Outline

- Single-Player Games = Planning
- Planning under incomplete information
Single-Player Games
A (Very) Simple Planning Problem

Pressing button $a$ toggles $p$.
Pressing button $b$ interchanges $p$ and $q$.
Initially, $p$ and $q$ are off. Goal: $p$ and $q$ are on.
Problem Specification

role(robot)

Legality

legal(robot,a)
legal(robot,b)

Update

next(p) <= does(robot,a) ∧ ¬true(p)
∨ does(robot,b) ∧ true(q)

next(q) <= does(robot,a) ∧ true(q)
∨ does(robot,b) ∧ true(p)

Termination and Goal

terminal <= true(p) ∧ true(q)
goal(robot,100) <= true(p) ∧ true(q)
State Transition System

Features: \( p, q \)

Actions: \( a, b \)
Solving Single-Player Games = Planning

- **Initial state**
  \{ \}

- **Actions**
  a. Preconditions: none
     Effects: toggles truth-value of \( p \)
  b. Preconditions: none
     Effects: interchanges truth-values of \( p \) and \( q \)

- **Goal**
  \( p \land q \)

**Solution (= Plan):** a, b, a
Many single-player games can be solved using standard search techniques introduced earlier in this course

- Iterative deepening
- Bidirectional search

Special techniques

- Constraint solving (suitable for Sudoku, Gene Sequencing and the like)
- Answer set programming (suitable for Peg Jumping, 15-Puzzle and the like)

Informed search uses heuristic functions. In general game playing, the rules are not known in advance and heuristics must be constructed automatically. More on this a little later.
Planning Under Incomplete Information: Maze World

Initial State: \(ac\) (robot in a, gold in c)
Environment Model

\( i = "in\ hand" \)
Agent Actions

- m = "move clockwise"
- g = "grab"
- d = "drop"
Initial State and Goal
Planning

Planning is the process of finding a transition diagram *for our agent* that causes its environment to go from any initial state to a goal state.

Planning can be done *offline* and the resulting plan/program installed in the agent or the planning can be done *online* followed by execution.
State Space Planning
Incompleteness

Possible sources of incompleteness:
Partial knowledge of
- Initial state
- Transition diagram for environment
- Goal

Complete Planning Techniques under incomplete information
- Coercion (e.g. do the grab action at all locations)
- Conditional plan (e.g. if see the gold grab it; else move)

Postponement Techniques
- Delayed planning
Initial State Uncertainty
Sequential State Set Progression
Sequential State Set Plan

aa  
   ab  
   ac  
   ad  

move

ba  
   bb  
   bc  
   bd  

grab

ba  
   bi  
   bc  
   bd  

move

ca  
   ci  
   cc  
   cd  

grab

cac  

move

da  
   di  
   dd  

grab

da  
   di  

move

aa  
   ai  

drop

aa

© Michael Thielscher, Michael Genesereth 2011
Plan Execution
Conditional State Set Progression

\[ bi \rightarrow g \rightarrow \begin{cases} bi & \text{if } 1 \\ bb & \text{if } 0 \end{cases} \rightarrow m \rightarrow \begin{cases} ci & \text{if } 1 \\ cb & \text{if } 0 \end{cases} \]
Conditional State Set Plan

\[ m \rightarrow 1/g \rightarrow 0/m \rightarrow 1/g \rightarrow g \rightarrow d \rightarrow m \]
Comparison

Sequential plan
- possible that no plan exists
- plan may contain redundant moves

Conditional plan
- large search space

Delayed planning
- irreversibility problematic

As we can see from this analysis, it is sometimes desirable for an agent to do only a portion of its planning up front, secure in the knowledge that it can do more later as necessary.

Planning can be done offline and the resulting plan/program executed during play or the planning can be done online and interleaved with execution.