AI Programming Example: Age of Kings

CS 395 Game Design
Spring 2002
Updated class schedule

- Progress presentations will be May 23rd and May 28th
  - Someone from your team (and preferably all of you) must be present on your assigned presentation day.

- Two more homework assignments:
  - Today: Age of Kings AI programming
  - Creating an object in The Sims

- Rest of your effort should go to your term project
AI in Age of Kings

• Someone to outwit, outfight
• Crucial element in gameplay
  – Makes or breaks the single-player game
  – Enhances the multiplayer game
• AI for strategy games is an extremely hard problem
  – Requires reasoning about space, time, resources…
  – Workaround 1: Simplifications in game world
  – Workaround 2: Cheating
Rational reconstruction of AoK AI

• No access to source, inferring from scripting language & its documentation
Builder AI

Where to build things.
Concerns: Efficient layout
Handling battles
Concerns: Path-finding, formations, unit-level decision making
Overall strategy
Concerns: What to build, tuning military strategy, diplomacy, how to win (or lose)
Sensors/effectors for hardwired AI’s

Builder AI

Tightly tied to game engine

Tactical AI

Need extensive spatial computations, rapid manipulation of many units

Strategy AI
Sensors/Effectors for strategy AI

Most actions tune the behavior of the other AI's

Strategy AI

Builder AI

Tactical AI

Some hard-wired sensors (global properties of civ), some direct actions
Gameplay tuning

- Really fast and arcane, need to be developer to tweak
- Most responsible for personality of opponent, make scriptable
Scripting languages for AI’s

- Range from homebrew to Python/Perl to Scheme
- Age of Kings’ scripting language is based on production rule systems
Production rules motivation

• Designed to model human problem solving
  – Developed by Allen Newell and Herb Simon
• Each piece of knowledge can be added independently
  – Models human ability to learn incrementally
• Often used in building expert systems
• Sometimes used in psychological modeling
  – Examples: Act-R, SOAR, 4CAPS, …
Simplifications in Age of Kings rules

- No symbolic assertions or pattern variables
  - Instead, use tests from pre-defined vocabulary of procedures
  - Significance: Can’t easily select from multiple items satisfying a criterion, can’t extend the representation language

- No assertions of symbolic data in actions
  - Instead, use vocabulary of executable procedures for actions
  - Significance: Marginal reasoning abilities, due to lack of chaining.

- No conflict resolution (e.g., picking one rule to run)
  - Instead, run all that match every cycle (several cycles/second)
  - Significance: Good idea given purpose is control of a system with multiple parts all operating (conceptually) at once)
#load-if-defined DIFFICULTY-MODERATE

;attack timer - once in feudal, or if rushing, launch attacks every 5 minutes

(defrule
  (or (current-age >= castle-age)
       (goal rush-control RUSHING))
  =>
  (enable-timer t-attackgroup 1)
  (disable-self))

(defrule
  (timer-triggered t-attackgroup)
  =>
  (disable-timer t-attackgroup)
  (enable-timer t-attackgroup 300))

#end-if
Time to spelunk…