Whole Picture

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

© 2004 Lawrence Snyder
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 we know how to do all of the parts -- new knowledge may be required
- Not sure it is within our capability -- maybe an expert is needed

Assume 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 an old 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
What Are Steps for PC?

Red + Blue = Purple Concentration
What Are Steps for PC?

Purple Concentrate:

- Build Basic GUI
- Set up control keys
- Build the Display Grid
- Build mouse-sensed KeyGrid
- Write functions for ctrl keys
- Set up customizing keys
- Primp design & make cool
PERT is Program Evaluation & Review Technique ... developed in 1950s

- Diagrams show the dependencies visually

Build GUI

Setup Control Keys

Build Mouse Keypad

Build Display Grid

Write Functions for CK

Primp & Coolify
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

Often, “last” decomposition done during solution
2&3. Solve or Decompose

“Build mouse-sensed keypad”
2&3. Solve or Decompose

“Build mouse-sensed keypad”

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

Need to learn about mouse events

Red + Blue = Purple Concentration
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

Getting something working quickly is best
4. Assemble Parts

Proj2 solves & assembles parts together in a ‘good’ order
4. Assemble Parts

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

- Most parts of Project 2 use the developing solution for testing -- that’s ‘good’
- Notice adding steps to test a solution may be wise
- Parts mismatch is common problem, but not in Proj2
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