DATE OF PUBLICATION 26 NOV 2013 ISSN: 2348-4098

Jaya Sachan

Volume: 1 Issue: 1

# A Combination of Video Games and Artificial Intelligence

**Jaya Sachan**

M.Tech Scholar, Jayoti Vidyapeeth Women's University, Jaipur, INDIA, E-mail- jayaschn@gmail.com

## Abstract

**AI** research and video games are a mutually beneficial combination. On the one hand, **AI** technology can provide solutions to an increasing demand to add realistic, intelligent behavior to the virtual creatures that populate a game world. On the other hand, as game environments become more complex and realistic, they offer a range of excellent test beds for fundamental AI research.

This paper will give an introduction to the area of applying AI techniques, such as learning, search and planning, to video games. The tutorial will focus on past and recent applications, open problems and promising avenues for future research, and on resources available to people who would like to work in this space. We will present concrete AI techniques used in games and give references to relevant work. We hope that the topic is relevant to both game developers looking for ways to improve their products, and researchers looking for realistic benchmarks to test new algorithms and ideas.

**Keyword:** *Artificial intelligence research, video games, learning, planning and search*

## 1. Introduction

In which area of human life is artificial intelligence (AI) currently applied the most? The answer, by a large margin, is Computer Games. This is essentially the only big area in which people deal with behavior generated by AI on a regular basis. And the market for video games is growing, with sales in 2007 of $17.94 billion marking a 43% increase over 2006. However, growth is not only in sales but also in the diversity of content offered, ranging from educational games to first-person shooters. In addition, a fascinating convergence of media is taking place with video games often having movie quality cut-scenes and narrative.

## 2. So, where does artificial intelligence come into play?

We argue that both games and AI research can greatly benefit from each other. From a research point of view, video games offer fascinating toy examples that capture the complexity of real-world situations while maintaining the controllability and traceability of computer simulations. As an example, consider the problem of driving a racing car under realistic race conditions. While the full problem is too complex to be tackled right now because it involves problems around limited actuators and noisy sensors in addition to the AI problem, important aspects can be tackled working inside a state-of-the-art racing game simulation. As game designers work hard to create more realistic worlds for their customers, AI researchers can benefit from access to benchmarks that accurately reflect real-life problems. Games exhibit many combinations of features that are important in current AI research. For example, a game environment can be either static or dynamic, there can be either single-agent or two-player or multi-agent problems, transitions can be either deterministic or non-deterministic, and game worlds can be either fully known or partially observable. From a games perspective, one key problem is the creation of AI driven agents that can interact with the player and be adaptive so as to create a great interactive gaming experience. These agents can take a variety of roles such as player's opponents, teammates or other non-player characters. Online planning and reinforcement learning have the ability to create adaptive behavior, which might become a key feature in future, games. This is useful to respond to changes in the human player strategy, the environment, the current problem instance, etc. Games like Creatures and Black & White have attempted to build entire games around the concept of teaching behavior to adaptive AI agents. A few concrete examples of AI challenges in games, which we plan to cover in this tutorial, include driving a car in a racing game, path finding on a map, planning the behavior of non-player characters in a role-playing game, resource gathering in a real-time strategy game, and planning the strategy of a combat team in a first-person shooting game. We anticipate that people from the AI community will have a lot to contribute to the field of computer games once the wealth of opportunities in this space has been understood. However, computer games offer a great variety of other challenges including problems in graphics, sound, networking, player rating and matchmaking, interface design, narrative generation, game world design, scripting etc. All of these areas would benefit from various learning and planning paradigms.

## 3. Milestones in the Development of Artificial Intelligence in Games

While discussing the evolution of artificial intelligence in computer games, one definitely should mention the games which have turned out to be milestones in the development of intelligent behavior in games. One of the most popular games of the 1990s was War Craft – a game developed by the Blizzard studio. It was the first game to employ path-finding algorithms at such a grand scale, for hundreds of units in the game engaged in massive battles. SimCity, created by the company Maxis, was the first game to prove the feasibility of using A-Life technologies in the field of computer games. Another milestone turned out to be the game Black and White, created in 2001 by Lion head Studios, in which technologies of computer-controlled characters' learning were used for the first time.

## 4. AI in FPS-type Games

FPS-type games usually implement the layered structure of the artificial intelligence system. Layers located at the very bottom handle the most elementary tasks, such as determining the optimal path to the target (determined by a layer higher up in the hierarchy) or playing appropriate sequences of character animation. The higher levels are responsible for tactical reasoning and selecting the behavior which an AI agent should assume in accordance with its present strategy. Path-finding systems are usually based on graphs describing the world. Each vertex of a graph represents a logical location (such as a room in the building, or a fragment of the battlefield). When ordered to travel to a given point, the AI agent acquires, using the graphs, subsequent navigation points it should consecutively head towards in order to reach the specified target location. Moving between navigation points, the AI system can also use local paths which make it possible to determine an exact path between two navigation points, as well as to avoid dynamically appearing obstacles.

The animation system plays an appropriate sequence of animation at the chosen speed. It should also be able to play different animation sequences for different body parts: for example, a soldier can run and aim at the enemy, and shoot and reload the weapon while still running. Games of this kind often employ the inverted kinematics system. An IK animation system can appropriately calculate the parameters of arm positioning animation so that the hand can grab an object located on, e.g., a table or a shelf. The task of modules from higher layers is to choose the behavior appropriate for the situation – for instance, whether the agent should patrol the area, enter combat, or run through the map in search of an opponent.

Once the AI system has decided which behavior is the most appropriate for the given

Jaya Sachan
Volume: 1 Issue: 1

situation, a lower-level module has to select the best tactics for fulfilling that task. Having received information that the agent should, for instance, fight, it tries to determine the approach that is the best at the moment – e.g., whether we should sneak up on the opponent, hide in a corner and wait for the opponent to present a target of itself, or perhaps just run at him, shooting blindly.

## 5. AI in RTS-type Games

In RTS-type games, it is possible to distinguish several modules of the artificial intelligence system and its layered structure. One of the basic modules is an effective path-finding system – sometimes, it has to find a movement solution for hundreds of units on the map, in split seconds – and there is more to it than merely finding a path from point A to point B, as it is also important to detect collisions and handle the units in the battlefield avoid each other. Such algorithms are typically based on the game map being represented by a rectangular grid, with its mesh representing fixed-sized elements of the area. On higher levels of the AI system's hierarchy, there are modules responsible for economy, development or, very importantly, a module to analyze the game map. It is that module, which analyses the properties of the terrain, and a settlement is built based on the assessment, e.g., whether the settlement is located on an island, thus requiring higher pressure on building a navy. The terrain analyzer decides when cities should be built and how fortifications should be placed.

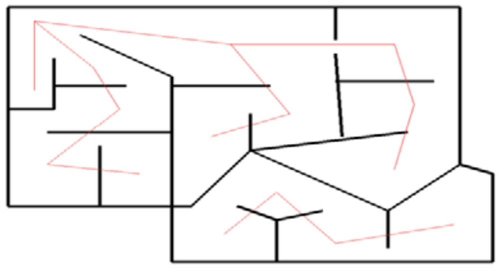

Figure 1: Representation of the world in a RTS-type game

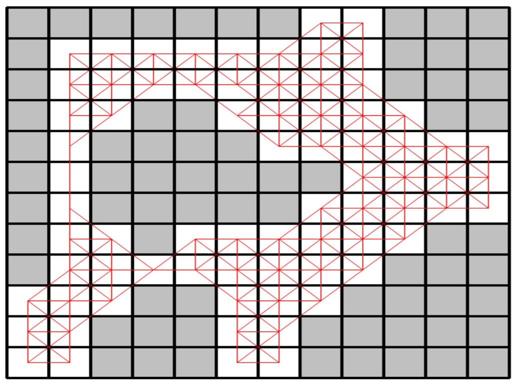

Figure 2: Representation of the world in a FPS-type game

## 6. AI in Sports Games

Basically, in the case of most sports games, we are dealing with large-scale cheating. Take car racing games, for instance. For the needs of the AI, from the geometry of the game map, only and only the polygons belonging to the track of a computer-controlled opponent should travel on and get distinguished. Two curves are then marked on that track: the first represents the optimal driving track, the second – the track used when overtaking opponents. The whole track gets split into appropriately small sectors and, having taken parameters of the surface into account, each element of the split track gets its length calculated. Those fragments are then used to build a graph describing the track, and to obtain characteristics of the road in the vehicle's closest vicinity. In effect, the computer knows it should slow down because it's approaching the curve, or knows that it's approaching an intersection and can, e.g., take a shortcut. Two important attributes of Artificial Intelligence systems in such games is being able to analyze the terrain in order to detect obstacles lying on the road, and strict co-operation with the physics module. The physics module can provide information that the car is skidding, having received which the Artificial Intelligence system should react appropriately and try to get the vehicle's traction back under control.

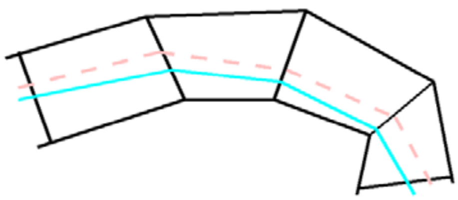

Figure 3: The method of presentation of reality in car race (segmentation and optimalisation of the track)

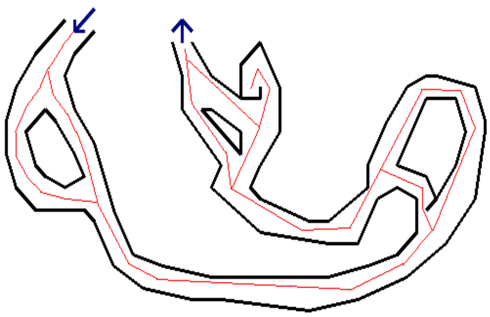

Figure 4: The method of presentation of reality in car race

Similar cheating can also be found in other sports games. In most cases, a computer-controlled player has its complete behavior determined even before the beginning of the turn – that is, it will, e.g., fall over while landing (acrobatics, ski jumping etc.), have the wrong velocity, start false etc. Additionally, in games simulating sports with scoring by judges, the scores are generated according to the rules defined by the appropriate sports bodies.

The predefined scenario of a computer-controlled player is then acted out by the character animation system.

## 7. Choosing an Algorithm

There are a lot of algorithms for finding the optimal path in a graph. The most simple of such algorithms, commonly called fire on the prairie, works by constructing consecutive circles around the starting point, with each step of the algorithm building another, wider circle. Consecutive circles and elements belonging to them are assigned larger and larger indices. As one can see in Figure 5, the circle with index 4 passes through our target point.

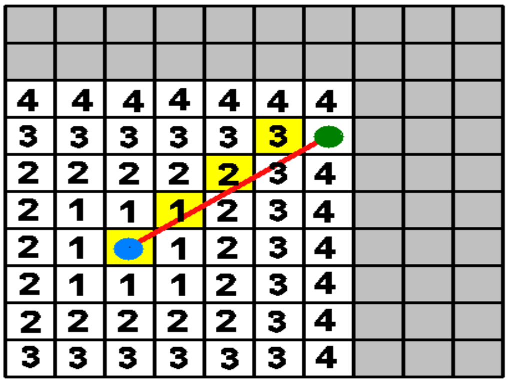


Figure 5: A simple path-finding algorithm

Now, heading in the opposite direction and following the rule that in each step we move to the nearest map point located on the circle with a smaller index, we reach the starting point; the elements of our map we have returned through make up the shortest path between the starting point and the destination. Examining the way this algorithm works, one can see that, in addition to its great advantage – the simplicity – it also possesses a severe drawback. The path the algorithm has found in our example consists of only five elements of the game world, even though 81 fields of the map would have to be examined in the worst-case scenario. In case of a map consisting of 256x256 fields, it might mean having to examine 65536 map elements!

Enter A* and its primary advantage – minimization of areas being examined by consciously orienting the search towards the target. Keeping it brief, I could say that, when calculating the cost of reaching a point on the map, the A* algorithm adds to it some heuristics indicating the estimated cost of reaching the destination; this function is typically the distance to the destination from the point currently being examined.

## 8. References

- Steve Rabin, AI Game Programming Wisdom, ISBN: 1584500778
- Steve Rabin, AI Game Programming Wisdom 2, ISBN: 1584502894
- Game Programming Gems 1, 2 ,3, 4
- Game Developer Magazine
- Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Design Patterns CD – Elements of Reusable Object-Oriented Software