Project: Configuration Puzzles

Logistics

- E-mail
  - Anyone not receive test e-mail?

Goals

1. Improve your design skills
2. Development experience in C++ on a large project
3. Appreciation that C++ is not Java

What You Will be doing

- Designing a general framework
  - Set of classes and interfaces that can be applied to a variety of related problems.
- Implement your design in C++
- Apply to 3 related (and specific problems)

Motivation: Tic Tac Toe
Problems to solve

- Clock problem
- Measuring Water
- Lloyd Block Puzzle

Clock problem

- Clock battery has gone dead
- Must reset clock to correct time
- No minute hand...only hour hand
- Can move clock hands forward or backwards

Measuring Water

- A man needs a given volume of water.
- He has a set of containers of different volumes
- He can:
  - Fill any empty container (completely) from the river
  - Empty any full container completely
  - Pour the contents of one container into another until the source is empty or the destination is full.

Measuring water

- Example:
  - Goal: Man needs 4 liters of water
  - Containers: 1 5 liter, 1 3 liter

Measuring water

- Example:
  - Goal: Man needs 4 liters of water
  - Containers: 1 5 liter, 1 3 liter
Lloyd Block puzzle
• Best shown by an example

Problems to solve
• Clock problem
• Measuring Water
• Block Puzzle
• Each problem can be expressed as
  – Being in a given configuration
  – There is a start configuration
  – There is a goal configuration
  – There are a number of actions that can be taken to get from one configuration to the next.

Clock Problem
• Configuration:
  – Position of clock hand
• Initial state:
  – Incorrect time
• Final state:
  – Correct time
• Rules:
  – Can move hour hand forward 1 hour.
  – Can move hour hand back 1 hour.

Measuring Water
• Configuration:
  – Amount of water in each container
• Initial state:
  – All containers empty
• Final state:
  – One container with desired amount of water
• Rules:
  – Fill any empty container (completely) from the river
  – Empty any full container completely
  – Pour the contents of one container into another until the source is empty or the destination is full.

Block Puzzle
• Configuration:
  – Label and position of each block
• Initial state:
  – Initial Configuration
• Final state:
  – Goal configuration
• Rules:
  – Can move adjacent block into empty slot.

Generic Puzzle
• We can “solve” a generic puzzle in a general manner.
• Some possible classes:
  – Configuration
    • Start configuration
    • Goal configuration
  – Action
Algorithm

- Create an empty queue of configurations
- Insert start configuration onto queue
- while (queue is not empty and head of queue does not fulfill goal)
  - remove configuration at head of queue (C)
  - For each Action (A)
    - Apply A to C
      - Place resultant configuration on queue (if it has not already been seen)
- If queue is empty then there is no solution
- Else goal configuration is on the queue.

Things to note

- Algorithm is generic
  - No mention of any specific problem
- Result of applying an action to a configuration is another configuration.
- Configuration must test if it is the goal.
- Algorithm is a breadth first search of search space.
  - Remember tic-tac-toe

Motivation: Tic Tac Toe

Your task

1. Develop the generic framework
2. Implement generic framework
3. For each problem
   1. Code problem specific classes
   2. Modify design as necessary
   3. Modify framework code as necessary
   4. Submit framework code as well as code for all problems solved to date.

Questions so far?

Submissions

- Part 1 (due Sept 25th) – Individual
  - Design framework and document
  - Write code for design
  - Implement clock using the framework.

Things left unsaid

- Still need to record path to the solution.
- How does one know if a configuration “has been seen”?

Things to note

- Algorithm is generic
  - No mention of any specific problem
- Result of applying an action to a configuration is another configuration.
- Configuration must test if it is the goal.
- Algorithm is a breadth first search of search space.
  - Remember tic-tac-toe
Submissions

- Part 2 (due October 16th) – Group
  - Choose best framework design of group members
  - (or develop a third as a combo of the two)
  - Reimplement clock using chosen framework
  - Implement water using chosen framework

Submissions

- Part 3 (due October 30th) – Group
  - Implement block puzzle using common framework

Submissions

- Part 4 (November 13th) – Individual
  - Implement a variant of clock with final framework.

Note on all submissions

- All problems must be solved using framework
- Framework can “evolve” from one checkpoint to the next.
  - All problems solved to date must work with modified framework.

Questions?

- Thursday
  - C++ classes.

- Reminder…
  - Do prelab activities before lab tomorrow.