



Intelligence for New Worlds

A Whitepaper

Aiseek Ltd.
7 HaBonim St.
Ramat Gan, 52462, Israel
www.aiseek.com

Introduction

Games today are better than at any time in the past. Thanks to the latest powerful hardware platforms, artists are creating near photo-realistic environments, game designers are building finely detailed worlds, and programmers are coding effects more spectacular than ever before.

Unfortunately, this leap in graphics quality and richness of detail has not been matched by a similar increase in the sophistication and believability of artificial intelligence (AI). In the words of one leading developer, “characters still act like cardboard cut-outs”. Naturally, leading studios are working hard – through sophisticated programming – to enhance the behavior of non-player characters (NPCs) and other computer-controlled agents. However, as we shall see, these efforts are severely limited by hardware performance constraints. Thus the vision of a “living, breathing” world is yet to be fulfilled – a world where thousands of NPCs move and act intelligently at all times, within a large, constantly changing environment.

We argue that acceleration of low-level AI routines can address this fundamental challenge. In doing so, AI acceleration will usher in the creation of entirely new game worlds, offering a vastly enhanced playing experience.

Game AI Today

The interest in game AI is greater than ever. For example, in the pre-release excitement surrounding a much-anticipated sequel, one leading RPG developer singled out a newly built AI engine as one of the game’s hot features. The proliferation, in recent years, of third-party AI middleware solutions is another indicator of this trend.

The AI Hierarchy. These in-house and third-party development efforts address a variety of AI capabilities. These capabilities form a clear hierarchy, with more sophisticated functions depending on lower level building blocks. Thus, we will distinguish between “high-level” and “low-level” AI. In general, high-level AI is concerned with how NPCs ought to behave in various circumstances. This includes the ability of an NPC to choose appropriate behaviors and goals in a constantly changing world. In contrast, low-level AI focuses on performing the chosen actions and providing input for higher-level behavioral decisions.

For example, consider an NPC seeing a city at a distance, deciding to go to the city to sell an item, and then traveling to the city through the woods. The various mechanisms mostly closely surrounding the taking of the decision (i.e., “I want to sell this item”) we call high-level AI. On the other hand, the external acts of seeing the city (perception) and moving appropriately (pathfinding) we call low-level AI. This example also illustrates the hierarchical nature of AI; our NPC’s sophisticated decisions depend on lower-level functions (pathfinding and perception). A similar analysis would apply to the case of an

NPC seeing an enemy, deciding to attack that enemy, and finally moving in the right direction to attack.

Higher-level AI tends to be game specific and is created using many different methods and tools. On the other hand, lower-level tasks – including movement, sensory simulation (perception), and terrain analysis – are common across many different games and game genres. Moreover, it is often these lower-level actions that are most obvious to gamers and that most affect the believability of the game experience. If an NPC moves in an awkward or aimless route from one point to another, stumbles into a static or dynamic object or fails to notice an obvious change in its environment, the illusion of a life-like world is lost.

The NPC Trade-off. Unfortunately, many of these lower-level tasks require CPU-intensive algorithms – the pathfinding algorithms used for movement are a classic example. Hence, game developers are forced into what we call “the NPC trade-off”. This compromise involves three dimensions: the number of NPCs, the intelligence of the NPCs, and the complexity of the environment. Typically, a significant increase in one of these dimensions will require constraints in one or both of the other dimensions. For example, in order to populate a city with thousands of NPCs, all moving and behaving in an intelligent fashion, we will be forced to constrain the complexity of our world. Constraining world complexity means limiting ourselves to a rather small world, or to a fairly static world, or both. Of course, with today’s sophisticated physics engines introducing continuous, unpredictable changes into the game environment, the fiction of a stable world is a thing of the past. This makes the NPC trade-off all the more acute.

Sophisticated developers may succeed in maintaining believability within the bounds of the NPC trade-off, yet by doing so they are ultimately limiting game play and the game experience itself. In order to create a “living, breathing” world, where thousands of NPCs move and act intelligently at all times, within a large, constantly changing environment, the NPC trade-off must be overcome.

We now review in greater detail the major algorithms and techniques in use today for low-level AI. The areas considered include movement and sensory simulation. We also discuss the problem of AI level-of-detail.

Movement. The current methodology for simulating movement is to use a pathfinding algorithm (e.g., A*) to plan a path, another algorithm to smooth the path found, and finally a steering algorithm to move a unit along the path. A* uses a heuristic to reduce processing requirements, which essentially amounts to trying more promising routes first. As a heuristic algorithm, A* must be tailored to specific levels and maps, yet even then it may fail to find the right path within the allocated time. Moreover, for large, complex maps, simple A* is not good enough and a hierarchical A* is used.

To accelerate computation at game time, much of the path data required by A* is calculated before the game starts (e.g., waypoint graph, navigation mesh, tiered approach

etc.). This optimization in the interest of performance seriously limits game play, since the game map is constrained to change minimally and infrequently (if at all).

The steering algorithms used also present specific problems. One of the objectives of these algorithms is to avoid collisions with obstacles and other agents along the pre-planned path. Highly dynamic games with a high object count use rudimentary collision detection techniques, causing some objects to cross each other or to maintain unrealistically large distances between them. With the imminent integration of high-end physics engines, the shortcomings of these algorithms when dealing with a large number of moving obstacles will require even greater design compromises.

Lastly, group movement remains a challenge. The movement of groups is often faulty because the basic pathfinder does not take the paths of other units into account. Thus, group movement solutions are generally programmed on a per case basis, resulting in very limited group and formation behaviors.

Sensory Simulation. All games require a means for deciding what NPCs see to allow them to interact with their surroundings. The methods used depend on the specific genre and the concessions made possible by gameplay considerations. A common method is to model each NPC's vision by casting a few rays in the virtual world and then checking against other objects' bounding boxes. Since only a few rays are cast many problems occur, such as failing to understand partial cover. This means that it is not uncommon for an NPC to see things that the player knows it should not, or to fail to see things that it should.

Furthermore, these calculations are usually not fast enough for the needs of more advanced algorithms, such as tactical unit placement and tactical pathfinding. These advanced algorithms require many line-of-sight calculations for all possible NPC destinations and for all possible covers. The alternative that many developers choose is a pre-computed solution within a static environment. In extreme cases, genres that feature a high NPC count, such as RTS games, have vision systems so rudimentary as to be almost non-existent.

More sophisticated capabilities and human-like reactions must be programmed on a per case basis. These include "camouflage" (the ability to hide on a similar-colored background) and "attention getting" (bright or fast moving objects should catch the eye and be detected faster).

AI Level-of-Detail. All the problems discussed above are aggravated by what is known as "AI level-of-detail" (AILOD). Put briefly, this means that NPCs not immediately on-screen are allocated less CPU time. This compromise can easily damage the game's believability. Consider a player entering a room, interacting with its (non-player) inhabitants and exiting the room. Upon re-entering a short time later, the player may find that nothing has changed, that the NPCs are exactly as he or she left them (or performing random activities), in marked contrast to what can be expected in the real world. This is a classic example of an AILOD-induced problem.

The Intia™ Processor: The AI Acceleration Solution

To address today's AI challenges, and to enable the creation of entirely new game worlds, AIseek developed the Intia processor. The Intia processor is the world's first AI acceleration chip. Computationally speaking, AI problems are typically search-intensive, translating into large graphs with a high degree of branching. The Intia processor tackles these specific processing challenges with a technological breakthrough – the Graph Processing Core (GPC). The GPC lies at the heart of the Intia processor, delivering fast and optimal solutions to graph-based computations.

The Intia processor is accompanied by a software development kit (SDK) for game developers. Using the SDK, studios can empower their AI modules and games with accelerated AI. The Intia processor's accelerated AI functionality includes movement, sensory simulation and terrain analysis. We will briefly review each one of these areas.

Movement. The Intia processor features pathfinding functionality that is both optimal and extremely fast. Unlike today's software-based approaches (e.g., A*), the Intia's pathfinding uses no heuristics, thereby guaranteeing that the optimal path will always be found. This optimality also means that the Intia processor avoids the common pitfalls of A*, including failures to find a path when one exists, and the generation of "artifacts" (e.g., weird, unrealistic paths). If a path exists, the Intia processor will always find it.

The Intia's pathfinding is not only optimal, it is also extremely fast. Processing time for each 100 nodes of path depth is only 10 μ s, making the Intia processor about 100-200 times faster than A*. Most importantly, this very significant speed increase gives the Intia processor the adaptability to support large, dynamically changing maps. Unlike traditional approaches, which use relatively static, pre-processed maps, the Intia processor supports maps that change continuously, with no pre-processing required. By removing this limitation, the Intia processor enables the creation of new game worlds that are based on large, rapidly changing environments.

The optimality of the Intia processor's pathfinding includes excellent support for tactical considerations. Thus, in finding the required path, the pathfinding algorithm can take full account of any tactical information, such as the need to find a path that passes through certain locations (e.g., hiding points, enemy positions).

To facilitate the integration of the Intia processor with existing studio AI modules, all common game graph formats are supported. These formats including grids, navigation meshes and waypoint graphs.

Sensory Simulation. The sensory simulation capability of the Intia processor features fast, highly accurate line-of-sight functionality. Unlike current methods, the Intia processor does not rely on simple point or box approximations for the bodies viewed. Instead, the actual visible area is computed. Again, as in the case of pathfinding, this

increased accuracy eliminates “artifacts” (e.g. seeing through walls, failures to see an object that should be seen, etc.).

In terms of speed, the line-of-sight checks are several orders of magnitude faster than current methods: 512 agents can be checked against 512 agents in 0.02s. Once again, by leveraging this immense speed, very dynamic maps are naturally supported.

Alongside the speed and accuracy of its line-of-sight checks, the Intia processor makes available a sophisticated vision model. Under this model, the probability of detection depends on the size and characteristics of the area actually visible. Thus, as in the real world, small objects or objects that are farther away have a smaller probability of being detected. On the other hand, bright or fast-moving objects have a higher probability of being seen. Excellent support for hiding and camouflage is also included.

Terrain Analysis. One area of AI that is particularly underexploited is terrain analysis. In today’s games terrain analysis is usually performed statically during the game design process; strategically important locations are identified once for the benefit of NPCs and remain fixed throughout the game. However, the advent of dynamically changing maps calls for a leap in terrain analysis speed and sophistication. The Intia processor answers this challenge with advanced terrain analysis capabilities that can be run in real-time to enable truly adaptive decision making.

To support such strategic planning at both character and game-wide levels, the following built-in topology analysis functions are provided:

- T-connectivity: the area consisting of all points that can be reached at a given cost T.
- Critical Points: strategic locations, sometimes called “choke points”, that form the bottleneck for getting in or out of an area.
- Controlling Regions: areas that permit strategic domination of the terrain.

Moreover, via the SDK provided, the terrain analysis abilities can be easily extended to encompass custom functionality (e.g., wall building, ambush points etc.). In particular, tactical information may be overlaid procedurally on the game map.

Summary

In the ongoing quest to build better games, gameplay, graphics and physics are continuously being enhanced. However, one important area that has remained largely untapped is AI – until now. With the Intia processor, developers can at last populate their games with intelligent life on a scale impossible previously. Moreover, game worlds can finally come alive with physics and other dynamic effects without compromising the intelligent behavior of the virtual inhabitants of these environments. Indeed, by freeing studios from crippling design constraints, the Intia processor enables the creation of entirely new game worlds. Be it urban warfare with thousands of soldiers fighting between exploding buildings, epic battlegrounds, or a life-like city where every man,

woman and child appears truly alive, accelerated AI promises a vastly improved gaming experience.

About Alseek Ltd.

Alseek provides innovative technologies to power tomorrow's computer games. The company's flagship product, the Intia processor, is the first dedicated processor for artificial intelligence (AI). By accelerating and optimizing behavioral computations, the Intia processor allows developers to populate games with intelligent life and to build entirely new game worlds.