not sure how to do all of the parts -- new knowledge may be required
• Not sure if it is within our capability -- maybe an expert is needed

Solve you will succeed; not trying concedes defeat

Problem Decomposition

“Divide and conquer” is a political
strategy, military strategy & IT strategy
Top-level Plan --
1. Describe (in any language) a series of steps
    that produce a solution
2. For each step, solve it or decompose further
3. For steps needing decomposition, repeat 2
4. Assemble solutions and test correctness
5. When solution fully assembled, evaluate

More Specifics

We will step through the process, using
Project 2 as an example:
• Problem decomposition is mostly common sense
• Process is not algorithmic
• Problem decomposition is
to help you, so apply it as needed

1. Give Steps to a Solution

Specify (in any language) a series of
steps that produce a solution
• For a huge problem the steps may at first be
    vague, but they can be (& must be) made
    more precise as the whole picture emerges
• The goal is an algorithm(s), so …
• List & describe the inputs
• List & describe the outputs
• Be guided in figuring out the steps by the
    need to transform the inputs into the outputs

You will be
coming things.
**What Are Steps for OC?**

Orange Concentrate:
- Build Basic GUI
- Build mouse-sensed keypad
- Build the pattern array
- Set up control keys
- Write functions for ctrl keys
- Set up customizing keys
- Primp design & make cool

---

**PERT**

PERT is Program Evaluation & Review Technique ... developed in 1950s
- Diagrams show the dependencies visually

---

**2&3. Solve or Decompose**

For each step, solve it or decompose it further, i.e., apply same technique
- Most “top level” steps can’t be brained out, and need further decomposition
- “Top level” steps often seem huge, too
- The technique allows one to concentrate on only one problem at a time
- As before, focus on inputs, outputs, process to transform inputs into outputs

---

**2&3. Solve or Decompose**

“Build mouse-sensed keypad”

- Layout GIF 7x7 grid
- Build onClick e-handler
- Setup to change grid color
- Define GIF prefetch array
- Prefetch brown & orange
- Build mouse e-handlers
- Update colors in e-handlers

---

**2&3. Solve or Decompose**

“Build mouse-sensed keypad”

- Layout GIF 7x7 grid
- Build onClick e-handler
- Setup to change grid color
- Define GIF prefetch array
- Prefetch brown & orange
- Build mouse e-handlers
- Update colors in e-handlers

---
4. Assemble Parts

Assemble Solutions & Test Correctness

- Putting solutions together can be tough because of different assumptions made while solving the parts – it always happens
- When working alone it is common to combine parts along the way and to test continuously
- Because of the need to test, pick a good order to solve the problems

Finding something working quickly is best.

4. Assemble Parts

Proj2 solves & assembles parts together in a ‘good’ order

- Most parts of Project 2 can use the developing solution for testing – that’s ‘good’
- Notice step 284 in Project 2 was included simply to test
- Parts mismatch is common problem, but not in Proj2

4. Assemble Parts

Summary

Large problems can be solved by the ‘divide and conquer’ technique

- The process is “top down” – get a top level solution even if it is vague, imprecise
- Whenever you cannot produce a solution to a step directly, reapply the technique
- The start and first several steps will be daunting, but the process works
- Get part of solution working quickly if possible