Making Decisions

CS395 GAI
Spring 2005
Overview
How to make decisions?

• The decisions to be made
  – For military units, move, fortify, attack…
  – For cities, what to build, where workers go, …
  – For nation, what to research, what government, …

• Decision-making strategies
  – Random
  – Procedures
The FreeCiv version of the problem

| Explore World |
| Conquer World |
| Win Game |

Diagram of a map with cities such as Tuckburrow, Hobbiton, Bywater, and Longbottom.
Model 0: Just code it

• Example: The FAP

• Advantages:
  – We know how to write code
  – Close to the details

• Disadvantages:
  – Unless very careful, hard to debug and maintain
  – Too close to the details
Example
Problems with Model 0

• Too many decisions to keep track of
  – Leads to brittleness

• Lack of modularity
  – Leads to unmaintainable code
Model 1: Very high-level programming language

- To explore world,
  - Explore starting continent
  - Research technology for ships
  - Build ships
  - Explore ocean
  - For each new continent,
    - Ferry explorer to it
    - Explore continent
Build another ship, or settler for improvements?

Hug shoreline, or head for open ocean?
Example
Problems with Model 1

• Standard programming model leads to sequential thinking
  – Orchestrating lots of parallel activities critical to success in strategy games

• Where is the state kept across turns?

• How do you dynamically respond to threats and opportunities?
Model 2: Use local control

• Assign each unit a finite state machine or simple program
  – Completely determines its behavior
  – Can be conditional, based on world around it
Example: Explorer

• State: GetMission
  – Ask for mission (<mission> <loc>)
  – If <mission> = explore,
    • State := Explore; Destination = <loc>
  – If <mission> = embark,
    • State := Embark; Destination = <loc>
  – If <mission> = none,
    • State := Wander;
Explorer state machine

- Wander
- Go to Port
- Wait
- Embark
- Ride
- Disembark
- GetMission
- Explore
- Go to Port
- Embark
- Ride
- Disembark
- GetMission
- Explore
Explorer state machine

- Wander
- Explore
- GetMission
- Go to Port
- Wait
- Embark
- Ride
- Disembark
- Port
- Wander
- GetMission
- Explore
- Go to Port
- Wait
- Embark
- Ride
- Disembark
- Port
Problems with Model 2

• Coordination
  – How does Explorer know when its ride has sunk?
  – How does Explorer know who its ride is?
  – Exactly who is assigning these missions?
Model 3: Combine 1, 2a

• High-level program provides executive function
  – Assigns state machines to run units
  – Responds to requests back built into some of the states
  – Keeps track of information needed for coordination
    • Example: Explorer/Transport pairings
Tasks

• A set of activities carried out to serve some goal
  – E.g., explore a continent
• Units are assigned to tasks
  – Assigned task determines state machine
• Tasks can end
  – Finished, e.g., continent explored, enemy destroyed
  – Shelved, e.g., postpone building spaceship to fend off invading horde
Example: Exploration

• Explore Home Continent
  – Unit112 (Explorer): LandExplore

• Explore Atlantis
  – Unit113 (Warrior): LandExplore
  – Unit115 (Trireme): Ferry

• Explore Ocean
  – Unit 114 (Trireme): ShoreHugExplore
  – Unit116 (Trireme): DeepOceanExplore
Problem: Managing cities

- Cities can be assigned to tasks
  - ProductionInvestment
  - DefenseInvestment
  - WonderProduction
  - UnitProduction
  - SettlerProduction
  - MoneyProduction
How to assign tasks?

• Consider current needs
  – To explore land, need assignable land unit
  – To explore ocean, need assignable ship
  – To explore other lands, need assignable land unit and assignable ferry

• Consider strengths/weaknesses of cities
  – Some have lots of resources
  – Some have strategic locations
Detecting trouble

• Destruction of a unit
  – Look at task it was assigned to
  – Is task still worth accomplishing?
  – Find/build other units to accomplish it

• Spotting unit from another civilization
  – Is the spotter well enough defended?
  – Should an interaction task take precedent over its current task?
  – Should tasks be re-prioritized?
Next week

• Planning
• AI Architectures