Goal Oriented Action Planning AI

PERSONAL PROGRAMMING PROJECT

BY JORDAN MARTIN

What is Goal Oriented Action Planning?

Opposed to Traditional AI Techniques

- Navigation Meshes, Finite State Machines, Behavior Trees

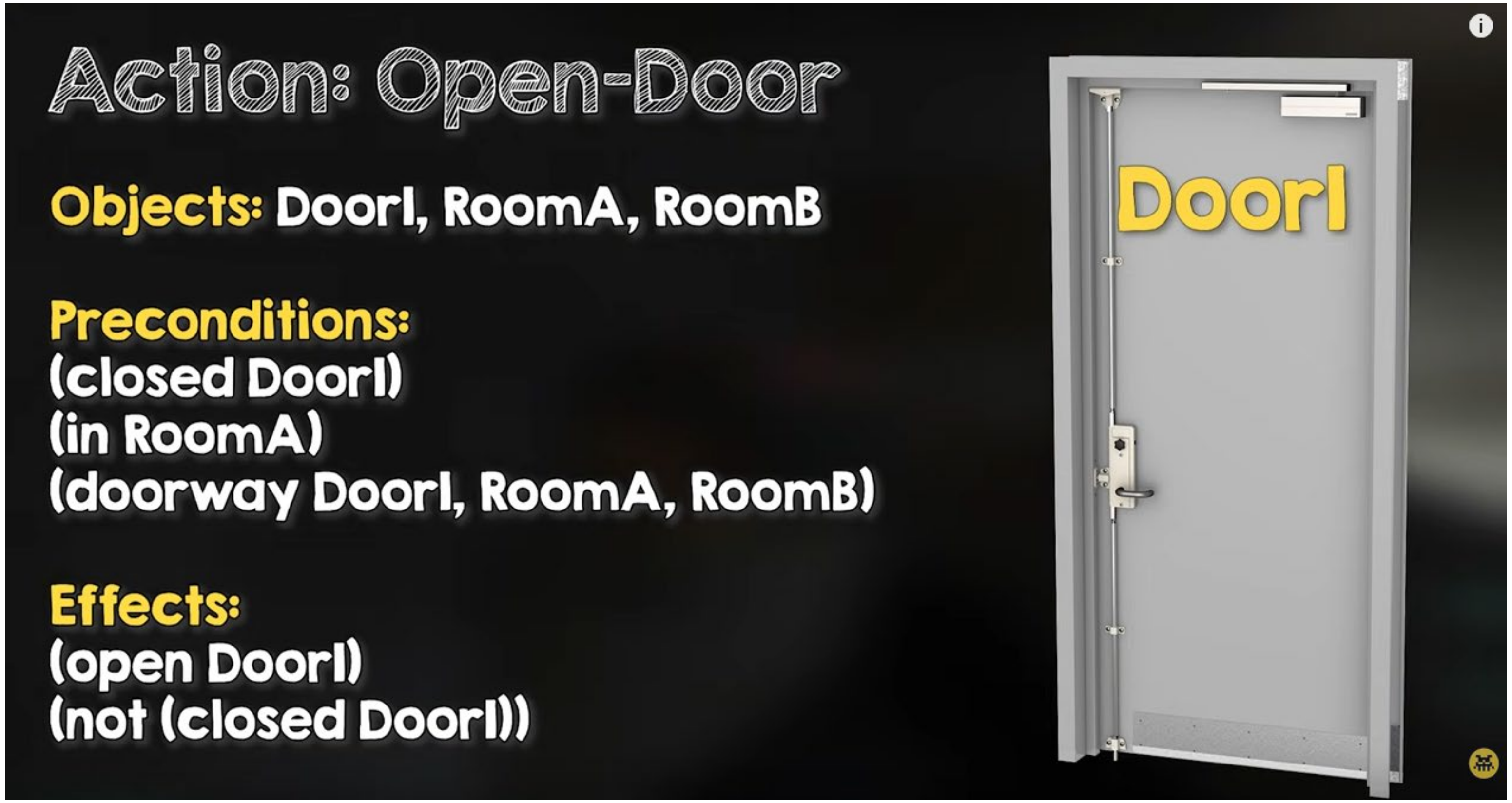
Based on Automated Planning

- A system attempts to figure out a sequence of actions that will achieve a distant “goal”
- Queries the world state for facts
- Creates a plan from a set of actions

Building an action

- Objects – items involved in the action
- Preconditions – facts that must be true for the action to work
- Effects – How the world state changes when the action is complete

Example



Action: Open-Door

Objects: Door1, RoomA, RoomB

Preconditions:
(closed Door1)
(in RoomA)
(doorway Door1, RoomA, RoomB)

Effects:
(open Door1)
(not (closed Door1))

From: Building the AI of F.E.A.R. with Goal Oriented Action Planning

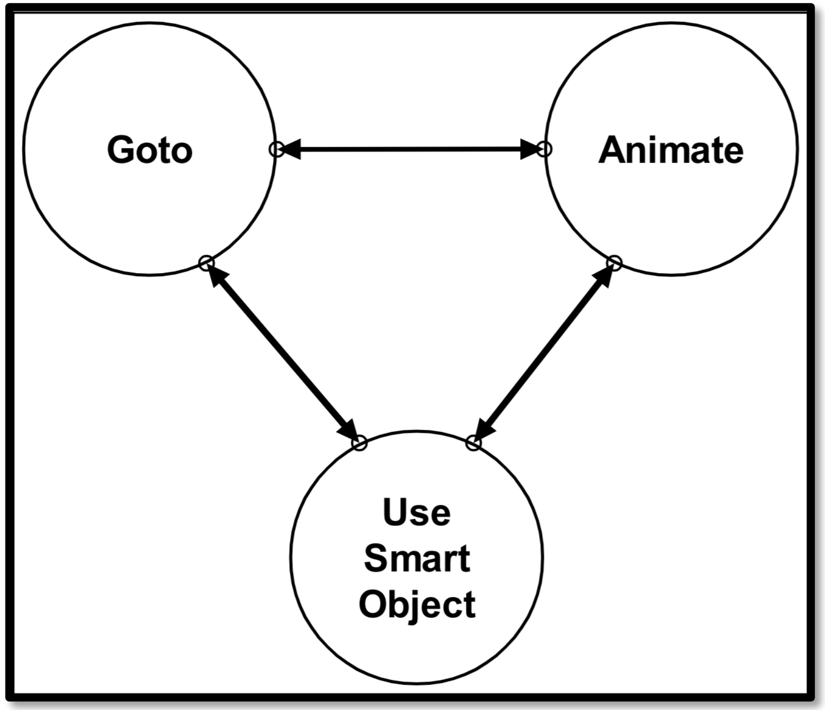
Pros and Cons

Pros

- Smaller, data driven approach
- Priority of multiple goals
- “Smarter” AI

Cons

- Revalidation of plan during execution
    - Actions also need to be validated
- Planner makes all decisions
    - AI are not aware of each other
- Performance overhead
    - AI need to be given a goal at all times



State of the Art

F.E.A.R.

- Pioneered using GOAP for FPS Enemy AI
- Research publications and source code available

Other games that use GOAP include:

- Condemned, Stalker, Just Cause, Deus Ex Human Revolution, Tomb Raider (2013), Middle Earth: Shadow of War

Evolved into High Task Network Planning

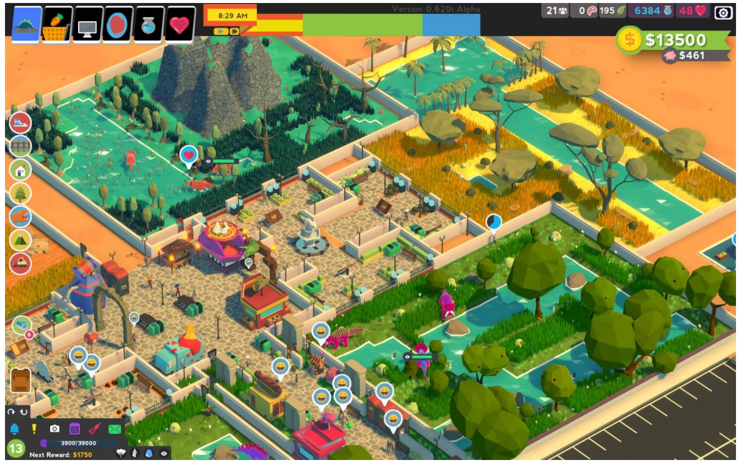
- Allows designers to place action costs on a per-character basis
- Used in Transformers: Fall of Cybertron

Objectives

- Create a GOAP AI system in Unity
- Integrate the GOAP AI system into an agent-based simulation
- Analyze the results and explore usage in games

Stretch Goals:

- Incorporate into a tycoon game



Schedule

Creating the GOAP AI System	Week 1	**Proposal Presentation** Further Research Agent simulation moving with Basic NavMeshes
	Week 2	Creating the Environment • World State, Actions, Agents
	Week 3	Creating the Planner • Executing a simple plan on an agent
	Week 4	Expanding on the Planner • Executing multi-step plans on multiple agents
	Week 5	Monitoring Agents State in Real Time • Debug tool to give details through UI
Integrating GOAP System into Agent-Based Simulation	Week 6	**Update Presentation** Adding "Smart Objects" to the world • Add objects that can be used to fulfill goals
	Week 7	Revalidation of plans • Add changes to the world state that require plan changes
	Week 8	Priority of Goals for agents • Execute plans while having competing priorities
Applications towards Games	Week 9	Stretch Goals: Create a tycoon game • Add player agency to the game and have planner adjust
	Week 10	Stretch Goals: Create a tycoon game • UI, Resources, Placement of Buildings, Game Logic
	Week 11	**Final Presentations**

Resources

Jeff Orkin - http://alumni.media.mit.edu/~jorkin/goap.html

- Implemented GOAP for F.E.A.R
- Resources, Implementations, Source Code, Books

STRIPS - http://ai.stanford.edu/users/nilsson/OnlinePubs-Nils/PublishedPapers/strips.pdf

- Used to abstract AI systems in F.E.A.R.

Advanced AI For Games with Goal-Oriented Action Planning Udemy Course - https://www.udemy.com/course/ai_with_goap/

- Tutorial on creating a GOAP simulation in Unity

Thank you!