3-23-2004

AI in Computer Games: From the Player's Goal to AI's Role

Jeremy A. Glasser
University of Nebraska, jglasser@cse.unl.edu

Leen-Kiat Soh
University of Nebraska, lsoh2@unl.edu

University of Nebraska–Lincoln, Computer Science and Engineering
Technical Report TR-UNL-CSE-2004-0004
Issued 3/23/2004

AI in Computer Games: From the Player's Goal to AI's Role

Jeremy A. Glasser and Leen-Kiat Soh

Computer Science and Engineering
University of Nebraska-Lincoln
Lincoln, NE 66588-0115
{jglasser, lksoh}@cse.unl.edu

Abstract

This paper addresses the role of Artificial Intelligence (AI) in a variety of game genres. Every game aims to entertain (though educational games have secondary objectives). Each genre approaches entertainment in a unique way. We explore the methods used to draw the game player's attention. We then consider how the AI interacts with the player to promote both entertainment and an interactive environment. We also consider some of the techniques that will shape tomorrow's games. Included are opponent strategies, interactive environments, and multiagent systems (MAS). While different, each approach can aid in creating more immersive and challenging gaming experiences. Our goal is to create an AI opponent that is not only a capable opponent but also reacts to and improves with player. We consider what elements are necessary to create an opponent capable of creative use of its environment. The game environment, too, is extremely important in creating an immersive experience. Finally, we consider the possibility of creating games that include emergent behavior and multiagent systems.

1. Introduction

In this paper, we describe the current role of Artificial Intelligence (AI) in games. We will address some techniques that are currently being used to provide a more challenging and engrossing game experience. We also explore current problems in game AI and current approaches for combating these problems. To avoid confusion, we define several terms that are used throughout the paper. The human playing the game is called the player. A character is the representation of the player or computer AI in the game world. Finally, the opponent will refer to a character that is competing with the player. Though often controlled by AI, another human player may also fulfill the role of the opponent.

Our research project is related to creating computer games that are just difficult enough to be always entertaining to the players. For a computer game, if it is too difficult, a player may be disheartened and give up; if it is too easy, a player may become bored and stop playing. We envision a computer game that models human players and adapts to their abilities so that it is always entertaining.

Note that this paper was inspired by the work of Laird and van Lent (2000). We have created a layout that similarly covers a variety of game genres. While most of the genres listed are similar, our work focuses on finding elements in each genre that can be used to make a challenging and rewarding game experience. We show some techniques that are currently used in video games and we also look at shortfalls that detract from or entirely destroy the realism created by the game. Using these observations, we present ideas for creating game AI that will play the game more effectively thereby creating a "successful" opponent.

Note also that our focus is on computer games where the game states are open and not completely defined. Thus, we do not discuss computer games in the classic (searchable) genre such as chess, checkers, tic-tac-toe, and Othello. Our discussions address games of the following genres: First Person Shooter, Action/Adventure, Real-Time Strategy, Role Playing, God, and Sports Games. While many AI problems, as well as games themselves, tend to cross genres, we attempt to find key roles for each genre separately.

To avoid confusion, we define several terms that are used throughout the paper. We use the term *player* to refer to the human playing the game and we use *character* to refer to his/her representation in the game. Finally, we use the term *opponent* when referring to a human or AI-controlled competitor.

The reward for playing comes in a variety of ways depending upon type of game the player is engaged in. The reward may be financial in nature, the knowledge learned from the game, and/or the enjoyment from playing the game.

2. Game Genres

2.1. First Person Shooter (FPS) Games

These games, by definition, grant the player a view similar to what the character would see, thereby placing him/her directly into the action. They are often referred to "twitch" games because of the emphasis on fast movements and accurate shooting. To aid the player, power-ups (med. packs, ammo, armor, etc.) are scattered throughout the levels. The number and difficulty (better aim or tougher hide) of the enemies will increase with the level of the power-ups (new, more powerful weapons or items) thereby providing new experiences while maintaining a desirable challenge for the player as the game progresses.

2.1.1. The Player's Goal

Traditionally, these types of games tend to focus on rapid movement and aiming skills rather than intelligent computer opponents. Because of the increasing availability of Internet access, these games have begun to focus on multi-player gameplay. This shift has resulted in a change of opponents from simple "see and chase" AI-controlled players to human opponents.

2.1.2. Role of AI

AI-controlled opponents are generally used as support characters or as opponents. Recent games, such as Bungie's Halo, have made an increased effort to provide squad based support from its AI-controlled characters (Bungie 2003). There are several situations in the game where the player has a supporting cast (controlled by game AI).

As mentioned earlier, the enemies in this type of game are not necessarily intelligent. Early games made use of simple tracking (go to where the player is and attack him/her) and the only thing distinguishing the enemies at the beginning of the game from the ones at the end was the amount of damage they could take and the amount of firepower they used to dispose of the player. This means that the same tactics used at the beginning of the game would work at the end of the game (it may just take a while longer to destroy the enemies). Later games, such as id software's Quake II included bots (AI-controlled characters) that were challenging opponents (id 2003). However, they were challenging because of their superhuman reaction times and aiming abilities as opposed to their complex tactics or game knowledge (Laird 2001).

To make more engrossing FPS games, developers will have to rely on a strong online presence (providing ample human-controlled players) or develop better AI. In today's game market, a successful single player experience requires better AI than what was common even just a few years ago. It seems likely that better AI will fill at least part of that void. Further, even though it is unlikely that the popularity of online gaming will wane, there will always be a need for the "one extra" player to even teams or the tireless support character that is content to play backup to tomorrow's game players.

2.2. Action/Adventure Games

Of all the game genres discussed in this section, action games are likely the most diverse in terms of gameplay and content. For this reason, we focus on the games with a linear progression of levels. Examples include the Super Mario Brothers series, Sonic the Hedgehog, and Crash Bandicoot. The focus of these games is to advance a (usually visually unique) character through a progression of fantastic, fictional levels (either 2D or 3D), collecting odd objects such as coins, rings or apples as in our three examples above. The collected objects provide power-ups that aid the player in dispatching his/her myriad foes.

2.2.1. The Player's Goal

Action games generally use a linear storyline to drive the player's progression through the game. The Super Mario Brothers series made a common theme of "save the princess". As the player progresses through a level-based game, he/she will face tougher foes and more challenges. Often a level or stage (a group of levels) will culminate with a battle against a boss character. Generally much harder to kill (because of the greater vitality and bigger weapons) than the opponents that populate the level, the boss character must be defeated to progress to the next level. Visual cues, attack patterns or indications of weakness, are presented to aid the player in defeating the boss character. For example, an enemy ship at the end of an action shooter may have a "flashing red dot" that must be hit in order to cause damage.

2.2.2. Role of AI

Because many of these games rely on timing or jumping challenges, current AI tends to be limited. Super Mario Brothers, for example, was populated with enemy turtles that walked or hopped until they hit something. They then turned them around. In the future, we expect to see enemies that can adjust their movements (create dynamic paths) to create new challenges each time the game is played. These enemies could also adapt their movements so that a player cannot rely on a single attack or move or become overly frustrated with a specific part of a level.

2.3. Real-Time Strategy (RTS) Games

Real-time strategy games give the player a view akin to an army general looking down on a miniature model of the battlefield (God's eye view). They usually involve a variety of different units (game pieces) each of which has a set of strengths and weaknesses—e.g., a scout unit may be really fast, but have poor armor, while another unit may be slower but have much better armor and more firepower. The players race to secure resources, construct new units and structures, and to destroy the opponents' structures and units. The last player to have a structure (or unit) standing wins the game.

2.3.1. The Player's Goal

A real time strategy game will usually comprise of three main goals: (1) use specialized units to collect and gather resources, (2) use these resources to build new structures and units, and (3) use the new units to defend your structures and to destroy the opponents' structures.

2.3.2. Role of AI

The AI for RTS games has to manage the three goals outlined above. These tasks are actually quite complicated. The AI must decide how many basic gathering units should be created in order to provide a suitable influx of resources necessary for completing steps 2 and 3. Further, these units are often quite weak and will require protection if they will be traveling far from the home base. Step 2 requires an overall knowledge of the building hierarchy. Often, certain structures must be built before others. This means a build order must be defined. Finally, the AI opponent must balance the construction of new structures (providing new technology, buildings or units) while sustaining a suitable defense. To be successful, the AI opponent must be able to devise a plan for each step. Further, these goals must fit into an overall plan for destroying the other player(s).

Even an AI opponent that is capable of successfully building and defending a home base will still have weaknesses. Frequently, there are times when it becomes necessary to move the base to a new location or to create a second one. Securing a local hold on resources is perhaps the most common reason for expanding the base. This technique is rarely seen in AI opponents.

A technique currently used to aid the AI opponent in managing the difficulty implicit in playing a game with this type of structure is for the developers to grant the AI opponent some benefits. Often referred to as "cheating" by players, the AI opponent will start with a larger store of resources, more units or buildings, or both. While it may be more difficult to beat an opponent under these circumstances, the player's strategy does not have to change from one skill level to the next. Once the player is able to overcome the initial disparity in the balance of power, the same techniques that worked to finish off the "easy" opponent will work to defeat the "hard" opponent. This fact makes for a less enjoyable solo experience, leaving the player no other option than to turn to Internet play for new challenges.

Another shortfall of existing AI opponents is that their attack strategy is often very predictable. This is because an AI opponent is not able to evaluate its units. Consider a game that has an anti-air unit that has very weak ground-based armor and attack capabilities, but is unmatched when it comes to destroying air targets. Another unit may have excellent ground-based attack power and defense, but is vulnerable to air targets. The game developers could set rules so that the AI opponent always builds four anti-air units and three ground-based units and then uses them to stage an attack. While this squadron of units may be balanced, it is easy to counter. A human player will quickly realize that once this squadron has been completed, the AI opponent will attack with them. Further, the human will be able to determine the amount of time required to rebuild this squadron and will use that time to make appropriate defensive adjustments. An AI opponent that is able to evaluate the player's weaknesses and construct custom squadrons of units specifically to exploit those weaknesses is far more formidable than one that has a "hard-coded" solution to each perceived situation. To do this, an AI opponent could use its knowledge about the game environment and units to counter an attack. If the player creates air units, the AI should develop anti-air defense comparable to the perceived threat. Just as a human does, the AI could learn to recognize repeated attack strategies and devise a counter-measure. An AI opponent could also perform self analysis to determine how well defended it's base is. This way, it would know how and where to allocate its resources.

In addition to mimicking human construction behavior, the AI plays an entirely different role in RTS games. Each unit must be able to navigate the map. There are air units that can travel over virtually anything on the map, there are water vessels that are (obviously) restricted to water environments and there are land-based units. The majority of these units should be able to find their way from one point on the map to another. This requires, at minimum, some basic path finding capabilities. In addition, the unit should be aware of anything that may be causing it damage. For example, if an enemy is firing at the unit, it should be able to alter its path, return fire, or retreat depending upon the situation. A welcome improvement would be to have the AI find a best path -- one through friendly territory as opposed to unexplored (or enemy) territory, the safest (easiest to defend oneself) route, or the one that offers the greatest visibility.

Another shortcoming of existing unit AI is that when moving a group of units, the unit formation either devolves into a "disorganized mass" or a single-file line. The mass of units makes it difficult to manage individual types of units and the single file line tends to make the units more susceptible to attack. Either way, the AI design is forcing the player to unnecessarily "micro-manage" his/her units. One way to combat this would be to enforce formations while moving. Each unit would then be part of a multiagent system that would require agents (characters) to communicate so that they maintained proper distance and speed while traversing the game world.

These AI opponents should have both strategic and tactical components. The RoboCup Robot Soccer World Cup has provided research for real-time competitive environments (Bruce et al. 2002, Veloso et al. 1998).

2.4. Role Playing Games (RPG)

Role playing games tend to place a higher emphasis on story-driven gameplay and character development than any other genre. Many of these types of games have some basis in the paper-based Dungeons and Dragons games.

2.4.1. The Player's Goal

The player is usually confronted with some sort of mystery regarding unusual or mystical events that have recently transpired in the local area. As the search for the source of these occurrences continues, the player is drawn deeper into the story and will often encounter new enemies and frequently, allies. He/she will also encounter great wealth and many mystical items and armaments. These items, as we describe later, can be used to augment the character.

A primary focus in RPG games is that of character development. This is usually determined by some experience factor. That is, as the player progresses in the game, he/she will be able to augment his/her character's abilities. These are dependent upon the game, but character strength, vitality and ability to use magic spells are common. Tougher enemies offer greater rewards; they are often a source of experience points as well as new items. A magical item may increase strength or vitality making it more valuable than standard (non-magical) equipment.

2.4.2. Role of AI

A RPG game will require two primary types of AI. The first is concerned with support character AI and enemy AI. Both are concerned with character movement and strategy, though their goals differ. The other type of AI is used for dialog with non-player characters (NPCs). NPCs populate the landscape and are often used to drive the story as well as offer side quests (requests or missions for the player to finish that do not necessarily relate directly to the overall game story). Of course, there is often a reward (item, experience points, etc.) for completing these quests.

Support Characters. A support character is an NPC that is part of the player's party, but not actively being controlled by the player. Many RPGs have a support character of some sort and some offer the ability to switch between the main character and the supporting characters. Ideally, a support character will do just that, support the player's character. They should do this in a manner consistent to their character type and abilities. For example, a wizard character should not run into the middle of a group of enemies with a sword in hand.

Enemies. Scattered throughout the RPG worlds are countless magical or mythical beasts and creatures. Each one must contain at least some primitive AI. The one common theme for an enemy should be its ability (based upon its intelligence) to assess its strengths and weaknesses as well as those of the player's party. If a dragon notices that the player's party has a poor resistance to fire or heat damage, it will likely find itself at an advantage. However, if they have enlisted the support of a well-known dragon slayer and have even moderate protection from fire, the dragon's pillaging days may be over. An enemy wizard, on the other hand, should know what types of spells it can cast effectively, which ones will be effective, and how many it can cast before its mana (or magical power) wears out. An inexperienced or brash wizard may try a powerful spell over which he has little control in an attempt to end a battle quickly while a wiser wizard may opt for a more strategic approach to finishing his opponent.

Dialog. Finally, there is the issue of character dialog. As mentioned earlier, the RPG games often have a very rich storyline. To help draw the player into the game, NPCs often relay information relevant to the story. Delivering the game story in this manner deepens the experience because the characters relaying the information are actually part of the game world. In the past, static decision trees dictated the character's dialog. Decisions made by the player cause the character dialog to change. While this approach can be effective, the illusion of intelligence is spoiled when the NPC is visited more than once. Often, the same dialog branches can be revisited on subsequent visits. For example, a bartender may request five pieces of gold for his finest ale. The player is then prompted with three responses: "No thanks, I don't really need a drink," "A drink would sure be nice, but I only have three gold pieces" or "Five gold pieces is a lot, but the quality of your ale is well known. I'll take two." The player, not willing to part so easily with his hard earned gold offers the bartender three gold pieces. Even though the bartender may reply with a short retort, he/she will gladly offer the player the same three options if the player talks with the bartender again. A far more believable, but much more expensive (in terms of modeling, computation and storage space) solution is to give the bartender an initial disposition and memory. Thus, even if the player leaves, the bartender will "remember" previous conversations and experiences with the player and react accordingly.

Fable is a new game (developed jointly by Big Blue Box Studios and Lionhead Studios) that promises to add some new features to traditional RPG gameplay (Big Blue Box Studios 2003; Fable 2003; Lionhead Studios 2003). In this game, the player is granted the freedom to play as an admirable hero, as a shady adventurer, or anywhere in between. If a player chooses righteous deeds, his/her character will reflect that. On the other hand, if the player opts for more evil or self-serving deeds, his/her character will become a dark, foreboding person. The character will also keep scars from battles fought and his muscles will weaken as old age takes him/her. As is the case with other titles, the NPCs, too, will be able to recognize these changes. They may also “hear” tales about the character’s remarkable exploits or about how he/she should not be trusted. Thus, this game provides players with a lot of freedom in how they play the game.

2.5. God Games

God games often place the player high above the world he/she is manipulating. They are designed to give the player god-like control over some or all factions of the game world. Perhaps the most well known of god games is SimCity. The latest version of the game, SimCity 4, grants players the ability to “sculpt mountains, gouge valleys, and seed forests” (Maxis Games 2003). The primary goal of SimCity, as the name implies, is to build a city. The player takes on the role of the mayor of the city. Sections of land (the game map) are zoned for industrial, residential or commercial properties. The player must also choose the location of city parks, power plants, water mains and power lines. The inhabitants of the city will complain when crime or pollution is high and laud and praise the player when taxes are low or when a new amusement park is constructed. Other games offer similar levels of control to the player.

2.5.1. The Player’s Goal

Generally, the player has no direct influence over the inhabitants in these games. Instead, the goal is to create an environment that is suitable for growth and expansion. The player is then rewarded through the advancement of the autonomous inhabitants in the environment.

2.5.2. Role of AI

The AI controls the actions of the autonomous units. In the case of SimCity, the player must be notified when the crime rate is high or when a fire breaks out in a specific part of the city. The city’s inhabitants provide the player with this information. The buildings, too, are autonomous units. They determine if they have enough power and water. They decide if the surrounding area is suitable for growth and if it is, they change their appearance to reflect the fact to the player.

As new techniques for simulating different animals, insects, creatures, etc. progress, god games will make use of them. Given enough accuracy in the modeling of the individual units, these games could lead to the development powerful research tools. This type of simulation could also be used to train professionals in a variety of tasks in decision-making, for example. Also, discussions on real-world simulation lead to interesting ideas presented in the philosophy section of “The Matrix” website (Matrix 2003) regarding humans living in a simulation (Bostrom 2003).

2.6. Sports Games

This genre gives the player an opportunity to play a variety of real-life sports. Included are games such as football, soccer, tennis, basketball, baseball, and automobile racing. There is also a growing subclass of games that mimic the extreme sports. Perhaps the most notable of these is the Tony Hawk series developed by Neversoft (Neversoft 2003).

Sports games can be categorized into team and individual sports. In the games depicting team sports, the player will either manage the team or play as one of its members. The player will usually have control of the key team member. For example, in a football game, the player would control the quarterback; in a basketball game, the player would control the character dribbling the ball; and in baseball, the player would control either the pitcher or the batter. In individual sports such as golf or racing, the player is given control over the game character.

2.6.1. The Player's Goal

Generally speaking, the rules for the team sport computer games closely resemble their real-life counterparts (though most games offer settings that can drastically change gameplay). There are also some notable exceptions such as the NFL Blitz and NBA Jam series, both published by Midway Games Inc. (Midway 2003). These titles focus on a fast-paced arcade (as opposed to simulation) style of gameplay.

In individual competition, players compete in races, play tennis matches, or play rounds of golf just like their real-life counterparts. The extreme sports give the player control of athletes who perform stunts and tricks on bikes, skateboards, snowboards, etc. Often the focus of these games is not on realism, but on creating new tricks and combining others.

To make these games more strategic, there are often "unlockable" items, or rewards, offered after the player completes a game challenge. A racing game might grant the player a new car or racetrack while some tennis or volleyball games let the player choose a new character. These items extend game life by providing new challenges for the player.

2.6.2. Role of AI

According to the Entertainment Software Association (ESA), 19.5% of all console games sold in 2002 were related to sports. In contrast, less than 9.6% of computer games purchased were sports titles (ESA 2003). This fact, coupled with the notion that the online gaming is still in its infancy, indicates that there is a need for good AI opponents. This implies that this genre, perhaps more than any other, will see improvements in AI opponents to create better coaches and individual characters.

Even if online gaming becomes extremely popular, it is not likely that a football game will support 22 players at the same time. This is simply because it is not likely that many players would prefer to play center rather than quarterback. Unless some considerable effort went into designing a system for blocking mechanics many players would opt to host a new game rather than joining an existing one as an offensive lineman. The quarterback must read the coverage and make passes, the receivers must beat the coverage and are able to carry the ball and the tailbacks are used primarily for running the ball but can also act as receivers. The defense has players that try to stop each of these players. While obviously important, the linemen are rarely, if ever, directly involved in moving the ball. This situation arises in other genres as well. For example, an RPG might contain a wizard and a warrior. The wizard will be good with spells, but poor in hand-to-hand combat. The warrior, conversely, will prefer direct combat. The key is to have the game present comparable challenges to the player despite the differences in the game characters. Because not everyone will want to play every available character it is important to have AI-controlled characters that will react naturally in the game.

Current AI-controlled characters act according to simple scripts (e.g., "move to this location and then look for someone to block" or "run five yards up field and then turn 90 degrees left".) Good scripts are hard to notice and may work well most of the time, but they leave no room for the game characters to adjust to new situations. For example, linemen do not actually adapt to the blitz, they just block the linebacker because he got close enough. A center will not, in general, drop back and follow the quarterback when he has no one in front of him. Receivers break off routs *when their route ends* and "improvise" but they are not intelligent enough to find a hole in the coverage. Thus, most these AI players are scripted, reactive entities with no strategic modeling or pro-active capabilities.

What if there were a defensive line shift and the scripted lineman had nobody in front of him? Imagine if an AI receiver could find a "hole" in the zone and wait for the pass instead of continuing the out route. Even more impressive is an *adaptive* offensive line that can make adjustments during the play in order to protect the quarterback. Just as important is a fullback that can anticipate the pressure and make a block to buy some extra time for the quarterback. These scenarios beg for individual recognition of the situation as well as agent-to-agent communication or modeling between the game characters. For example, a third baseman could recognize the fact that the batter is a good bunter and may play in on the grass. A basketball player could recognize the 2-3 zone and move out to the perimeter to find an open shot. These problems will eventually call for improvements not only in individual game characters, but within team coordination and communication as well.

One other point of interest in this genre is the role of the coach. An artificial coach could evaluate a situation and propose a counter move. Many football games use a rudimentary AI process; for example, a simple Boolean check for the 4th quarter and determine the best move is to go for the first down when there is no conceivable way to achieve victory. But, lacking of flexible and contextual reasoning (such as temporal issues), these AI processes often fail to make correct or novel decisions. Consider the following illustration. To save time, a standard quarter length for a football (video) game is 5 minutes (while regulation quarter in the NFL is 15 minutes). Because it has no conception of the pace of the game and is unable to model human players at a strategic and cognitive level, it is not uncommon to see the computer AI make poor time management decisions. For example, if a team is losing a football game by 35 points, there is no need to call a timeout with 50 seconds remaining on the clock. An understanding of how the game flows and even momentum should provide more realism. Perhaps a game developer should include an arcade mode if a player wishes to win a game 80-7.

Many of the team sports games are mimicking the things that television stations are doing with the *visual presentation* of the game. (One could also argue the vice versa.) One of the keys to creating this experience is to have a commentator relay what is going on during the game. ESPN and John Madden both promote their own NFL football games. While this type of features is much more common than it used to be, there are still contextual problems. For example, when the quarterback takes a knee to run out the clock at the end of a game, the announcers should simply state that fact. It would be surprising for a television announcer to state that there was "a loss of two on the play". Instead, they would likely begin their end of game wrap up. An AI character should be able to recognize the contexts and their appropriate responses.

Further, current game commentary tends to lack breadth. The phrases and comments that were humorous or clever the first time around become repetitive and annoying. A virtual commentator should have a number of ways in which it can describe the same sort of play. An improved commentary system could prove beneficial for virtually any video game in this genre.

In the case of individual sports, such as racing, it becomes important to have other drivers that are not all the same. It can be frustrating when every other driver on the track runs the same, "perfect" line around the course. This is a dynamic, real-time environment; drivers should make adjustments to how each other are driving, they should show "personality" and "emotion". In this manner, it is possible to have a reckless or aggressive driver.

3. Successful Opponents

The previous section highlights many of the game environments and tactics the AI must master in order to become a “successful opponent”. In this section, we look to define what it means for AI to be successful as well as some approaches that lead to successful AI.

The first rule in game development is that the game must provide a fun and rewarding experience. Generally speaking, the role of the AI is to promote and extend this experience to the player. We believe the most successful opponent is one that is “human-like”. In order to be human-like, an AI opponent should follow three rules. First, it should react or respond to the player’s actions. Second, it should provide a sufficient challenge. Third, it should “play fairly”. The more faithfully the AI adheres to the rules, the more successful it will be at being human-like. Overall, the AI must provide an entertaining and challenging game experience.

3.1. The AI Must Provide Realistic Feedback

The AI is responsible for drawing the player into the game environment. To accomplish this, the AI must perceive the player’s actions. It is true that scripts, triggers and other tricks can lead the player into believing the AI is doing more than it is. However, a player who is given a second look at the game readily notices these tricks. There will be times when a trigger does not fire resulting in part of the storyline being told out of context, granting the player an unintended advantage (knowing the location of an ambush point, for example), or even trapping the player in the game with no way to advance.

By adjusting to and accommodating for a player’s actions, the AI can create an interactive environment. Imagine if the player’s goal is to infiltrate the enemy’s stronghold. The player discovers a way to get into the building that was not accounted for by the developers. If the enemy leader reacts with a scripted set of actions, it will not be able to handle the unexpected entrance from the player. Several things may happen. The game could crash. More likely, however, the challenge of the mission will have been lessened (or removed completely). Perhaps most importantly, the game illusion will be lost.

This example illustrates the need for an AI character that not only reacts to a given situation, but is also able to react to any situation the human player may present to it. Further, it must react in a manner consistent with the human player’s expectation of what is considered realistic. For example, if a type of enemy is cowardly when encountered in the first level, it should also be cowardly when encountered in the seventh level. An NBA center should be more likely to dunk the ball than to go for the lay up. Creating characters with distinguishable traits can be very effective at keeping a player’s attention. However, to be effective, the characters’ reactions must remain consistent.

While realism is important to creating a believable game, it is just as important to know when to relax this requirement. A common example of this is when an AI character “broadcasts” its intentions to the player. For example, if a player has just been imprisoned, the AI may openly “display” the key to the lock on the wall or its belt. Similarly, if an AI character is washing his/her hands, she may remain focused on this task to the point that it allows a player to sneak by without being noticed. Other examples include “leader” characters that have a different appearance, a conversation that can be “overheard” by the player, etc. Often, we make a small exception to the realism rule in order to give the player clues or hints to help drive the story. We explore this idea more when we discuss adaptive games.

3.2. The AI Must Be Challenging

Consistent with the first rule, a game must be challenging or it will not retain its entertainment value. Currently, difficulty levels (e.g. easy/medium/hard or rookie/all-star). Each skill setting changes the actions of the AI. Too often, the difference between the skill levels requires a user to adjust to each difficulty level. That is, there is no real fluid or smooth transition from one skill level to the next—the player must always advance in discrete steps. Ideally the player's skills would progress as they do in real life. That is, the player will enhance his/her skills without necessarily being aware of the change of difficulty. In this way, the player will never become frustrated to the point of quitting. This phenomenon occurs every day. For example, a child learning to play basketball improves gradually while playing with others of equal (or slightly better) skill. Perhaps the only video game arena that provides this type of situation is an online environment where game players are competing against other humans.

A game that is capable of dynamically changing skill levels avoids another problem. A player can no longer "out grow" the game: once a player finishes a title on the hardest difficulty setting, there is not much else to do with the game.

It may also be good to note (perhaps in a future version) how gameplay can be altered to create new challenges for the player. That is, certain missions may require the player to be more stealthy while others force the player to make use if his/her arsenal (perhaps an opponent is hidden in a bunker and is nearly impossible to destroy with ballistics but a well placed grenade could be used to remove the enemy threat).

3.3. The AI Must Play Fairly

The final rule dictates that the AI play fairly. A truly good, competitive AI opponent should not need resources or information that is not available to the player (barring special abilities specific to the game character). A study conducted by Laird and Duchi (2000) found that AI characters that fired with average accuracy and had a human-like decision time were considered more like human players than characters with shorter decision times or greater accuracy. Game players quickly realize when a computer opponent has an unfair advantage. This can be done in a number of ways: the AI may initially have access to more resources than the player, it may be able to "see" through walls or fog, it may react at superhuman speeds, it may take much more damage or shoot much more accurately. These occurrences can destroy the illusion just as quickly as an AI character that does not respond appropriately to a player's actions.

4. Successful AI Environments

We expect to see tremendous improvements in game AI over the next few years. There are several reasons for this. First, like any other industry, the video game market is extremely competitive. A few years ago, the numbers of polygons per character model seemed to be an industry benchmark (at least for FPS games) (Laird and van Lent 2000). Today, creating high quality graphics is relatively easy and many believe the biggest innovations to gaming will be in interactive environments and more realistic opponents (Unreal Wiki 2003). In the next section, we explore "smart environments." These game environments could use AI techniques such as "smart objects" (Peters et al. 2003) and simulated physics (e.g. a simulating cracks in a window instead of using a static image). Without concerning ourselves too much with complexity issues, we explore the effect these environments will have on the appearance and presentation of future games.

4.1. Environment Models

Faster graphics processing has allowed game environments to become more realistic. Artists and developers have the freedom to create large, realistic, and interactive game environments.

For example, Valve's Half-Life employs such interactive environments (Valve Software 2003). This game included vending machines that could be broken, breakable glass, movable crates, and destructible boxes. The highly anticipated Half-Life 2 promises to deliver even more (Gamespy 2003).

Being able to interact with the world is taking on added significance and has led to some very interesting and reactive game environments. However, most current computer games tend to use the same effects over and over again. For example, objects such as rocks and trees will be repeated (be exactly identical) throughout the game. Damage effects will consist of a palette of three or four different damage marks. This approach, while resource friendly, greatly decreases the realism of the world.

One type of approach we have alluded to earlier would be to use multiagent systems. For example, a game building could be constructed out of individually coded bricks. Each brick could have information about itself (e.g. I am composed of baked and cured clay, I am red, I have a stress fracture, I have mortar on four sides, etc.), though it does not necessarily need to know where it lies within the wall. Now consider a simulated wrecking ball making contact with the building. The bricks that take the greatest force of the ball are likely to break. They can then pass this information on to their immediate neighbors while moving backwards in response to the impact. The result of the impact will likely be a hole slightly larger than the wrecking ball. The bricks that used to be in place have now responded to their internal rules about their construction (whether to break apart or not) and have also fallen to the ground due to the gravity rules of the game world. The bricks surrounding the hole now know that some of their neighbors are missing causing added weight from the rest of the building to be transferred (from the bricks at the top of the building) to those at the bottom. Though modeling the impact forces on the wrecking ball are not likely needed, other situations (e.g. a car hitting the building) would require that both objects receive information about the impact.

A different type of approach was used in the making of Maxis's very popular game, The Sims. This game makes use of a technique that Maxis co-founder and Sims designer, Will Wright, calls "smart terrain." According to (Woodcock 2003), all characters have various motivations and needs, and the terrain offers various ways to satisfy those needs. Each piece of terrain broadcasts to nearby characters what it has to offer. The offer or information is propagated automatically to support object-level decision-making.

This type of concept can be used in all sorts of games. Imagine a first person shooter (FPS) game that uses this technology. If a lot of players have been dying in a given room or hallway, the game could then alert a non-player character (NPC) to this fact, who then mentions that he/she has a "bad feeling" about the area (Woodcock 2002). Additionally, sports games could have a field that changes during play, showing "wear-and-tear," as the game progresses in the rain. Correspondingly, the players' uniforms could also show increasing amounts of mud.

Moreover, individuals within crowds could be modeled. The crowd is often overlooked. Frequently, they are two-dimensional sprites that are simply "copied" over and over again to fill the stadium. A much more engrossing approach would be to have a crowd that reacts to a team's performance. This may include distracted or sleeping fans in a slow paced game and very excited groups that celebrate a team score much as real-life fans do (high fives, cheers, tear down goalposts, etc.).

It is also often important to fill the world with NPCs that are not key components to the story. Consider Rockstar Game's Grand Theft Auto 3 (Rockstar Games 2003). One of the greatest features of the game is the open-ended nature of the city. If a player wants to spend his time stealing cars and racing through the city, he/she can. An important part of the illusion of a real city street is the large number of pedestrians, each reacting differently to certain events. Most, if confronted with violence, will scream and run away from the player. If, however, the player is on "gang turf" and attacks a member of a local gang, the gang member is likely to fight back. Any other member of the gang close enough to see the battle will be certain to join the fray. It is this type of agent behavior that we expect to see more frequently in upcoming games.

4.2. Path Finding

Path finding is critical to many games that include computer-controlled opponents. The A* algorithm, or some variant, remains the "industry standard" for path finding in game development. However, many developers have used influence maps, attractor-repulsor systems and flocking to some extent (Woodcock 2002).

Terrain analysis is one technique that could be used to improve opponent AI by providing information about the playing field. The AI would then be able to analyze the area to determine ambush points, hills, ridges, etc. in order to attain a tactical advantage (Woodcock 2002). More information should grant the AI an improved "sense" of the game world, thus (ideally) increasing its effectiveness and making it easier to achieve game goals.

In RTS games, influence maps could warn the game AI of enemy sight. For example, guard towers would have a much larger range of sight than a common foot soldier, and thus could better help determine the optimal paths.

Another type of terrain mapping that is showing promise is the idea of a "visibility graph" (Woodcock 2002). Visibility graphs are used to break down large areas into smaller, inter-connected ones. This method allows the game AI to use the inter-connected areas for high level path finding while using a traditional A* search for the lower level path finding within each area.

4.3. Character Models

As graphics improve it becomes necessary to create more believable characters. Creating a visually believable player or NPC avatar requires what is known as a character model. A model can be used to represent anything from an airplane wing to a human being. We envision models that are based on hierarchical organization and representation of knowledge for its component parts.

For example, this type of structure would make it possible to have individual components (in the case of a human model, the head, body and limbs) register different effects, such as damage to individual body parts. One type of game that makes use of "limb damage" is Studio Gigante's fighting game, "Tao Feng: Fist of the Lotus" (Studio Gigante 2003). In this game, blocking an opponent's attacks is discouraged by causing individual limbs to become damaged. A damaged arm will result in less effective punches just as a damaged leg will result in less effective kicks (IGN 2003).

The above concepts could easily be taken a step further. Sports games currently simulate injuries, but they seem random and sometimes out of place. A recent baseball game demonstrates the apparent randomness. On a single to right field, the third baseman tore his biceps. He was never part of the play and was sidelined for eight weeks. A far more believable approach would be to have each limb know what sorts of stresses it can take. If these parameters are exceeded, the limb could describe what sort of injury it received. For example, if an offensive lineman is engaged with an opposing lineman and a third player rolls onto his leg, it should be possible to have the lower and upper leg respond to this action. The two parts could decide if the impact of the third body in addition to the weight of the character himself was enough to strain a ligament, tear a muscle, or break a bone. If modeling were carried to this extent, individual body parts could be defined to experience fatigue and, given past injuries, be more susceptible to the same type injury.

NaturalMotion is a UK-based company that is currently integrating AI and adaptive techniques into three-dimensional character models (NaturalMotion 2003). Their primary focus is not on "dead" rag doll character models, but reactive, agent-like characters. They use what they call "Active Character Technology" (ACT). ACT allows developers to create interactive characters. These characters are able to learn to walk and will react to stimuli in their environment.

4.4. Adaptive Games

We believe adaptive AI will distinguish game titles in the near future. We have mentioned several roles in games that could be filled by adaptive agents. Now we consider a slightly larger adaptive role. As mentioned earlier, there are frequently problems when a player has to choose a skill setting. Sometimes the easy setting is too easy while the medium setting is too difficult.

Games of the future will no longer require a player to choose a difficulty setting. Action games will have opponents that make better decisions and choose a more tactical approach to defeating the player. Puzzle games will give fewer hints as the player becomes adjusted to the style of the game. NPCs will broadcast fewer messages regarding the approach the player should take to solve a problem or advance to the next section of the game.

Adaptive games will continually adjust to the player's level. They will improve with the player and will always play fairly. In this way, they provide a continuous challenge to the player. This is the very reason that online gaming is popular. The primary reason online games are popular is because they provide a persistent challenge to the player: there is always someone out there who is better than you.

4.5. Realistic Agent Control

"As computer games become more complex and consumers demand more sophisticated computer-controlled agents, developers are required to place a greater emphasis on the artificial intelligence aspects of their games" (van Lent and Laird 1999). There is considerable research being done to create realistic AI characters. Examples include agents that handle social interaction (Brooks et al. 1999), social dynamics (Scassellati 2001), natural language understanding (Varchavskaia, Fitzpatrick and Breazeal 2001), conversation interfaces (Cassell et al. 1999; Zue 1997), creative play (Bryson 1999) as well as artistic characters and actors (Hayes-Roth 1995). We briefly describe some other interesting projects here.

The Soar/Games project attempts to capture the effect of online gaming by providing more intelligent and realistic (human-like) opponents. This project aims to create an AI engine that provides a common experience base that can be used in a variety of different games.

The next approach is a Behavior-Oriented Design (BOD) development methodology for creating complex agents (Bryson 2002). These agents have an experience base of proactive plans. They are designed to use reactive plans to choose between these modules. In this way, they should be able to react quickly to situations, but can still perceive the environment and make more deliberate, adaptable actions.

Finally, we consider an approach being used to help agents learn in sports games. The goal of this research is not to create an opponent that never loses, but to create one that never loses in the same way (van Rijswijck 2003). The approach uses behavior-based force fields that are used to direct the AI character in a specific direction. The player's direction is determined through a combination of forces. The magnitude of the force will increase proportionally as the strength of the desire increases. Using desire maps, an ambient sheep will be able to graze happily, looking for the heartiest blades of grass. If a sheep dog were to approach, however, the forces directing the sheep to the grass would be overridden by the desire to avoid the dog. Once the sheep had obtained a comfortable distance from the dog, it would be able to resume its grazing. In this way, the sheep is able to manage (with low computational requirements) multiple desires at once.

5. Multiagent Systems and Games

We have explored a number of AI approaches for enhanced gameplay in the previous section. There is another enhancement that could be made regarding multiagent systems. We see that MAS as a powerful and useful paradigm applicable to a variety of games and features such as team sports, squad-based missions, simulations, flocking or herding behavior, and negotiations. For example, groups of agents could be used to simulate a number of real-life occurrences ranging from a rioting crowd to an exploding bomb.

As game environments become more vast and detailed, the desire to create background or ambient creatures is taking new importance. These characters will not be necessary for telling the story or playing the game, but they do go a long way in extending the game illusion. Crowds moving down city streets will require not only individual needs and desires, but communication as well. A stimulus, such as a car wreck should demand the attention of all but a few nearby citizens while two NPCs walking at each other should be able to communicate between themselves to negotiate a way past each other.

Games in other genres could also take advantage of MAS. Individual racecar drivers could find new paths around the track. Basketball players could set up fast break or negotiate zone coverage. Personnel in a military simulation could coordinate an attack.

An important property of MAS that is useful for promoting realism within games is its emergent behavior. Emergent behavior is defined here as the resultant observable actions of a group of agents making local and autonomous decisions. For example, programming a support character to flee upon hearing gunshots would cause a crowd of such characters to hastily retreat from a location where gunshots are heard. Pack animals could coordinate attacks independent of their environment. Support characters could learn the player's style and play with him/her instead of forcing the player to play with (or more frequently, deal with) them. In the next paragraph we describe how emergent behavior could be used to create a RPG that is different each time the game is played.

Virtually every RPG has a wide variety of NPCs that it uses to drive the story. These NPCs often live in towns or other encampments that are discovered by the player as he/she advances through the game. Most towns will have similar features, though the characters (at least by name) and layout will differ. For example, a common town will have an inn that the player can use to refresh his party, an armory where armor and weapons can purchased, a magic shop where potions and elixirs are dealt, a tavern where the player can find additional narrative, etc.

Given the city described previously, consider the following characters. John is the city mayor. He does not like outsiders and discourages the city's inhabitants from aiding strangers. Paul owns the inn. The closest he has ever come to adventure is listening to the tales told by the many travelers who visit his inn. Pete is the local smith. He longs for adventure. The city never held much appeal for him. Mary runs the magic shop. She is very talented and friendly, but mostly keeps to herself. Cindy owns the tavern. She is outgoing and is always anxious to hear the latest gossip.

We propose that at the beginning of the game, the NPCs only have personalities and the cities only have roles. Then, each NPC will "pick" its best fit for its role. For instance, John could run the magic shop, Paul could run the tavern, Pete could own the inn, Cindy could choose the role of mayor, and Mary could become a smith. While each of the characters maintains its "personality" this city presents itself very differently than the one mentioned above. First, the mayor is much more open to strangers, but may be taken less seriously. In the first scenario, Mary could provide high quality elixirs while in the second, she would likely provide above average armor and weaponry. Because the NPCs have different roles within the city, their interaction with the player changes. This design significantly alters and enhances the game experience without adversely affecting gameplay or changing the story.

6. Conclusions

In this paper we have described AI shortcomings in a variety of different computer games, in terms of the player's goals and AI's roles. We explained the motivations for improving these shortcomings and provided a number of approaches that could be used to make game experience more enjoyable. We also explored new approaches for AI in game development including an adaptive and dynamic RPG game. Even though there are many challenges to confront, we believe that the greatest distinguishing factors for upcoming game titles will be AI related.

7. Acknowledgements

We would like to thank David Hamilton for his valuable critiques and insights on a previous version of this paper. We have benefited greatly from his suggestions.

8. References

Big Blue Box Studios. *http://www.bigbluebox.com*. (current 5 Nov. 2003).

Bostrom, N. 2003. Are We Living in a Computer Simulation? *Philosophical Quarterly* 53(211):243-255.

Brooks, R.; Breazeal, C.; Marjanović, M.; Scassellati, B.; and Williamson M. 1998. The Cog Project: Building a Humanoid Robot, *Computation for Metaphors, Analogy and Agents*, 1562.: Springer Lecture Notes in Artificial Intelligence.

Bruce, J.; Bolwing, M.; Browning, B. and Veloso, M. 2002. Multi-Robot Team Response to a Multi-Robot Opponent Team. *ICRA Workshop on Multi-Robot Systems.*

Bryson, J. 2002. The Behavior-Oriented Design of Modular Agent Intelligence. *Proceedings of Agent Technology and software Engineering (AgeS 02). 61-76*. Erfurt, Germany.

Bryson, J. 1999. Creativity by Design: A Character Based Approach to Creating Creative Play, *AISB Symposium on AI and Creativity in Entertainment.*

Bungie Studios. *http://www.bungie.com*. (current 5 Nov. 2003).

Cassell, J.; Bickmore, T., Billinghurst, M.; Campbell, L.; Chang, K.; Vilhjalmsson, H.; and Yan H. 1998. An architecture for Embodied Conversational Characters, *Proceedings of the First Workshop on Embodied Conversational Characters.*

Entertainment Software Association (ESA). *http://www.theesa.com*. (current 5 Nov. 2003).

Fable. *http://www.fablegame.com*. (current 5 Nov. 2003).

Id Software. *http://www.idsoftware.com*. (current 5 Nov. 2003).

Gamespy. Half-Life 2: The Very First Look (PC). *http://www.gamespy.com/previews/may03/halflife2pc.* (current 5 Nov. 2003).

Hayes-Roth, B. 1995. Agents on Stage: Advancing the State of the Art in AI, *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence.* 967-971. Montreal, Quebec, Canada.

Ign.com. Tao Feng Review. *http://xbox.ign.com/articles/389/389505p1.html.* (current 5 Nov. 2003).

Laird, J. and Duchi, J. 2000. Creating Human-like Synthetic Characters with Multiple Skill Levels: A Case Study using the Soar Quakebot. *AAAI 2000 Fall Symposium Series: Simulating Human Agents*.

Laird, J. and van Lent, M. 2000. “Human-level AI’s Killer Application: Interactive Computer Games,” *Proc. Nat’l Conf. A.I., AAAI Press*, Menlo Park, Calif., 1171-1178.

Laird, J. 2001 Using a Computer Game to Develop Advanced AI. *Computer*. 34(7):70-75.

Lionhead Studios. *http://www.lionhead.com*. (current 5 Nov. 2003).

The Matrix. *http://whatisthematrix.warnerbros.com*. (current 5 Nov. 2003).

Maxis Games. SimCity. *http://simcity.ea.com*. (current 5 Nov. 2003).

Midway Games Inc. *http://www.midway.com*. (current 5 Nov. 2003).

NaturalMotion. *http://www.naturalmotion.com*. (current 5 Nov. 2003).

Neversoft. *http://www.neversoft.com*. (current 5 Nov. 2003).

Peters, C.; Dobbyn, S.; Mac Namee, B.; and O'Sullivan, C. 2003. Smart Objects for Attentive Agents. In Short Paper Proceedings of the Winter Conference on Computer Graphics.

Rockstar Games. *http://www.rockstargames.com*. (current 5 Nov. 2003).

Scassellati, B. 2001. Foundations for a Theory of Mind for a Humanoid Robot. PhD Thesis. Dept. of Electrical Engineering and Computer Science, MIT.

Studio Gigante. *http://www.studiogigante.com*. (current 5 Nov. 2003).

Valve Software. *http://www.valvesoftware.com*. (current 5 Nov. 2003).

van Lent, M. and Laird, J. 1999. Developing an Artificial Intelligence Engine. *Proceedings of the Game Developers Conference*. 577-588. San Jose, CA.

van Rijswijck, J. 2003. Learning Goals in Sports Games. *Proceedings of the Game Developers Conference*. San Jose, CA.

The Unreal Wiki. Introduction to Game AI. *http://wiki.beyondunreal.com/wiki/Introduction_To_Game_AI*. (current 5 Nov. 2003).

Varchavskaia, P.; Fitzpatrick, P.; and Breazeal C. 2001. Characterizing and Processing Robot-Directed Speech. *IEEE-RAS International Conference on Humanoid Robots 2001,* Tokyo, Japan, Nov. 22-24.

Veloso, M.; Stone, P; Han, K.; and Achim, S. 1998. The CMUnited-97 Small Robot Team. *RoboCup-97 Robot Soccer World Cup I*. Berlin, Germany.: Springer Verlag.

Woodcock. S. 2002. Game AI: The State of the Industry *Game Developer Magazine.* 2001-2002:26-31.

Zue V. 1997. Conversational Interfaces: Advances and Challenges, *Proc. Eurospeech '97.* kn9-kn18. Rhodes, Greece.