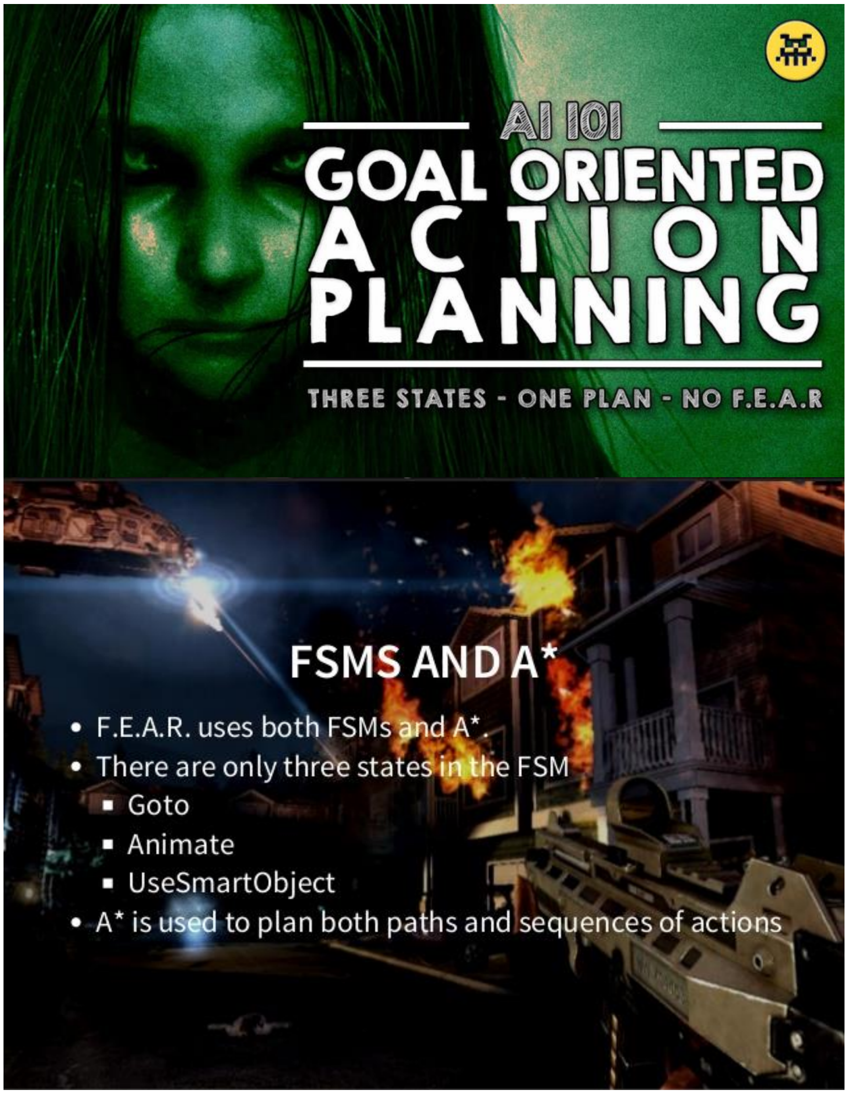

AI 101
GOAL ORIENTED ACTION PLANNING
THREE STATES - ONE PLAN - NO F.E.A.R

FSMS AND A*

- F.E.A.R. uses both FSMs and A*.
- There are only three states in the FSM
  - Goto
  - Animate
  - UseSmartObject
- A* is used to plan both paths and sequences of actions

”https://i.ytimg.com/vi/PaOLBOuyswI/maxresdefault.jpg , https://www.slideshare.net/lp4390a/fear-47151147”

Goal Oriented Action Planning (G.O.A.P) vs Finite State Machine (F.S.M)

COS30002 RESEARCH DOCUMENT

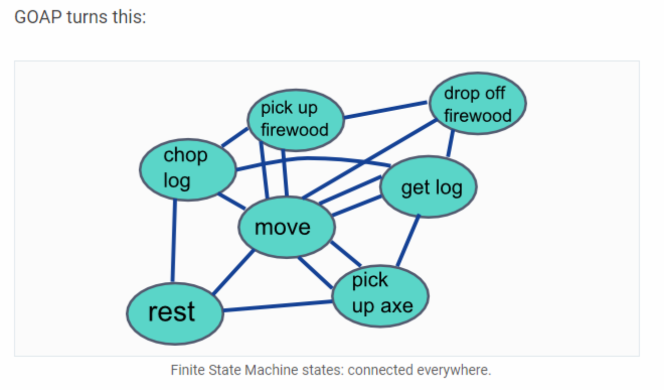
Hasaan Akhtar | AI for Games | 17/06/2020

Contents

INTRODUCTION TO G.O.A.P & F.S.M?

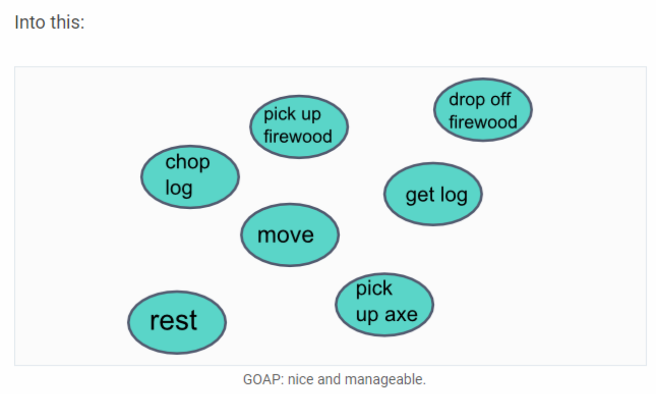
Goal oriented action planning is an AI scheme that takes into consideration several action plan sequence and opts the most suitable sequence that takes into consideration the long-term effect it could have on character goals, it satisfies the actions taken to complete the goal without having long and complex state transition machine. On the other hand, FSM depicts the cyclic behavior of transitioning between stages to achieve the outcome, which can be become quite complex based on adding more transitions as it offers poor scalability. GOAP uses a planner to select the most suitable action, every action in GOAP has an action which has a precondition and an effect that marks how the state of agent leaves the world after it has occurred whereas FSM has predefined conditional transitions between states.

An example of how GOAP can replace FSM



GOAP turns this:

Finite State Machine states: connected everywhere.



Into this:

GOAP: nice and manageable.

“https://gamedevelopment.tutsplus.com/tutorials/goal-oriented-action-planning-for-a-smarter-ai--cms-20793”

PROBLEMS WITH GOAP AND FSM

FSM being hierarchical follows certain step of state transitioning can be depicted as independent state machines, The problem with FSM is that if you want to change the execution order of state, you would also have to change the transitions between the states. The more number of states you add the difficult it gets to set up transitioning between them whereas With GOAP, you decouple the actions and then focus on each action individually, which in turn results in easier scalability and modular design for testing and maintenance, this flexibility offered by GOAP makes adding actions easier.

On the other hand, GOAP has downfalls as well, like in the earlier example above removing connections could lead to much more worse situation where AI can practically be broken and would require some sort of input to work, for instance the NPC moves to Chop the log but there is no animation for Axe and that might create a problem. Unlike FSM, GOAP uses more processing power to search through the number of possibilities using the A* Searching algorithm, so the bigger the possibilities the more the searching progresses which in turns uses more processing power. The problem with GOAP is that it allows less direct control over AI’s actions which at first would save you from trouble but later could cause problems like leading into a loop following repeated several behaviors which could break the NPC AI. GOAP, not preferably for fast-paced games due to its processing usage.

Some examples of games with GOAP and FSM

- FEAR uses 3 state model and where actually relies on the Action Planning aka GOAP while the rest still works by utilizing the FSM.
- Half-Life 2 also followed the GOAP which the only exception that each state in the FSM had definite set of actions that enemy AI could take.
- Quake III used FSM where most of the actions were carried by calls made back and forth through transitioning to invoke specialized methods for the action to be performed.

Other game examples to consider which utilize GOAP.

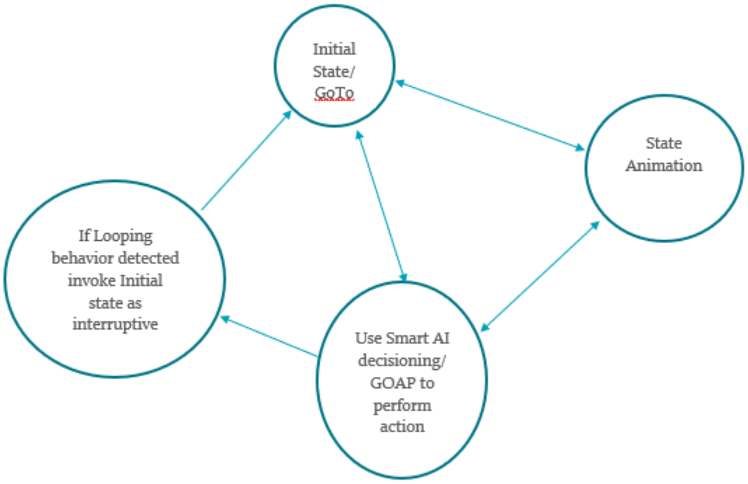
- S.T.A.L.K.E.R
- Fallout 3
- Just Cause 2

By considering these examples it is in fact clear that using FSM is simply better when avoiding loop holes, as it allows AI to formulate new sets of action by interrupting plans, this works if all planning activities are linked in one state where the game can later force a transition to another state if the earlier action in the plan failed which in turn lead NPC to devise a new plan.

WHAT NOW? WHICH IS BETTER? WHAT DO WE IMPLEMENT?

The answer is pretty obvious "It depends upon the game itself", like I stated above simple game are good with FSM but if number of state increases it becomes more complex, similarly with GOAP the problem lies with best action which might become a repeated sequence of actions taken which in turn could break the AI, plus in regards to performance it would be worse than Non Player since the AI would be super hard for the player to deal with.

So honestly, there is no such answer but with the amount of gameplays and theories I could find online, it is in fact better to combine the behavior of both like with F.E.A.R as it used both FSM and GOAP in a more practical approach to dealing with looping sequence of action and with the fact that FSM shouldn't be made complex while also utilizing the emergent gameplay behavior of GOAP with better decisions for the NPC and the enemy to take in order to look more alive for the player to interact with.


Initial State/ GoTo
State Animation
If Looping behavior detected invoke Initial state as interruptive
Use Smart AI decisioning/ GOAP to perform action

The way this methodology could work is if we used FSM as an initial state for the game transition and then use GOAP for AI smart decisions while also utilizing the interruptive behavior of FSM to create more action plans in order to avoid loop holes where possibly our AI could have been stuck and broke, the below illustration describes the possible processes that the F.E.A.R game may have used based on my understanding of the game methodology.

Highly recommended you check this out: https://www.slideshare.net/lp4390a/fear-47151147

Conclusion

GOAP takes up more processing but gives you a set of suitable actions while it may create loop but those loop can be interrupted by use of FSM, while some may argue that it can all be done with FSM but the problem like it would require more modular programming for every transition and in the end more complexity for game tester to dive into.

So, Our game would have a small FSM for the GOAP to work in, the reason why it is more practical is because it can vary with player proximity from target and GOAP can be better utilized for its better action planning delivering sets of actions with effects into consideration that can avoid complexities to deal with instead of ridiculously adding more states in FSM. Hence it make an AI more Smart, dynamic and flexible while not creating a gameplay where the AI is unbeatable since that would just ruin the whole fun of the game if player cannot beat it because of its hardened best actions delivered by GOAP whiling looping sequence of actions.

References used for research

Renaldas Zioma (2016). *Practical AI in Games*. [online] Available at: https://www.slideshare.net/RenaldasZioma/practical-ai-in-games [Accessed 17 Jun. 2020].
Houtkooper, A. and Aldershoff, F. (n.d.). [online] Available at: https://www.computing.dcu.ie/~gmdbwenniger/Finished_Projects/GOAP-Report.pdf.
Jared Mitchell. (2015). *Goal-Oriented Action Planning Research*. [online] Available at: https://jaredemitchell.com/goal-oriented-action-planning-research/ [Accessed 17 Jun. 2020].
ElProCus - Electronic Projects for Engineering Students. (2019). *Finite State Machine (FSM) : Types, Properties, Diagram, Design and Applications*. [online]

Available at: https://www.elprocus.com/finite-state-machine-mealy-state-machine-and-moore-state-machine/.

García, J. (2018). *Advantages & Disadvantages of Finite State Machines: Switch-cases, C/C++ Pointers & Lookup Tables (Part II)*. [online] Ubidots Blog. Available at: https://ubidots.com/blog/advantages_and_disadvantages_of_finite_state_machines/ [Accessed 17 Jun. 2020].

Chaudhari, V. (2017). *Goal Oriented Action Planning*. [online] Medium. Available at: https://medium.com/@vedantchaudhari/goal-oriented-action-planning-34035ed40d0b [Accessed 17 Jun. 2020].

www.davideaversa.it. (n.d.). *Choosing between Behavior Tree and GOAP (Planning) · Davide Aversa*. [online] Available at: https://www.davideaversa.it/blog/choosing-behavior-tree-goap-planning/ [Accessed 17 Jun. 2020].

Aakash Chotrani (2018). *What is goap, and why is it not already mainstream*. [online] Available at: https://www.slideshare.net/AakashChotrani/what-is-goap-and-why-is-it-not-already-mainstream [Accessed 17 Jun. 2020].

reddit. (n.d.). *r/gamedesign - [AI Design] What is the difference between Hierarchical Finite State Machine and Goal Oriented Action Planning?* [online] Available at: https://www.reddit.com/r/gamedesign/comments/bjh58u/ai_design_what_is_the_difference_between/ [Accessed 17 Jun. 2020].

Game Development Envato Tuts+. (n.d.). *Goal Oriented Action Planning for a Smarter AI*. [online] Available at: https://gamedevelopment.tutsplus.com/tutorials/goal-oriented-action-planning-for-a-smarter-ai--cms-20793 [Accessed 17 Jun. 2020].