AI in Computer Games

Who am I?
- Lecturer at Uppsala University, Dept. of information technology
- AI, machine learning and natural computation
- Hobby: Gamer and game AI-analyst since 1980

Goals
- Games are entertainment!
- Important that things behave naturally
  - not necessarily perfect
  - "things" are not always creatures
- Follow (the game's) natural laws
  - and avoid cheating
- Characters should be aware

Game A(I?)
- Academic AI is usually concerned with making rational decisions
  - Searching for the optimal solution
- Game AI is more often about
  - Artificial Life
  - Believable behaviour
    - including realistic physics
  - Game balancing
    - challenging, but not unbeatable opponents

Game categories
- Role Playing Games (RPG, MMORPG)
- First Person (Third Person) Shooters (FPS/TPS)
- Strategy games (RTS, DTS, 4X)
- Sports games
- Simulation games
- Adventure games
- Classic strategy games
- Fighting games
- ...
**History -1980**

- **1960's**
  - First computer games
    - SpaceWar! (PDP1, for two human players) (1962)
    - Board games (e.g. chess) against the machine
  - 1970's
    - Pong (early arcade game) (1972)
    - Computer controlled opponent in arcade games
    - Space Invaders (1978)
    - Predefined patterns, no awareness
    - "AI" takes 1-2% of CPU

- **1980's**
  - Pac-Man (1980)
    - aware opponents with personality
  - A computer beats a master chess player (1983)
  - First fighting games
  - Adventure games
    - Dungeon, Zork, ...
  - First MORPG (MUD)

- **1990's**
  - FPS and RTS games
  - Games about/with evolution and learning (Creatures, Black&White)
  - Deep Blue beats Kasparov (1997)
  - Graphic cards take the load off the CPU
  - AI takes 10-35% of CPU

- **2000-**
  - Computer games is a big industry
    - Games sell for about 25 billion USD per year
    - Market grows with 16% per year
    - A game project: 2 years, 8-15 million USD
  - Less cheating in AI
  - Characters are more aware
  - Characters collaborate better
  - Focus shift from graphics towards AI
    - Large part of the code is AI code (often made from scratch for each game)

**Typical Game AI topics**

- Strategical/tactical decisions
  - Against or with you
  - Search for best counter action
  - adaptivity
- Director level AI
- Simulation
  - of natural behaviour
  - for animation (e.g. bird flocks)
- Shortest path problems

**Why is Game AI hard?**

(what makes them interesting)

- Huge state space
- Huge action space
- Multiple tasks
  - on different levels of abstraction
  - of different types
- Non-deterministic
  - makes planning difficult
    - post-conditions difficult to set
- Often real time
Some methods
- Minimax
  - Logic games, search for best counter action
- Finite State Machines (FSM)
- Behaviour
- A*
  - For shortest path problems
- Particle methods
  - Simulation of flocks, smoke, water, grass,...
- Smart terrain

Minimax (counter actions)

Finite State Machines
Pacman ghost (red)

Reinforcement Learning

Best ≠ shortest
**Smart terrain**
- Store knowledge in objects instead of in the characters
  - 🍵 drink me! ➔ not thirsty, warm
- Easier to know what is relevant
- Easier to add new objects later
- Attributed to Will Wright (Sims)

**Thoughts on learning**
- Game characters are short lived
- Learning requires many attempts
  - **Keep it simple!**
- Probabilistic methods (Menace)
- Evolutionary methods
  - genetic algorithms and PSO
- Neural networks
  - in game development, but not in the game

**Thoughts on learning**

**AI in various game types**
- Board games
- Role playing games
- Strategy games
- Platform and sports games
- Racing games

**Board games**
- Discrete time / turn based
- Often deterministic
- AI is in the opponent
- AI goal is non-typical (for games)
  - usually strives for optimality
- Tree search
- Library
- Reinforcement learning

**Role Playing and Adventure**
- AI in enemies, bosses, party members and other NPCs, ...
- Scripting, FSMs, Messaging
- Role Playing ≠ Combat
  - combat oriented games are simpler to make
- Conversations (grammar machines)
- Quest generators
- Towns
Town behaviour

- Need-based system
  - Needs (e.g. hunger, business, ...)
  - Actions (e.g. eating, trading, ...)
  - "Need pathfinding"
- Problems
  - Finding people
  - Unwanted interaction between NPCs

Strategy games

- AI heavy (on both sides)
- Shortest path problems
- Strategical decisions
- Tactical decisions
- Town building and resource management
  - planning
- Indigenous life
- Reconnaissance (fog-of-war)
- Diplomacy
- Know thy enemy (observe and adapt)

Action games (FPS, TPS)

- Enemies
- Cooperative agents
- Weapons
- Attention
  - requires perception
    - requires a good physics engine
- Pathfinding
- Spatial reasoning
- Anticipation
Platforms and sports

- Platform games
  - Since 1996 (Mario 64) in 3D
  - Camera problems
- Sports games
  - Camera problems (harder)
  - Cooperation
  - Game balance can be difficult
  - Learning

Platforms and sports

- Platform games
  - Since 1996 (Mario 64) in 3D
  - Camera problems
- Sports games
  - Camera problems (harder)
  - Cooperation
  - Game balance can be difficult
  - Learning

Racing games

- Track AI
- Traffic (including pedestrians)
- Physics
- Tuning NPCs and vehicle parameters
  - Genetic algorithms
  - Particle swarm optimization

Racing games

- Track AI
- Traffic (including pedestrians)
- Physics
- Tuning NPCs and vehicle parameters
  - Genetic algorithms
  - Particle swarm optimization

Conclusion

- Making realistic games requires more than good graphics
- Computer controlled characters must behave
  - Naturally
  - Reasonably intelligent, without cheating!
- Graphics has dedicated hardware
  - More processing power available to AI
- In the future
  - Dedicated AI cards?
  - Combined AI/Physics/Graphics cards?
  - Multicore processors
  - Knowledge transfer from games to robotics

Robocup (Aibo league)