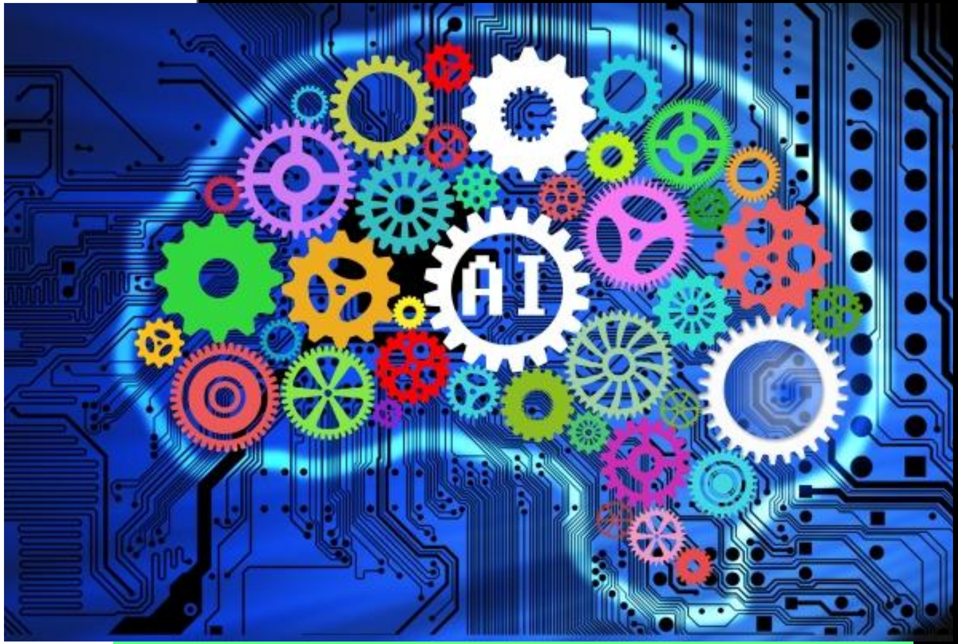
Artificial Intelligence In Games

START

Summary

- Artificial Intelligence for Games
- Game agents
- Sensing
- Thinking
- Acting
- Communicating
- Relevant techniques

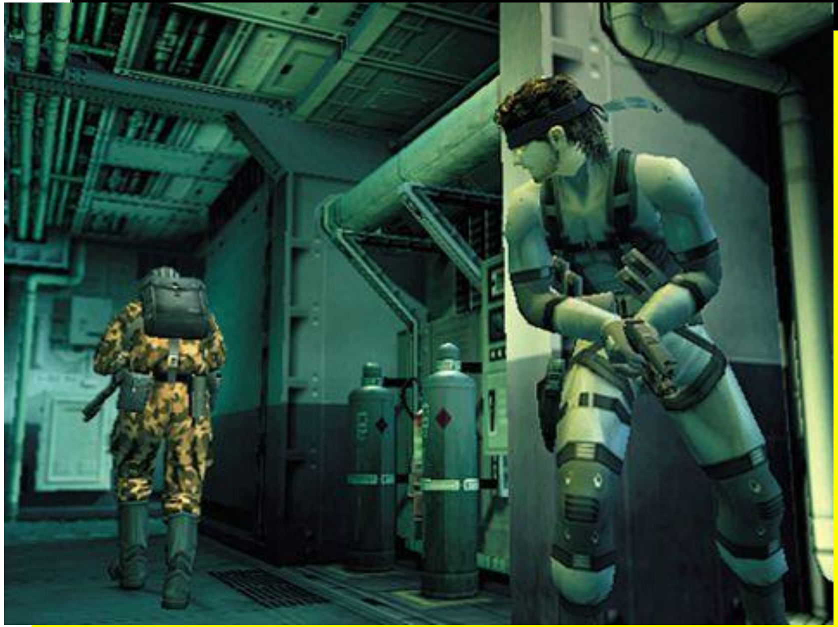


https://threatpost.com/security-and-artificial-intelligence-hype-vs-reality/136837/

AI for Games

Common base types and structures

- Part of the experience is challenge
- One way to challenge:
  - a virtual opponent able to simulate a real opponent
- Artificial Intelligence (AI)
  - study of simulating intelligent behaviors
- Perfect, human-level AI is not available (yet?)
- But for specific domains (e.g. games)
  - possible to achieve reasonable and believable results



Metal Gear Solid 2: Sons of Liberty

x3

AI for Games

Must be good, but not perfect

- Opponents: challenging but balanced,
- Keep the game entertaining and fun even when losing

Must have no unintended weaknesses

- To position and animate objects in the world

Must perform within an acceptable rate

- AI reactions should be real time
- However AI typically only gets between 10% to 20% of frame time

Must be configurable by game designers/players

- Designers must be able to easily tune it and extend AI,
- Players should be able to tweak it in moddable games
- Scripting is key

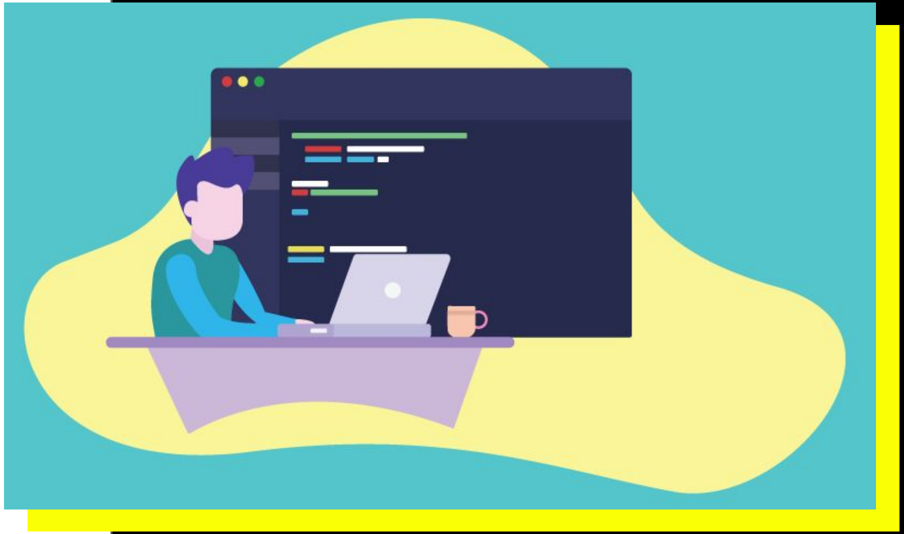
Must not stop the game from shipping

- AI should not put the game at risk
- Experiments should be done early, and tested thoroughly

x4

Developer Specialization

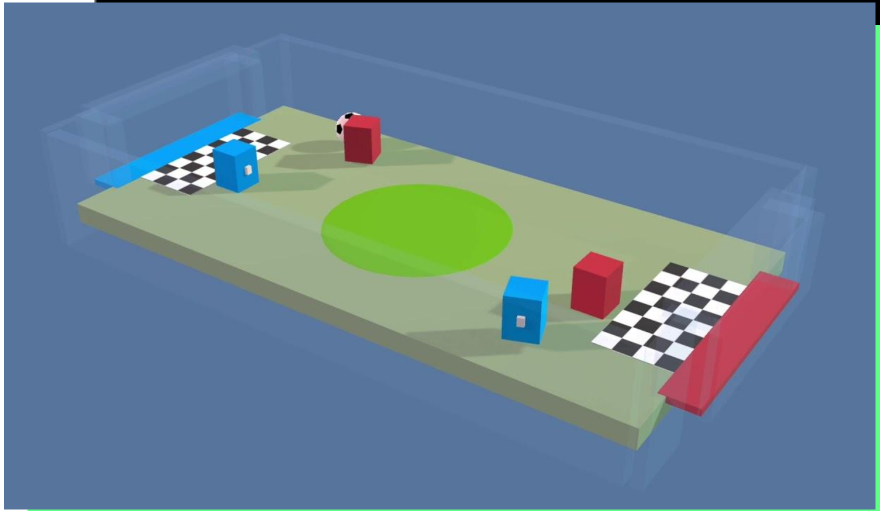
- AI has become highly elaborate
- Required skills may vary from genre to genre
    - e.g. strategy vs FPS
- AI for RTS are particularly demanding
- Scripting is moving AI closer to designers



https://dribbble.com/shots/4605995-Web-Developer-Illustration

Game Agents

- An intelligent agent, usually a non-player character (NPC)
- Can be an opponent, an ally, or a neutral entity
- Three key steps in loop:
  - Sense - Perceive the game world
  - Think - Evaluate and decide actions
  - Act – (Re)act in a seemingly intelligent manner
- Optional: learning or remembering to adapt to the player

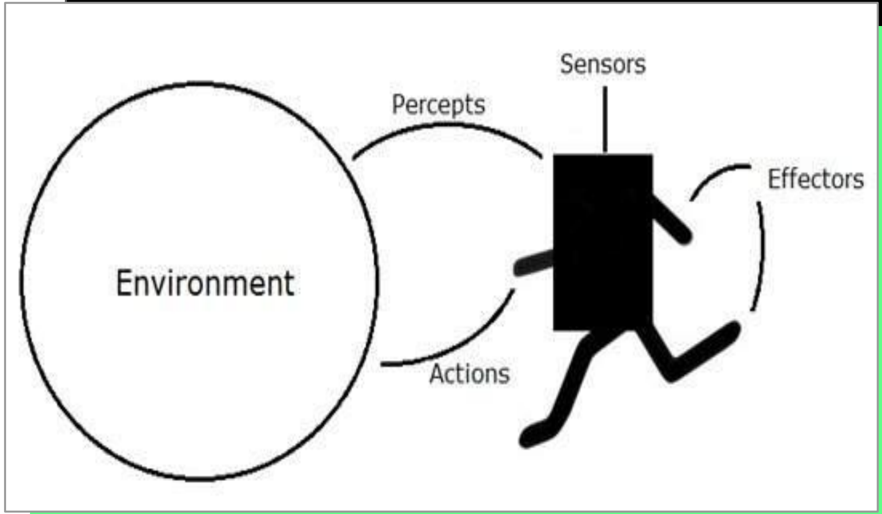


Unity ML Agents:
https://unity3d.com/how-to/unity-machine-learning-agents

x6

Sensing

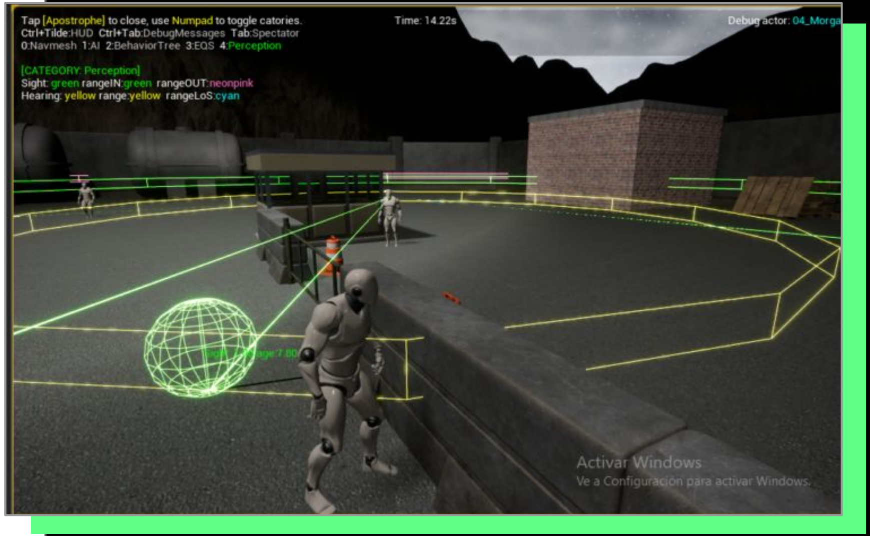
- Game agent must sense the world to react intelligently to it
  - just as a regular player
- It could have access to perfect information about the game state…
  - e.g. opponent position, level map, etc.
- …But using all that could be "cheating“…
- …Thus game agents must be limited so that they are "fair opponents"
  - e.g. cannot “see through walls“, do not know what weapons you have



x7

Sensing - Vision

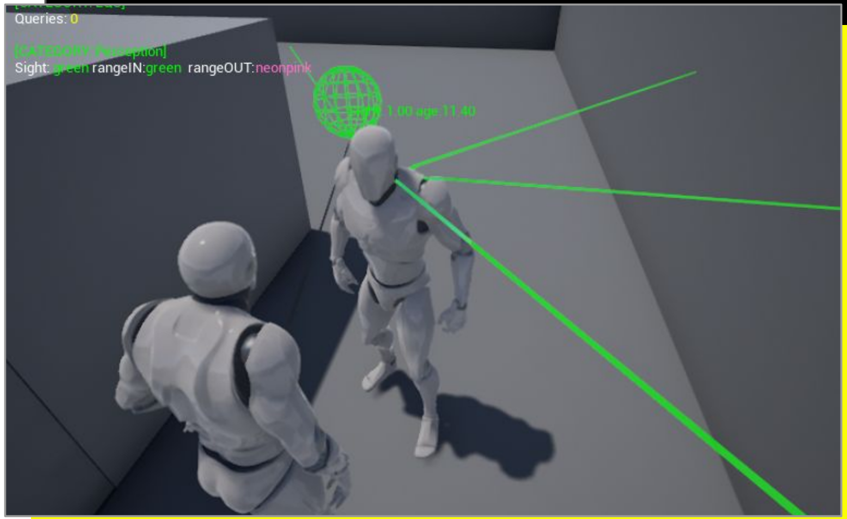
- Necessary to determine which objects should be visible for an agent
- Can be time-consuming, therefore it is limited to some agents, and with a low update rate
- Typically computed by:
    - getting a list of relevant objects,
    - computing a "line-of-sight" vector
    - checking for distance, frustum and occlusion (this is last as it is the most expensive)
- Movement-sensitive vision
    - from a distance, moving objects are more noticeable than stationary ones
- Special areas such as hide-outs or high-risk areas can be marked as objects to be searched



AI Sight Perception to Custom Points:
https://blog.gamedev.tv/ai-sight-perception-to-custom-points/

Sensing - Hearing

- Fast steps, firing, dropping objects may attract attention of enemies due to sound
  - even if they are not in line of sight
- Hearing is commonly simulated using events
  - an action that makes a noise will generate an event that will be sent to all agents within a given range
- Range computation is simplified to closed areas and ranges
  - E.g. a noise originating in area X will be heard by all agents in X within 10 meters of the sound source



Thinking

- After gathering all information, the agent must evaluate it and decide how to act/react
  - This is by many considered the "true" AI
- Two main ways of deciding:
  - Pre-coded expert knowledge (e.g. sets of rules associated with randomness)
  - search algorithms (for finding near-optimal solutions)
- Machine learning techniques such as neural networks or genetic algorithms are interesting…
  - … but their use is cautious, due to the computational cost compared to simpler techniques

https://health.usf.edu/is/blog/2018/06/05/ai-awakening

Thinking - Expert Knowledge

- Common techniques for encoding expert knowledge
  - finite-state machines (most used)
  - decision trees
  - logical inference…
- Example
  - If the enemy you see is weak, attack the enemy;
  - otherwise, run away and get backup support.
- Drawbacks
  - Not a complete solution, nor nicely scalable
  - Relies on testers to uncover bugs
- Typically used in narrow domains,
  - Usually sufficient for many cases of acceptable scale

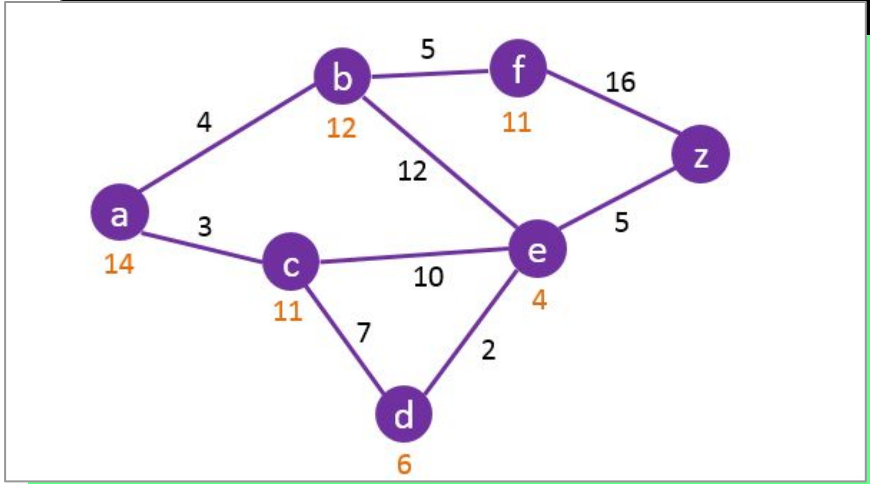
https://health.usf.edu/is/blog/2018/06/05/ai-awakening

Search algorithms

Given a set of possible moves and associated rules, it is possible to create algorithms that search through those moves for a (near-) optimal solution - the plan

Examples:

- board games
- path finding in FPS's (e.g. A*)



Acting

- After sensing and deciding, the agent needs to (re)act
- Actions should be noticeable by the player
    - even if not directly/immediately
- Examples:
    - change location, play animation, fire a weapon, interact with a player
- The way these are carried influences the player's perception of the agent's intelligence
- Thus increasing complexity and variety of expressions of agent's intelligence and actions such as e.g. animations

L.A. Noire, 2011

Acting - Reaction Times

- Although agents could sense, communicate and act almost instantly,
  - it would be unintuitive and wrong to do so
- It is necessary to introduce delays to simulate different reaction times
  - e.g. seeing is faster than communicating via speech
- Reaction times help to avoid flip-flopping between states
  - i.e. switching repeatedly between two states due to constant re-evaluation
  - The decisions and actions must sustain for some time

L.A. Noire, 2011

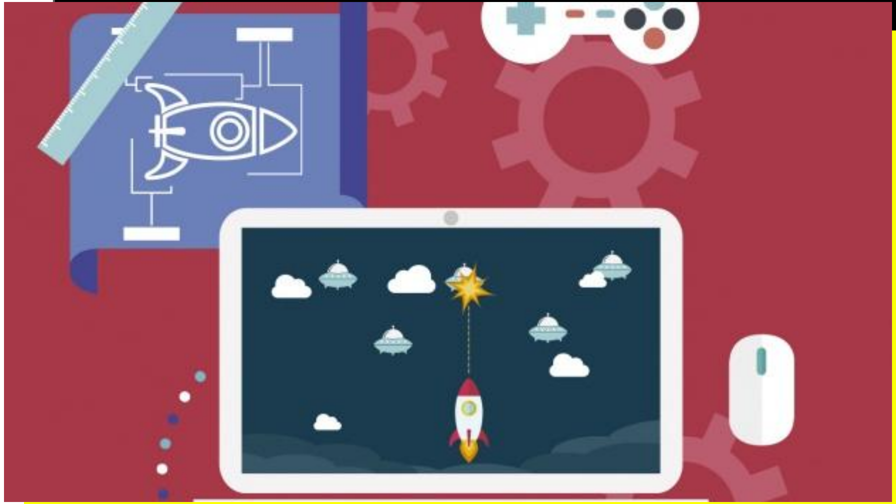
Communicating

- In many cases, agents are expected to communicate with each other
  - Can be seen as a mix of acting and sensing
- It is necessary to model the transfer of information between agents
- Communication is typically implemented in an event-driven way
  - as with hearing

GTA V, 2013

Some Relevant Techniques

- **Command hierarchy**
  - varying levels of decisions (e.g overall strategy, group tactics, individual actions)
- **Emergent behavior**
  - Resulting from implicitly combining low-level rules, e.g. Seek food, avoid walls, follow light; flocking is a common instance
- **Manager Task Assignment**
  - A manager has a list of tasks and assigns them to the most suitable agents (like a soccer coach)
- **Behavior trees**
  - Similar to a hierarchical finite-state machine

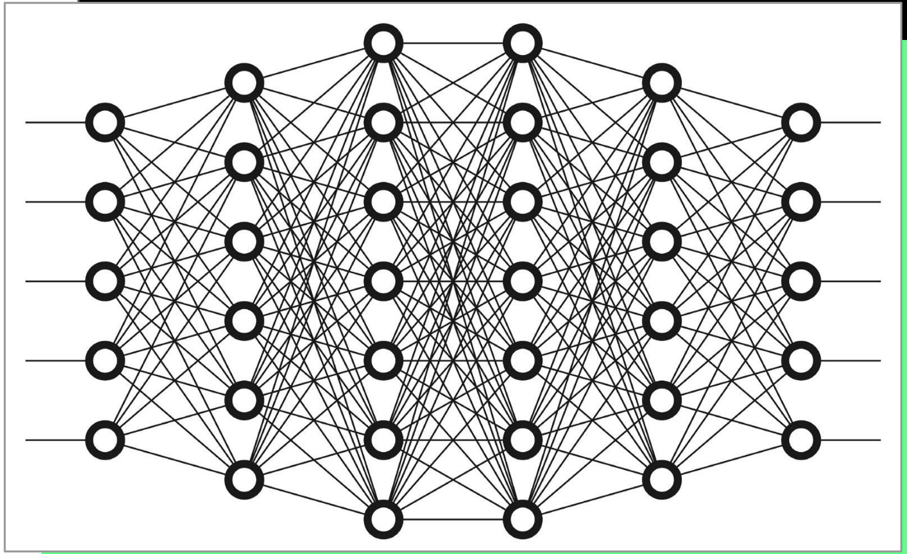


Prediction

- **Dead Reckoning**
  - For determining future positions of objects/opponents
- **N-gram Statistical prediction**
  - For predicting next opponent actions based on previous action sequences
- **Player modelling**
  - For assessing the player's profile and adjust playability accordingly

Machine Learning

- Bayesian networks
- Decision tree learning
- Neural networks
- Perceptrons
- Reinforcement learning
- Reputation systems



Artificial Intelligence In Games