cisc3665
game design
fall 2011
lecture # II.1
introduction to game AI and agents

topics:

- introduction to game AI
- introduction to agents

references:

- notes on agents from *An Introduction to Multiagent Systems*, by Michael Wooldridge, Wiley (2002), chapter 1-2
- notes on game AI from *AI for Game Developers*, by David M. Bourg and Glenn Seamann, O'Reilly (2004), chapter 1

introduction to game AI

- *AI* = artificial intelligence
- *game AI* typically refers to giving non-player characters some level of intelligence, the appearance of emotion, different personalities, etc.
- two categories of game AI techniques:
  - *deterministic*: specified and predictable; non-player character is specifically coded to always do the same thing in the same situation
  - *non-deterministic*: unpredictable and has a degree of uncertainty; non-player character may vary its behavior when faced with the same situation multiple times
- *cheating* in game AI:
  giving a non-player character access to all information stored in the game (beyond what the human player has and beyond what the character might realistically know)
- *finite state machines (FSM)*:
  used to enumerate possible game states and design possible transitions between them

- *fuzzy logic*:
  allows definition of less precise states, i.e., states within bounds
  for example:
  a FSM might define a state where $distance\_to\_goal = 10$ and $energy = 5$
  whereas fuzzy logic might define the same state as $distance\_to\_goal = close$ and $energy = low$, where $close$ is defined as a range of distances and low is defined as a range of energy values
- *path-planning*:
  technique for non-player characters to navigate within a virtual world; stems from robotics
- *artificial life*:
  swarming and flocking techniques for controlling the behaviors of multiple (typically many) non-player characters
- *machine learning*:
  techniques for non-player characters to learn from experience
- *autonomous agents*:
  techniques for controlling the behavior of intelligent non-player characters

introduction to agents

- an *agent* is a computer system that is capable of *autonomous* action within its environment
- an agent figures out what needs to be done to satisfy its *goals*, on its own—rather than being told what to do by a human user
- a *multiagent* system is one that consists of multiple agents which *interact* with each other
- multiagent systems *(MAS)* involve cooperation, coordination and negotiation amongst agents

properties of agents

- the main aspect of an agent is that it is *autonomous*—capable of acting independently, exhibiting control over its internal state
- an agent exists in an *environment*
- an agent receives input from its environment — *perception*
- an agent makes decisions about what to do, based on its perceptions of its environment, its knowledge about its internal state, and its goals
- an agent's output — *actions* — can effect its internal state, its environment and/or other agents
- an example of a trivial agent is a light switch
- key properties:
  - *intelligence*: an agent can make decisions on its own
  - *reactive*: an agent responds to changes in its environment
  - *pro-active*: an agent attempts to achieve goals
  - *social*: an agent interacts with other agents

- additional (optional) properties:
  - *rationality*: an agent acts in order to achieve its goals, and will not act so as to thwart or prevent its goals
  - *adaptive*: an agent learns through experience
  - *intentional*: an agent has *beliefs* (knowledge about itself and its environment) and *desires* (goals for what it "wants" to achieve)

agents vs objects

- an object:
  - encapsulates some *state*
  - communicates via message passing
  - has methods that correspond to operations to be performed with a given state
- an agent:
  - is autonomous
  - is intelligent
  - is active (has at least one thread of control—not necessarily true of an object)
- "objects do it for free; agents do it because they want to..."
- even though an agent is not an object, an agent can be implemented using objects

properties of environments

- *accessible* vs *inaccessible*: can the agent obtain accurate, current information about its environment?
- *deterministic* vs *non-deterministic*: does each action have a single, guaranteed effect?
- *episodic* vs *non-episodic*: can the agent's performance be broken down into independent, discrete episodes?
- *static* vs *dynamic*: does the environment change only in response to actions performed by an agent?
- *discrete* vs *continuous*: can the environment be represented as a set of fixed, finite number of distinct states?

formalisms for describing abstract agent architectures

- the set of possible environment states: $E = \{e, e', \ldots\}$
- the set of possible agent actions: $Ac = \{\alpha, \alpha', \ldots\}$
- a *run*, r, is a sequence of states and actions:

$$r : e_0 \xrightarrow{\alpha_0} e_1 \xrightarrow{\alpha_1} e_2 \xrightarrow{\alpha_2} e_3 \ldots \xrightarrow{\alpha_{u-1}} e_u$$

  and $\mathcal{R} = \{r_0, r_1, r_2, \ldots\}$, the set of all such sequences
- a run can end with an action: $\mathcal{R}^{Ac}$, or a run can end with a state, $\mathcal{R}^{E}$
- a *state transformer function* represents the behavior of an environment as it is effected by an action:

$$\tau : \mathcal{R}^{Ac} \rightarrow \wp(E)$$

  where $\wp$ is the power set of E, i.e., $E \times E$
- we make the assumptions that environments are *history-dependent* and *non-determimistic*
- if $\tau(r) = \emptyset$, then a run has ended ("game over")

- an *environment*, Env, is a tuple: $\langle E, e_0, \tau \rangle$, where $e_0 \in E$ is the initial state and τ is the transformer function
- an *agent*, Ag, is a function which maps runs to actions:

$$Ag : \mathcal{R}^E \rightarrow Ac$$

  and $\mathcal{AG}$ is the set of all agents Ag
- a *system* is a pair containing an agent, Ag, and an environment, Env
- associated with any system is a set of runs of Ag in Env:

$$\mathcal{R}(Ag, Env)$$

- we make the assumption that $\mathcal{R}(Ag, Env)$ contains only runs that have ended

- So, a sequence:

$$(e_0, \alpha_0, e_1, \alpha_1, e_2, \ldots)$$

  represents a run of agent Ag in environment $Env = \langle E, e_0, \tau \rangle$ if:
  1. e_0 is the initial state of Env
  2. $\alpha_0 = Ag(e_0)$
     and
  3. $e_u \in \tau((e_0, \alpha_0, \ldots, \alpha_{u-1}))$,
     where
     $u > 0$ and
     $\alpha_u = Ag((e_0, \alpha_0, \ldots, e_u))$

to do

- read ch 2 of Wooldridge book (handout)
- work on assignment for unit I (labI.2), which is due on SEPT 25
  (electronic submission instructions are forthcoming...)