Andrew Phillips

471 Intro to AI

Professor Dejing Dou

Term Project Research Paper

An Exploration of Game Playing

In an article by Susan L. Epstein, a game is described as a “a noise-free, discrete space in which two or more agents (contestants) manipulate a finite set of objects (playing pieces) among a finite set of locations” (Epstein 2). In the field of artificial intelligence, the sub-field of game playing utilizes agents, which perceive from and act on the game's environment, to compete with another player. Often these agents will be very complex and have many algorithms and heuristics to determine their course of action in a given game state. In this paper, I will explore this field and examine different types of game playing agents and algorithms. I will also touch on some of the following questions I have about this topic myself: What can the area of game playing in artificial intelligence teach us? What happens when multiple agents in the same computer game play with or against each other and how do they fair when different goal states are present? How do the nuances of optimality in the decision-making process affect the overall game an agent plays? Additionally, I will attempt to extend my research into an implementation effort with an experimental program in which two agents compete in a custom game environment, which I will discuss more in depth later. My work will be supported by multiple references from the same field, including journals, websites, and other types of publications. My most prominent thought on game playing agents is that they need to be versatile, not just in their adaptability in differing circumstances, but also in their ability to make “wrong calls” or make suboptimal moves. I believe this is something that is important when considering how to structure an agent for this type of environment, because ultimately we should want to play against an agent that is most human-like, and there is no human that makes the perfect move every time.

“The typical artificial intelligence in gaming is single-agent.” explains Kevin Markley and D.

Michael Franklin in their article from Southern Polytechnic State University (Franklin 1). They go on to describe how the typical AI focuses solely on the enemy player and pays no mind to other potential agents that may be doing the same (Franklin 1). Their article expands on how, more often than not, in video game and computer game software, multiple enemy agents usually do not act in accordance with each other resulting in the player being able to use exploitable strategies to play a game in an unintended way. If anybody was to actually look at most computer and video games today they could easily see that the gaming market is completely saturated with games in which there are mindless "brute force"-style enemy agents. The opposite—well-designed, learning, enemy agents—only exist in games that are far and few and I feel that the games that do have these types of very strong, dynamic enemy characters are impressive cooperative, challenging, daunting, and memorable. An example from each of these could be any of the games from the original Mega Man video game franchise on the original Nintendo Entertainment System, and the Dark Souls video game franchise on many of the newer game systems. In the Mega Man series, the enemy agents only had one directive, to attack your character and not pay attention to anything else, which was very prominent in games that were developed back then and even today. In contrast, the Dark Souls games included enemies that would not only cooperate and strategize with other agents and was a very difficult game as a result and is discussed in an article by Dan Ackerman, Jeff Bakalar, and Scott Stein (Ackerman 1). In addition, enemy agents in the latter caliber of games can often appear human-like because of their unpredictability via suboptimal decisions, granting a much more engaging experience overall.

Continuing on with different nuances for game playing agents, Epstein describes how the agent called Hoyle learns to play multiple board games and is different than other game playing agents in her article, "Learning to Play Expertly: A Tutorial on Hoyle" (Epstein 1). In the beginning of the article Epstein notes that Hoyle is different than other agents because of the way it learns, stating "Hoyle has been cognitively validated, that is, it has been shown on several simple games to learn and play much the way a person does." (Epstein 1) The program was designed to learn and play multiple different computerized versions of board games that provide a challenge. Over the course of the article, Epstein describes what makes Hoyle so unique; a tiered decision-making system, weighted learning system for the decision-making, a multitude of heuristics, pattern recognition, and being able to settle on a possible suboptimal move without opting to find an optimal move are just some of the features it has.“Hoyle works, in part, because it already knows what to learn and how to learn it (Epstein 27). The fact that the program can choose to make a seemingly sub-optimal move, called satisficing, at times makes me think of how human-like that is—for a human player to just find a move that looks “pretty good”, instead of thinking five or six steps ahead. Another brilliant feature of Hoyle is that its tiered decision-making system, run by data structures called Advisors, prevent the least optimal moves from being played while allowing for the sub-optimal moves to played (Epstein 5). Hoyle learns games fast and does so without always making the most optimal play, this is something that makes me believe that feature is very important to the game playing field. In addition to all these other features, this program is able to test itself against many other predefined agents of varying skill levels. The preprogrammed agents include a random player, a perfect player, and a reasonable player, where they reasonable player makes adequate choices at a set percentage (Epstein 5). Through competing against these agents, Hoyle able to train itself against opponents of differing skill over and over again.

To further explore the concept of game-playing, I wanted to have a hands-on experience in the subject so I made an implementation effort based on what I've learned. My implementation was initially supposed to be a game, but turned out to be more of a framework for an idea similar to the “Wumpus World” computer game. “Hunt the Wumpus”, colloquially named Wumpus World, as a map-based, hide and seek style early computer game where the player character has to find and shoot the Wumpus monster or get eaten. It is a game that is often used as model in the study of artificial intelligence. In the game I started building, however, both the human character and the monster character would have been agents that perceive and act on the environment. Since there are two unique agents in the environment, the game was meant to have a turn-based movement system; each character taking a turn to make a move. Each agent would be able to reach one of two possible goals as a result of their actions. The environment of this game is the game map which is separated into tiles with each tile containing: a doorway, a hole in the ceiling, a pit, a chest of gold, a sword, or a pile of boulders. The goal states for the human character would be if the environment reflects that the agent either found the sword and killed the monster, or that it found the chest of gold and exited through the stairs out of the environment or through the hole in the ceiling. The goal states for the monster character would consist of attacking the human character or picking up the pile of boulders and moving to both the stairs tile and the ceiling hole tile to close the exits. Like Wumpus World, each adjacent tile to an object gives some indication to the agents as to what is in the neighboring tile. Both character agents would be able to move twice, that way if the opposing agent was near, they would have a chance to catch them. Lastly, there are pits that both the human and the monster can fall through if they make a risky choice, which automatically causes a goal condition for the opposing agent to be true. I had planned for the title of the game to be "Adversarial World" since the flavor of the game is that a human is exploring ruins for treasure that is guarded by a monster and algorithms will likely use some form of adversarial searches.

There was a lot that I was unable to finish for my implementation of the game, unfortunately but I did plan out much of it, compare it to articles from sources like Susan Epstein and Jacques Pitrat. Pitrat says that a general game-playing program must have an algorithm indicating the winning state and an algorithm enumerating legal moves (Pitrat 1). I made sure to take care in designing the environment that the agents will be acting on. It consisted of a two-dimensional array of nodes, with each one containing data such as an object, information about an agent, and information about neighboring tiles. When an agent moves to a new tile they will be able to perceive the tile's data, make a decision, and act on it. As for specific changes to that objects and agents make to the game map environment, tiles will have the howling wind signal to indicate a pit nearby, the brightness signal to indicate the stairway out or the ceiling hole, the crumbling rock signal to indicate the pile of boulders close by, the gold gleam signal to indicate the treasure chest nearby, the silver gleam signal to indicate the sword, the growl signal to indicate the monster is nearby to the human, and the footsteps signal to indicate that the human is close to the monster. Initially I thought I would have both agents play a series of games, where the environment changes, and different game maps get played, similarly to the rigorous testing that Epstein's Hoyle could go through. I also wanted for the agents to be influenced by past decisions and to test them on both a changing environment and a static one and see the progress they make, using a weighted learning decision-making process, in a way similar to Hoyle's learning ability without the other high-level features. Unfortunately I was not able to get this far in my implementation to see both of these plans to fruition.

As far as the capabilities of the both agents go, the human agent was merely an extension of a possible human agent in Wumpus World, and the monster agent is nearly the same. The human character will start off at a disadvantage to the monster at the beginning of each game, however through finding certain objects they can level the playing field. In order to reach either of its agent-specific goals, the human must find a sub-solution—the treasure chest or sword. Once a sub-solution is found, the human can try to find the corresponding main solution (e.g. The stairways or ceiling hole if the treasure chest is found or the monster if the sword is found). One more thing to note about the human agent is that it is only be able to carry one object at a time, and could swap one it was carrying for one on a tile it entered if it reasoned the object on the tile would be a better path to take for reaching a goal state. The monster character agent will start off with an advantage over the human, being able to eat the human if the choice is available since the human agent would start the game unarmed. The monster will be able to reach one goal without needing to find any sub-solutions—attack the human—and will be able to reach the alternative goal by finding a small series of consecutive sub-solutions. First, the monster will need to find the pile of boulders object, followed by moving over a tile on the game map which contains the ceiling hole or stairway exit, and then move to the other tile where the other, unblocked exit, is located.

There will be times when both agents are required to make a guess move, exploring a tile on the game map without enough information as to what is on that tile. In these cases, the third goal state—when one of the two agents falls through a pit—presents itself; in this circumstance, the opposing agent reaches a goal by default. As I mentioned before I wanted to implement a weighted learning system to influence moves by each agent; I was, however able to create formulas to manipulate weights for different moves, even on moves that could have been equal parts optimal and unoptimal.

Through both my research and hands-on experience a lot of information about game-playing has come to light. First of all, game-playing is an extensive and core part of artificial intelligence that I feel we can use as testing grounds to learn so much about how to develop agents. My research also leads me to believe that to make an agent better at learning, we might need to make it better at being human, and by that, better at making "just adequate" decisions if it does not result in a dangerous situation. The game-playing agent Hoyle in Epstein's article emphasizes this incredibly well. In addition, many artificially intelligent enemies in modern computer and video games have become increasingly good at engaging human players and being unpredictable and even learning methods to fight the human player (Ackerman 1).

If I had been able to, there are many additions that I would have liked to have added to my implementation. As well, as completing the development of all the initial features of the "Adversarial World" game I proposed, I plan to extend the software to include a better decision making system than what I had attempted to build originally. Franklin and Markley's article has many good points, especially regarding the simplicity of certain agents (Franklin 1). Prior to this, the agents I was developing were to have a rudimentary to determine where to move, thinking of only the current board state in time. In the future I will expand on this to create a more dynamic decision-making system, in which both agents are able to compare possible board states one or two steps into the future. In addition, I also plan to implement sub-optimal decision making features for the agents to see how they will learn when they aren't required to make the optimal move and are able to just choose an adequate move instead. I feel that we have so much to learn from game-playing field of artificial intelligence. In competitive games between two or more players we are able to see one aspect that makes us human, our inability to always find the optimal solution. Programmed agents or ones that learn typically know the optimal moves or are able to find them. To take these some of these agents to the next step, perhaps it is, indeed, in the best interest to build them to smartly and strategically decide on a lackluster or only acceptable move. In an article by Peter Murray, of the Singularity Hub website, two agents programmed into the first-person shooter game Unreal Tournament 2004, passed a pseudo-Turing test. Murray describes how a test match was played between multiple human players and bots, with people judging who was a human and who was a bot. In the end, one of the bot's that passed had been given a personality, in that it would after suffering a loss it would target the player character who caused it and seek out revenge (Murray 1). Compare this agent's features with the “personality” of the Hoyle game-playing program and it seems like the idea of giving agents their own flavor or less strict decision making skills is a step in the most human-like, intelligent direction.

References

Ackerman, Dan, and Jeff Bakalar, and Scott Stein. "Is Dark Souls too hard?" CNET. Web. 11 Oct. 2011 <http://www.cnet.com/news/is-dark-souls-too-hard/>

Epstein, Susan L. "Learning to Play Expertly: A Tutorial on Hoyle" Learning in Games (2000): 1-28. Web. <http://www.compsci.hunter.cuny.edu/~epstein/papers/Hoyletutorialfinal.pdf>.

Franklin, Michael D., and Kevin Markley. "Multi-Agent Artificial Intelligence in Pursuit Strategies: Breaking through the Stalemate" Web. <https://www.aaai.org/ocs/index.php/FLAIRS/FLAIRS14/paper/download/7886/7847>

Murray, Peter. "Turing Test Prize Has Two Winners." Singularity Hub. Web. 08 Oct. 2012 <http://singularityhub.com/2012/10/08/turing-test-prize-has-two-winners/>

Pitrat, Jacques. "A General Game Playing Program." Edinburgh University Press (1971): 125-55. Web. <http://aitopics.org/sites/default/files/classic/Pitrat/JP4.pdf>.