AUTONOMOUS AGENTS IN VIDEOGAMES

THANNIA BLANCHET

INTRODUCTION

Nowadays, contrary to what many may believe, the videogame industry is one of the most important industries in the world, it has surpassed the film industry as the leading one in the market.

It’s not a matter only of economy, though, it is the area of human life in which artificial intelligence is applied the most, a regular person may not be in constant contact with a robot or some other kind of system, but most of us deal with computer games on a regular basis. And games are growing more and more diverse with time.

Both video games and A.I. research can benefit from each other, videogames offer an environment to test new forms of artificial intelligence, and to develop such. And as artificial intelligence gets better, it offers a more realistic experience for the player.

Videogames have far grown up from having looping enemies and bosses that repeat the same pattern of attacks, as they have evolved, and different genres have expanded, we now have the need for a balance between the intelligence of enemies, since too smart may make the game impossible and too dumb makes it boring, we need non player characters to help us on our quests, friendly machine partners that take the place of human players in cooperative games, and even minions to gather resources for us. This could only take place with the use of autonomous agents.

Not only is Artificial Intelligence used to emulate characters, but other parts of the games, like automatically setting up a match, need of a level of artificial intelligence to work properly. This being said, videogames indeed prove to be a very large area where A.I. research can expand and evolve.
WHAT IS AN AGENT?

Agents can be defined to be autonomous, problem solving computational entities capable of solving effective operations in dynamic and open environments.

A software agent is a computer program that acts for a user or other program with the authority to decide which, if any, action is appropriate.

Agents are more autonomous than objects.
Agents have flexible behavior, reactive, proactive, social.
Agents have at least one thread of control but may have more.
Agents are coupled to their environment.

Agents can be classified in different kinds:

<table>
<thead>
<tr>
<th>Property</th>
<th>Other Names</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>reactive</td>
<td>(sensing and acting)</td>
<td>responds in a timely fashion to changes in the environment</td>
</tr>
<tr>
<td>autonomous</td>
<td></td>
<td>exercises control over its own actions</td>
</tr>
<tr>
<td>goal-oriented</td>
<td>pro-active</td>
<td>does not simply act in response to the environment</td>
</tr>
<tr>
<td>temoparally</td>
<td>purposeful</td>
<td>is a continuously running process</td>
</tr>
<tr>
<td>continuous</td>
<td></td>
<td></td>
</tr>
<tr>
<td>communicative</td>
<td>socially able</td>
<td>communicates with other agents, perhaps including people</td>
</tr>
<tr>
<td>learning</td>
<td>adaptive</td>
<td>changes its behavior based on its previous experience</td>
</tr>
<tr>
<td>mobile</td>
<td></td>
<td>able to transport itself from one machine to another</td>
</tr>
<tr>
<td>flexible</td>
<td></td>
<td>actions are not scripted</td>
</tr>
<tr>
<td>character</td>
<td></td>
<td>believable “personality” and emotional state.</td>
</tr>
</tbody>
</table>

But the ones we are interested in are autonomous agents, which can be defined as:

An autonomous agent is a system situated within and a part of an environment that senses the environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future.

All software agents are programs, but not all programs are agents.
As the description above says, an autonomous agent is a complex kind of program that more less can make decisions on its own according to the inputs it gets from the environment it is on.

To describe an autonomous agent, one must describe its:
- environment
- sensing capabilities
- actions
- drives
- action selection architecture

As one can infer now, for videogame enemies and characters, an autonomous agent is the most appropriate approach.

ARTIFICIAL INTELLIGENCE IN VIDEOGAMES

In the case of videogames, the most common type of agent is the NPC (non player character), NPCs are autonomous and goal oriented, they should carry out their own agenda. If an NPC responds to certain conditions with predefined actions scripted in a certain programming language, it would have much less autonomy compared to an agent which can plan its future actions. Furthermore, the actions that NPCs execute cause changes in the environment. NPCs can then select further actions based on the consequences of an applied action, or they can ignore the consequences of their actions. An NPC which can consider the consequences of its actions can plan. Therefore, the more an NPC can plan its own agenda, the more agent like it is, this is what makes modern videogames different from the old-school ones, in the artificial intelligence area.

The best A.I. is the one that delivers the highest entertainment value. One of the aims of Game AI is to increase the difficulty of the game to challenge the human players. The game AI should lead to a fulfilling game experience. On the other hand, games take place in dynamic complex worlds in which complex decisions must be made, often based on partial knowledge [Fairclough, Fagan et al. 2001].

The most commonly used AI methodologies to achieve game AI can be stated as follows [McGee 2005]:
- Decision trees: can be realized by If-Then statements..
- Fine state machines
- Command hierarchies
- Manager task assignment
- Path finding (A*)
- Terrain analysis
MACHINE LEARNING

When playing a multiplayer game, other opponents are fun because, as humans, they make mistakes, may do stuff against the rules, or are basically unpredictable. Accurately mimic the human versus human playing experience requires a lot of fine tuning on the programming, that is why machine learning algorithms had been being implemented for sampling data from expert players. This way the programmer has less work when hand coding the combat rules and the learned behaviors are often more unpredictable and life-like than any hard-wired finite state machine.

Some types of learning used are:

- **Supervised Learning:**
  - Learning an input-output relationship from examples
  - Tasks: regression, classification, ranking
  - Applications: skill estimation, behavioural cloning

- **Reinforced Learning:**
  - Learning policies from state-action-reward sequences
  - Tasks: control, value estimation, policy learning
  - Applications: learning to drive, learning to walk, learning to fight

- **Unsupervised Learning:**
  - Learning the underlying structure from examples
  - Tasks: clustering, manifold learning, density estimation
  - Applications: modelling motion capture data, user behaviour

PATHFINDING

Videogame agents, as being characters, usually need to find a path between two given locations on a map full of obstacles, especially on multiplayer matches, where the action is unpredictable as opposed to a story based game.

This is a challenging search problem, a solution is needed in real time, but the CPU and memory resources are often limited and there’s a huge search space to explore. Modeling realistic features can impact difficulty, with uncertainty and heterogeneous terrains and units.

Important concepts for videogame pathfinding.
- **Grid Map:**
  - Discrete search space.
Tiles are either traversable or blocked.
Straight and (optionally) diagonal moves.
- **A***:
  Best first algorithm.
  De facto standard in games industry.
- Manhattan Heuristic:
  Estimates distance from current node to target.
  Computed as if there were no obstacles.

Some instances are inherently challenging for A*, regardless of heuristic quality. This includes instances that look easy to humans. A somewhat similar behavior has been observed in AI planning domains.

Some speed-up techniques are using better admissible heuristics, problem decomposition in multi-agent pathfinding and hierarchical abstraction.

Some admissible heuristics are:
Manhattan distance:
- fast to compute
- no memory required
- good quality sometimes

Memory-based Heuristics:
- reasonably fast to compute at runtime
- use some memory
- aim at good quality

Perfect Information:
- very expensive to compute and store
- fast to access (once pre-computed)
- impractical

Landmark Heuristics:
- Family of memory-based heuristics
- landmark: location from which distance to all other nodes is precomputed

A good heuristic doesn’t necessarily guarantee fast path computation, that’s why hierarchical abstraction is used to reduce the search space to be explored.
DIFFERENT VIDEOGAME GENRES

Depending on the genre, the way Artificial Intelligence is applied might vary, machine learning for example, has been usually applied to satisfy the needs from First Person Shooters, this needs are very different from the ones of a Role Playing game, or the material gathering and building of RTS agents.

ROLE PLAYING GAMES

The intelligence required from a team or from individual characters (collaborating or cooperating) depends on how complex it is to reach to the common goal, what kind of resources the team or individuals in the team have and whether the team members have primitive or advanced collaboration strategies. There is a need for more realistic and engaging NPCs in these games [Fairclough, Fagan et al. 2001].

ACTION – ADVENTURE GAMES

In action games, having the player as a member of a squad or a team provides an opportunity for the further use of more complex AI. [Fairclough, Fagan et al. 2001].

FIRST PERSON SHOOTERS

FPS-type games usually implement the layered structure of the artificial intelligence system.

Layers at the lower levels handle the most elementary tasks, such as determining the optimal path to the target or playing appropriate sequences of character animation. The higher levels are responsible for tactical reasoning and selecting the behavior which an AI agent should assume in accordance with its present strategy. The examples for tasks of higher layers are whether the agent should patrol the area, enter combat or run through the map on search of an opponent [Grzyb 2005].

In the case of Halo 2, each character is written to do certain things, but despite their individual roles, they all function in the same way. It breaks down like this:

- The character uses its AI "senses" to perceive the world -- to detect what's going on around it.
- The AI takes the raw information that it gets based on its perception and interprets the data.
- The AI turns that interpreted data into more processed information.
- The AI makes decisions about what its actions should be based on that information.
- Then the AI figures out how it can best perform those actions to achieve the desired result based on the physical state of the world around it.
"If we were writing artificial intelligence for a robot, we would have to write all kinds of computer vision and analysis of the images to figure out what it was seeing. But because [the Halo characters] live in a simulated world, the characters can directly perceive that world. We chose to do that through simulated senses, because that way, the characters perceive the world in a way that players can reason about, because the player understands how their senses work in that world. We made simulated senses [for the AI]. So we have simulated vision, hearing and also a little bit of tactile knowledge. Where the player has five senses to deal with, and they're well developed, the AI in Halo really primarily only responds to visual input and sound input. That's because the two ways the player generally makes himself known to the AI is: The AI sees the player or they make a noise, like shooting someone."

-Chris Butcher of Bungie Studios.

REAL TIME STRATEGY GAMES

In the RTS game environment, the human and computer controlled players compete for resources. The most common setting for RTS games is the war-game setting, where in addition to resource management the player engages in war. The resource management in RTS games encompasses obtaining resources and utilizing those resources to gain an advantage. Resources are valuable minerals, energy, or other materials. Building an army and attacking strategic objectives, such as locations with access to resources, are aspects which make an RTS game a “war game”.

At the low levels of AI, path planning is important whereas at higher levels, there are modules responsible for economy, development or, very importantly, a module to analyse the game map [Grzyb 2005].

Bots for Real Time Strategy (RTS) games provide a rich challenge to implement. A bot controls a number of units that may have to navigate in a partially unknown environment, while at the same time search for enemies and coordinate attacks to fight them down. Potential fields is a technique originating from the area of robotics where it is used in controlling the navigation of robots in dynamic environments.

There are, of course, a lot more of videogame genres, each with their own A.I. needs, and even inside the same genre, depending on the game are the requirements, some might need better enemies, some might need better friends.

CONCLUSION

Videogames offer a rich environment for the development of A.I. it is the place where A. I. can best be developed, since everything happens in a simulated environment, there is no need for physical sensors or other drawbacks, it is also the place where agents can be developed to approach a more human like A.I.
BIBLIOGRAPHY

Joaquin Candela, Ralph Herbrich and Thore Graepel, Machine Learning in Games, Microsoft Research Cambridge, 2011


Luc Steels, A case study in the behavior oriented design of autonomous agents, Brussels Belgium, Vrije Universiteit Brussel, 1994


Robert Valdes, The Artificial Intelligence of Halo 2, 2004

Stan Franklin, Art Graesser, Is It an agent, or just a program?: A taxonomy for autonomous agents, University of Memphis, Memphis, TN, USA, 1997

Sule Yildirim and Sindre Berg Stene, A Survey on the Need and Use of AI in Game Agents, Rena, Norway, Hedmark University College, 2008