Principles of Computer Game Design and Implementation

Lecture 29
Putting It All Together

• Games are unimaginable without AI
  – (Except for puzzles, casual games,...)
  – No AI – no computer adversary/companion

• Good AI makes a game interesting
  – No “silver bullet” solution
    • What is more important, smooth paths or smart decisions?

• We looked at some AI techniques
Virtual Player V Smart Agents:
Smart virtual player, simple agents
Virtual Player V Smart Agents: Smart Agents

Doom 3

Serious Sam
Virtual Player V Smart Agents: Groups of Smart Agents

F.E.A.R.
Virtual Player V Smart Agents: 
Smart Virtual Player, Smart Agents

Lord of The Rings: 
The Battle for Middle Earth

Perfect Dark Zero
Virtual Player V Smart Agents: Techniques

• Smart Virtual Player
  – Turn-based game search (board games)
  – Rule based system (RTS: C&C, Age of Empires,...)

• Smart Agents
  – Planning (F.E.A.R)
  – Behaviour Trees (Halo)

• Smart VP, smart agents
  – Agent coalition
  – Hierarchical models (FSMs, BTs)
Game AI Design

• Decide what AI should and **should not** do
  – Include *everything*
  – E.g. *Empire earth* project took a month, 30 pages

• Brainstorm different techniques

• Identify components

• Identify interfaces
Game AI: Requirements

• Be intelligent but *purposely* flawed
• Have no *unintended* weaknesses
• Perform within CPU and memory constraints
  – Cheat!
    • Precompute!!
• Be configurable by game designers /player
• Be *visible*
Artificial Stupidity

• AI researchers / developers want strongest possible AI

• Gamers want believable AI

• Player is supposed to win!
  – Winning because AI gives up is not as rewarding
Loosing Gracefully Techniques (1)

• Warn the player about an attack / be visible (Especially in action-adventure games)
  – Shout “Take that!” before attacking
  – Great camouflage makes for bad gameplay
  – Move before firing
    • Player enters room, the monster looks sideways

• Have horrible aim
  – Being killed is not pleasant for the player

• Miss the first time – aim at a close destructible object
Loosing Gracefully Techniques (2)

• “Kung-Fu” style attack
  – Only one team member attacks the player
  • Half-life:
    – One attack slot
    – When out of ammo, AI player shouts “Cover me”
    – Another player starts attacking
    – Illusion of agent communication
Loosing Gracefully Techniques (3)

• Pull back at the last minutes
  – A “boss” becomes vulnerable when players health and ammo are low

• Intended Vulnerabilities
  • Stand on mines
  • Gun misfires
  • ...

Cover Up Weakness with Design: Halo

• Rushing levels with the assault rifle isn’t fun
  – levels end too quickly.
  – **AI is more accurate** the longer the player is out of cover.

• AI isn’t very good at dealing with close quick targets
  – **Powerful melee attacks.**

• One can **refine** the design to fix flaws in AI
Cover Up Weakness with Design: Half-Life

• Player throws grenades
  – Pathfinder tries to find an escape route
  – Fails to do that for all agents
  – Standing helplessly is stupid

  • Play animation of crouching down and covering head
Explicit and Implicit AI Designs

**Explicit**
- Characters’ behaviour is predefined (Doom 3, Unreal 2,...)

**Implicit**
- Characters **work together** to create an emergent storyline (Pizza Tycoon)

**Modern games with implicit AI still have a storyline**
- GTA series
- Bioshock Infinite
AI Techniques

• We only had a look at simulation-based behaviour in this module
  – Specify rules / states / actions / perceptions
  – Let the system figure out what to do next

• Alternative: scripted behaviour
  – Agents follow some predefined behaviour
Scripted Behaviour

• Game designers decides what computer characters do
  – Fixed trigger regions
    • When player approaches, character starts talking
  – Scripts send units to attack at some time
Scripts

• Technique of specifying a game’s logic outside the game’s source language
  – Scripting languages

• These two notions are closely interlinked
  – If the behaviour is specified by designers, they need a way to access it
If (PlayerArmed == TRUE) 
    BEGIN 
        DoFlee(); 
    END 
ELSE 
    BEGIN 
        DoAttack(); 
    END
Verbal Interaction

If (PlayerArmed == Dagger)
    Say("What a cute little knife.");
If (PlayerArmed == Bow)
    Say("Drop the bow now and I'll let you live.");
If (PlayerArmed == Sword)
    Say("That sword will fit nicely in my collection.");
If (PlayerArmed == BattleAxe)
    Say("You're too weak to wield that battle axe.");
Scripting Events

If (PlayerLocation(120,76))
    Trigger(kExplosionTrap);
if (PlayerLocation(56,16))
    Trigger(kPoisonTrap);

If (PlayerLocation(kDoorway))
    PlaySound(kCreakingDoorSnd);
Advantages of Scripted Behaviour

• Faster / parallel game code development
• Easier to write and modify
• Much easier to execute
  – No search, no simulation
    • No pathfinding?
  – Simple execution of the script
• Possibility to create mods (PC)
  – Selling point long past the release date
Disadvantages

• Limits player’s choices
• Allows to exploit AI flaws
  – Players will learn the limits of the script
• Non-programmers are required to program

• To be interesting, games need LOTS of scripts
Best of Both Worlds(?)

- Combining smart agents with scripted behaviour
- FSMs as scripts
  - Game design & AI design done by the same people
  - Enforced transitions based on the storyline
- Override the default behaviour of characters
- *Bind* agents and objects
In Place of a Conclusion: Game AI Techniques (1)

- Agents and multiagent systems
- A* pathfinding
- Behaviour trees
- Blackboard architectures
  - Coordination method
- Command hierarchy
  - Taking decisions on different levels
In Place of a Conclusion: Game AI Techniques (2)

- Dead reckoning
  - Predicting a player’s future position
- Decision trees
- Emergent behaviour
  - Behaviour that was not explicitly programmed
- Flocking
- Formations
  - Group movements
In Place of a Conclusion: Game AI Techniques (3)

• Fuzzy logic
  – Yes / no → degree of (un)certainty

• Goal oriented behaviour

• Influence mapping
  – RTS games: how valuable a tile is

• Learning

• Level of detail AI
In Place of a Conclusion: Game AI Techniques (4)

• Markov systems
  – Uncertainty as probability. Markov FSM & Markov processes

• Minimax

• Rule-based systems

• Scripting

• State Machines (FSM, HFSM, Stack FSM)

• Steering
In Place of a Conclusion: Game AI Techniques (5)

- Subsumption architecture
  - Several layers of FSM, highest layer has priority
- Tactical and strategic AI
  - Global plans on top of short-sited goals
- Terrain analysis
  - Identify strategic locations
- Trigger system