Principles of Computer Game Design and Implementation

Lecture 19

Game Artificial Intelligence

In computer games, AI refers to a collection of techniques that control the computer player

- *AI is anything that contributes to the perceived intelligence of an entity, regardless of what*ʼ*s under the hood*

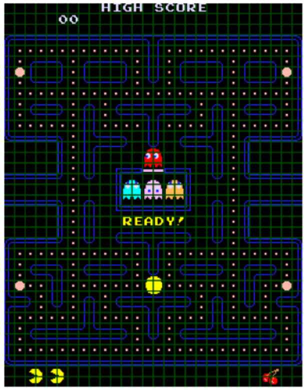
Outline for today

- Introduction
- Sense-Think-Act Cycle:
  - Sensing

Some History

- Initially, there was no computer player in games (Pong, Space Wars, ***Tetris***)
- Early examples are not very smart
  - Pac-Man ghosts (chaise mode):
    - Plan just one step ahead
    - Never reverse the direction of travel
    - Certain tiles enforce certain behaviour

AI and level developed together



- Target tile depends on the ghost ***personality***
  - Move where the player is
  - Move where the player *will be*

No really intelligent behaviour

Requirements

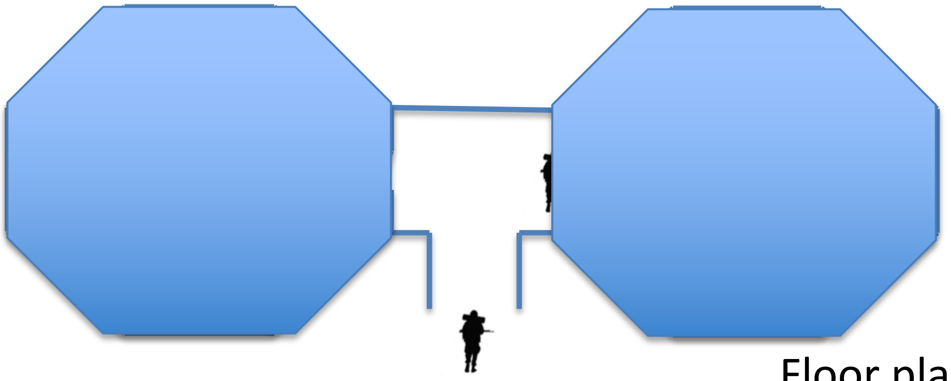
- Be intelligent but **purposely** flawed
- Have no **unintended** weaknesses
- Perform within CPU and memory constraints
  - Cheat! (but players don’t like it. If they notice.)
- Be configurable by game designers /player
- Be **visible**
  - Perception window can be very small

Be Visible

- Make sure player knows what agent is doing
  - *Patrolling agent speaks on the radio*
  - **Changes** in behaviour are easy to spot

- Eliminate invisible AI activity
  - "Level of detail": if player sees, show activity, else don't do it.

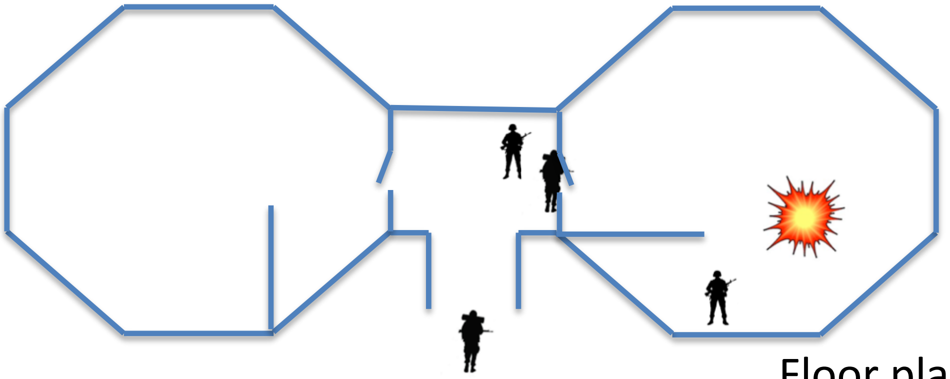
Example: 100% Optimisation



Floor plan

- That’s what we perceive

Example: 100% Optimisation



Floor plan

- What happens:
  - Soldiers are spawned by the control system *when needed*
  - Helps with the CPU cycles

Traditional AI

Science and Engineering

- the *science* of understanding intelligent entities - of developing theories which attempt to explain and predict the nature of such entities;
- the *engineering* of intelligent entities.

Four Views of AI

- **Systems that think like humans**
  - cognitive science, expert systems
- **Systems that act like humans**
  - The Turing Test, chess programs
- **Systems that think rationally**
  - Approaches based on logic and mathematics
- **Systems that act rationally**
  - Contemporary agent based approaches

It's all about **substance!**

Game AI

It’s all about ***appearance*!**

- Biased towards engineering: developing algorithms that *appear* to behave intelligently
  - human or animal like

- Sometimes these two approaches do overlap
  - But not necessarily

Approaches to Game AI

- “Proper”
  - FSMs, planning, path finding, collision avoidance, expert systems, fuzzy logic…
- “Ad-Hoc”
  - Hacks
    - Animation can *show* more intelligence than algorithm
  - Heuristics
    - Choose most constraint option
    - Do most difficult thing first
    - Try the most promising thing first

Intelligent Entities

It is convenient to distinguish

- Game agents
  - – autonomous entities that observe and act upon an environment.
    - Game characters
- Virtual Player
  - – performs the same operations as the human player.
    - Chess

Agents and Virtual Player

- Agents, no virtual player
  - Shooters, racing, …
- Virtual player, no agents
  - Chess, …
- Both
  - Strategy games, team sport games, …

Agents

- Act as
  - enemies, allies, neutral characters
- Constantly go through a
  - *Sense – Think – Act* cycle
    - Sometimes can learn new behaviours

- Example: first-person shooter enemies, other car drivers, units in strategies

Sense-Think-Act Cycle: Sensing

- Agent can have access to **perfect** information of the game world
  - May be expensive/difficult to tease out useful info
- Game World Information
  - Complete terrain layout
  - Location and state of every game object
  - Location and state of player
- But isn’t this cheating???

Sensing: Enforcing Limitations

- Human limitations?
- Limitations such as
  - Not knowing about unexplored areas
  - Not seeing through walls
  - Not knowing location or state of player
- Can only know about things seen, heard, or told about
- Must create a sensing model

Sensing:
Human Vision Model for Agents

- Get a list of all objects or agents; for each:
  1. Is it within the viewing distance of the agent?
     - How far can the agent see?
     - What does the code look like?
  2. Is it within the viewing angle of the agent?
     - What is the agent' s viewing angle?
     - What does the code look like?
  3. Is it unobscured by the environment?
     - Most expensive test, so it is purposely last
     - What does the code look like?

Partial Visibility

- Check for bounding body visibility
  - Expensive
- React to player **motion**
  - Corresponds nicely to the human perception

Sensing: Human Hearing Model

- Humans can hear sounds
  - Can recognize sounds
    - Knows what emits each sound
  - Can sense volume
    - Indicates distance of sound
  - Can sense pitch
    - Sounds muffled through walls have more bass
  - Can sense location
    - Where sound is coming from

Sensing: Modeling Hearing Efficiently

- Event-based approach
  - When sound is emitted, it alerts interested agents
- Use distance and zones to determine how far sound can travel

Sensing: Communication

- Agents might talk amongst themselves!
  - Guards might alert other guards
  - Agents witness player location and spread the word
- Model sensed knowledge through communication
  - Event-driven when agents within vicinity of each other

Sensing: Reaction Times

- Agents shouldn't see, hear, communicate instantaneously
- Players notice!
- Build in artificial reaction times
  - Vision: ¼ to ½ second
  - Hearing: ¼ to ½ second
  - Communication: > 2 seconds