Research on of 3D Game Design and Development Technology

Meng Yang, Zhen Wang, Shilong Xiao
School of Art design
ShenYang Ligong University
Shenyang, China
wangzhen586@yeah.net

***Abstract*-According to the great significance of 3D game technology, this paper will do detailed research on main technologies of 3D. Firstly, this paper puts forward to design concept, designs the main frame of 3D game. Secondly, through researching collision detection technologies, the improved collision detection method is proposed based on Aligned Axis Bounding Box according to some disadvantages of collision detection methods. The improved collision detection method effectively reduces operation time, advanced game speed. Thirdly, through researching artificial intelligence for 3D game, path search method is improved according to the needs of system. The improved idea uses restricting search range of A* search arithmetic with the shortest path. It can get intelligent path search efficiently and factually. Finally, the 3D game system is realized based on 3D engine technologies. The system is rendered smoothly, authentically, manipulated easily.**

Keywords-3D Game; 3D Engine; Collision Detection; Path Search

Collision detection technology and artificial intelligent technology could be applied in the development of game, as well as the areas of education, virtual reality, historic preservation, geo-information system, scientific research, entertainment and business. So, the reinforcing for the research on 3D game technology has great practical significance [1-3]. And the successful research and development of collision detection technology and artificial intelligent technology that are suitable for 3D game will have huge theoretical significance and practical significance, wide application prospects. With the development of 3D game as platform, this article deeply researches collision detection algorithm and artificial intelligent technology in game, focuses on the problem of large calculation amount, slow calculation speed and imprecise collision detection for collision detection algorithm to improve the collision detection in game by the method of combining class-bounding boxes and segment detection method with space partition method, which could effectively promotes the precision and speed of collision detection algorithm. In addition, this article perfects path search algorithm of game, improves the problems of long-time search, slow searching speed in A* path search algorithm by improving the path search method of A* algorithm, restraining search scope and aggravating the weight of heuristic function.

978-1-4244-5540-9/10/$26.00 ©2010 IEEE

I. COLLISION DETECTION ALGORITHM

The BVH（bounding volume hierarchy） in collision detection algorithm is one of the most widely used collision detection methods [4]. The crossing test among BBB is simple, but the compactness of BBB collision detection algorithm is bad. This article makes improvement for bounding volume's compactness and detection method of AABB collision detection algorithm.

Improve the idea: the core of collision detection algorithm is real time and precision. Some game could sacrifice some precision to get real time. The system will adopt collision detection algorithm based on AABB, and improve the problem of bad compactness and large calculation amount existed in AABB collision detection algorithm to certain extent, with the purpose of making the improved collision detection algorithm simple, speedy and precise. Improvement idea mainly includes the following aspects: 1) apply BSP to discompose space of scene to reduce the calculation amount for collision detection; 2) optimize bounding volume; 3)put forward segment collision detection algorithm to promote the speed of AABB collision detection.

BSP decomposition scene: the precision and speed of collision detection is the important property of game. If we suppose there are 100 objects in game world, for RPG game, system has to make collision detection for 100 objects and 100^2 calculations of collision detection. If there are other objects existing in system, too large calculation amount and too bad system performance are difficult to be accepted by players. So, it is necessary to promote the speed of collision detection. This article mainly provides the following two methods to promote the speed of collision detection. Firstly, as above mentioned, if the game scene is not decomposed, collision detection has to detect each object for collision, but in these detections, most of collision detection is useless, so, precious time and resource are wasted for idol work in this way. Then how to reduce this situation? Considering from the aspect of data structure, binominal method shall be adopted to decompose scene. Binominal method could not only be used to manage scene, but also be applied into collision detection. Binominal method could divide object into different area. Each unit contains little object, and collision detection could be done in line with area. BSP tree has the advantage to traverse three dimensional spaces along BSP tree. For each iteration, 50% objects that are impossible

to conflict in tree dimensional space could be eliminated until the coincident object is found, by comparing central sibling nodes of objects. This method is simple and speedy, which just need $\log 2^N$, or even N steps for bad situation, to finish collision detection. Therefore, the adoption of BSP tree for decomposition scene promotes the speed of collision detection to a great extent so as to reduce the calculation amount for collision detection. Secondly, the speed of collision detection could be promoted from the point of space. Game scene includes static and dynamic object. If the bounding volumes for them are updated continuously, this will cost many time to calculate the bounding volume for all the objects, which undoubtedly increases unnecessary calculation. Because the bounding volume of static does not change, so it is not necessary to update their bounding volume. Accordingly, object in static scene has already confirmed its bounding volume in calling stage, and does not update their bounding volumes. In order to guarantee the precision of collision detection, the bounding volume of dynamic player must be updated real time. The adoption of this method could reduce the calculation amount of collision detection algorithm to a very great extent, and promote the running speed of game.

Optimize bounding volume: in order to promote the precision of collision detection, the precise bounding volume is required for collision detection. But game scene includes dynamic role and static object, if the bounding volume of every object is updated real time, it will increase unnecessary time for calculation. Thus, AABB bounding volume for the object in scene shall be bound in line with class when developing this game. For the static object in scene such as wall, house, tree and other objects with regular shape, only one bounding volume shall be bound, and will not be updated. While, the bounding volume of dynamic role is changeable all the times, if the bounding volume is not updated real time, the collision detection in game will be seriously affected. The system just updates the bounding volume of dynamic role real time. The width of bounding volume shall be designed as the maximum stride for the acting movement of role. If the bounding volume is too small, the collision detection will be imprecise, and the collision will shows the phenomenon of distortion; if the bounding volume is too big, the real time of collision will be bad, so, a proper design of bounding volume will produce the effect of getting twofold results with half the effort for the collision detection between objects. The step length of dynamic role also affects the precision of collision detection. If the step length of role is too big, the following situations probably happen: when the previous frame hasn't collide with object, the next frame probably has gone through this object, which means that the penetration situation is very easy to happen for the object smaller than the step length of role. In order to prevent the penetrating, the step length of role shall be increased properly in the process of producing character animation.

Segment collision detection algorithm: game system has low requirement for the precision of collision detection, but has higher requirement for the running speed of game. The static objects in game scene do not need to take collision detection, and the collision would happen only between dynamic role and static object. The scene models of game are mostly regular object, so the adoption of AABB bounding volume is enough to express the object in scene approximately. The collision detection for two AABB objects needs to calculate at least for 6 times, while, if m collision detections are made before collision, 6^m calculations between AABB polygons is needed. This not only has large calculation amount, but also waste of time, because the answer could be obtained only with m calculations if there is no collision. The method for solution is to make crossing test with ray and static AABB before the collision of two objects. In this process, the AABB side that is possible to collide is confirmed, and is not changed into the side of dynamic AABB role and static AABB that is possible to collide until two objects is going to collide so as to make collision detection. However, the opportunity to decide to displace ray with AABB bounding volume is obtained through the distance of ray and AABB bounding volume of static object. With the improved collision detection algorithm, the calculation amount decreases nearly a half. The algorithm is described as follows: 1) select the central point S of bounding volume of player role and the vector V of moving direction and make a ray with (S, V) as parameter; 2) utilize BSP tree decomposition method to quickly find the area that is probably take collision detection; 3) take the collision detection for the ray and static AABB object: ① estimate the distance between ray and static AABB object, estimate AABB side that probably collide, eliminate the side that is impossible to collide, if reach the set threshold, then continue; ② take collision detection for dynamic AABB and static AABB plane, otherwise, turn back to ①.

II. OPTIMIZATION OF ALGORITHM

Large storage space and slow calculation speed is the biggest disadvantage of A* path search algorithm [7,8]. This article adopts three methods to optimize and improve path search. With the method of grading for path search, A* algorithm embodies the intelligence of NPC in path search, but has two defects [9]: A* algorithm needs to maintain heuristic information, present node and expanding node, while, it needs inner storage with exponential quantity for worst situation, and game requires the quickest speed; the starting point and objective point searched by A* algorithm are both static path, but the player role in this system is dynamic, this obviously increases the complexity for path search. A* path search is rather difficult to achieve. So, it is necessary to improve A* path search algorithm.

By observation, we find that all the optimal routes are the straight line between starting point and objective point in the condition of no obstacle. If there is obstacle, each turning point must locate in the peak of convex polygon of obstacle [10]. On the basis of the above idea for searching path, this system adopts grading path search method, which means that non-player role approach object at the path without obstacle, A* algorithm shall be used for path search when the distance between non-player role and obstacle reaches set threshold d, then the optimal node shall be selected real time to change path. For the movement in the straight line of two points, the

extending node is very little, and the calculation amount is very small. Figure 1 is the sketch map for the grading path search method adopted in this article, among which starting point indicates NPC, objective point indicates player role, U is the moving direction of NPC, V is the moving direction of player role. The part from starting point NPC to node adopts path movement of straight line. When the distance between NPC and obstacle reaches threshold d, the improved A* algorithm shall be used to make regular path search. Segment path search method will greatly reduce the searching times, and would not increase the searching time for the increase of distance.

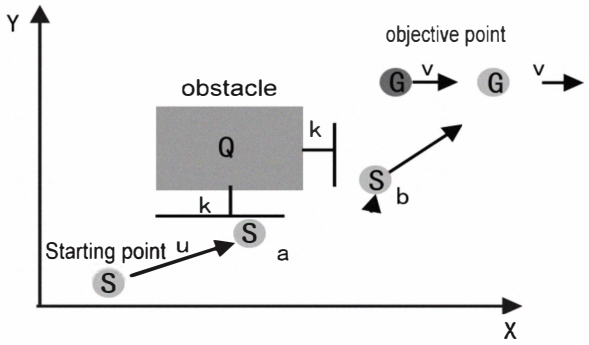

Figure 1: the sketch map of segment path search

The steps of improved algorithm: 1) let coordinate point S of non-player role as starting node and coordinate point G of player role as objective node; 2) calculate d=|G-S|; 3) if d <D (threshold set by people), NPC is activated, otherwise, turn back to 1); 4) path search in straight line, run after player role along the direction of ray; 5) if there is obstacle Q in path, calculate |Q-S|;① if |Q-S|<k((threshold set by people), call the improved A* algorithm; ② otherwise, turn back to 5); 6) return to 4), repeat.

III. EXPERIMENTAL RESULT AND ANALYSIS

Collision detection algorithm: in this procedure, the action responding to collision is that virtual character slides along the edge of the object colliding so as to avoid obstacle. In this process, you have to pay attention to the situation that d is 0. This experiment adopts AABB collision detection algorithm and improved algorithm. Then, compare the experimental data of storage's calling time and game's rendering frequency of two algorithms, where 100 groups of experimental data are collected for provision and the average for random selection of 20 groups is made for comparison. The comparisons of experimental results are shown in figure 2 and figure 3.

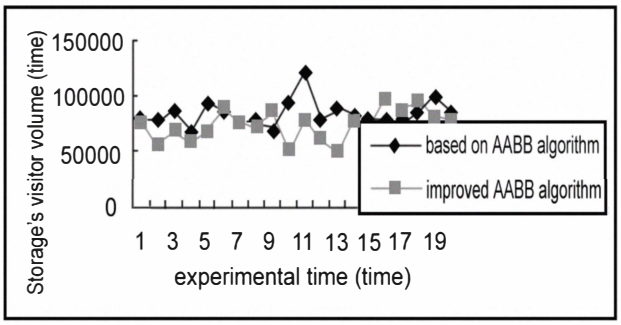

Figure 2, experimental result for the storage visitor volume of the algorithm

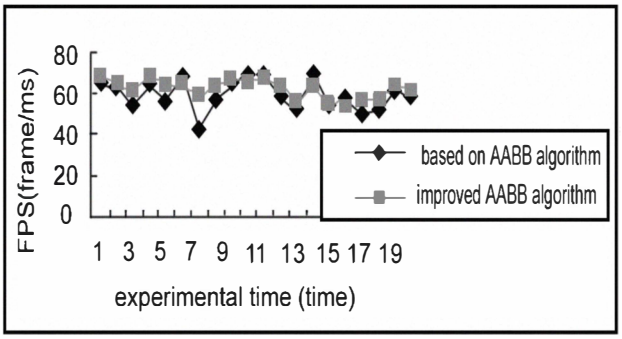

Figure3 experimental result for FPS of the algorithm

Seeing from figure 2experimental data, we could understand that the storage's visitor volume is generally about 90000 times per second when adopting original collision detection algorithm; while, storage's visitor volume is almost between 60000 times and 80000 times when adopting improved collision detection algorithm. The improved collision detection algorithm makes storage's visitor volume reduce. However, whatever storage's visitor volume reduces, it is precious to running. The improved algorithm also promotes game's rendering frequency. From figure 3, we could see the effect of original collision detection algorithm and improved algorithm on game's rendering frequency. Game's rendering frequency value with improved collision detection algorithm is still in the range of the game's rendering frequency value with original algorithm, but we could see from the experimental figure that the whole frequency has promoted. In a word, seeing from the experimental data, we could know that the improved collision detection algorithm plays certain role in promoting game speed.

A* algorithm: this system achieves path search function for non-player role based on C++ language. the path search environment in system is three dimensional space, so the corresponding coordinate data is three dimensional data, which increases more complexity for path search. Array shall be established to preserve the following information for improved A* path search algorithm: the indicator pointing at father node, the cost of present node g(n), total cost f(n), node identifier, the position of player role G, the position of non-player and other data, meanwhile, space shall be allocated for storing activated non-player, threshold d used

by role for path search, threshold k for calling A* algorithm. The data of average running frequency of game is shown in figure 4:

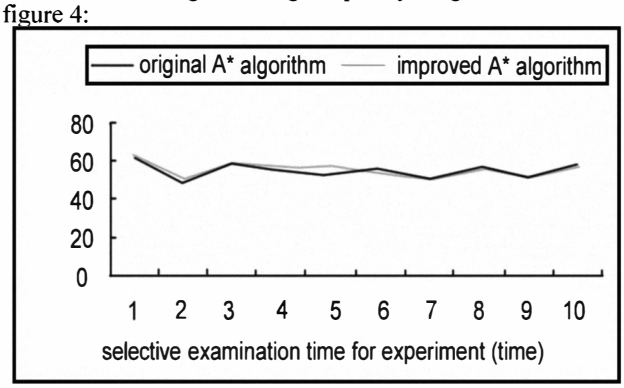

Figure 4, FPS experimental results of path search algorithm

From the experimental data, we could know that if we adopt original A* algorithm for path search, storage's visitor volume is big, and rendering frequency is also affected. But the adoption of improved A* algorithm reduces storage's visitor volume, decrease the occupation of storage space, and promote games'rendering frequency with the effective utilization increasing 34.76% and rendering frequency increasing 2.56%. Seeing the experimental result, we could know that the operational speed of game is fluent, and the rendering of scene is real.

IV. CONCLUSION

This article deeply studies the collision detection technology and artificial intelligent technology in three dimensional game technologies, designs to achieve three dimensional game systems, and puts forward collision detection algorithm based on segment detection by deep research on various collision detection algorithms. The experiment shows that the improved collision detection algorithm could effectively reduce the calculation amount of system and promote operational speed, and calculation amount reduces 11.85%, the operational speed of game promotes 5.33%. For the defects of large storage space, large calculation for node and long time for searching, this article puts forward the strategy of grading path search and improve A* algorithm. The improved algorithm could achieve optimal path search when the game map is not very complex. From the experimental data, we could see that the improved A* path search algorithm could make visitor volume for inner storage reduce 34.76% and the rendering frequency for game promote 2.56%.

REFERENCE

[1] Mobasher B,Cooley R,Jaideep . A Virtual Exhibition Scheme Based on 3D Game Engine [C]. . New York:IEEE Press,1999.

[2] Shahabi C, Zarkesh A M, Adibi J, et al. The Fight Game Based on Actual Person Countenance IEEE Press,2001.

[3] Yan T,JacobesnM,Garcia-Mo Lina H,et Analysis and Design of 3D Game Engine Based on Design Patterns [C]. Proceedings of the 5th Int'l World Wide Web Conf.Paris:Paris WAM Press,1999.

[4] Xu L. Research on Mobile Phone Online Game Development Engine [C]. Hash table. Alaska:Anchorage Press,2001.

[5] Bezdek J C. Design and Realization of 3D Graphic Rendering Engine. Co rnellUniversity, Ithaca, N ew Yo rk, 1973

[6] KamelM and Selim S Z. JOURNAL OF LUOHE VOCATIONAL TECHNOLOGY COLLEGE 2001, 27 (3) : 421～ 428

[7] Zadeh L A. Design and Implementation of Multithread Simulated 3D Shooting Game. 2004, 8: 338 ～ 353

[8] KamelM and Selim S Z. Texture Mapping and Its Application in 3D Game Engine. 2006, 61: 177～ 188

[9] A l- Sultan K S and Selim S Z. The Research of Our Country Forestry Enterprise Management Innovation under the Context of E-business. 1993, 26 (9) : 1357～ 1361

[10] Miyamo to S and N akayama K. Design & Development of an Advanced 3D CAD System Based-ACIS Cybernet. 1986, 16 (3) : 479～ 482

[11] Bezdek J C, Hathaway R, SabinM and TuckerW. Based on game engine 3D graph system research. 2003, 17 (5) : 873～877

[12] J. Greensmith and U. Aickelin, "Texture Mapping and Its Application in 3D Game Engine", Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2007), pp. 49–56.

[13] Aickelin U, Cayzer S, "T An Occlusion Culling Algorithm for Moving Objects in 3D Game Scene," 1st International Conference on AIS, 2002, pp 141-148.

[14] P Matzinger, "Texture Mapping and Its Application in 3D Game Engine," Annual Reviews in Immunology, 2009(12), pp. 991–1045,

[15] J.Greensmith, U.Aickelin and S.Cayzer, "Research on Developing 3D Games Based on Morfit 3D Engine," Proceedings ICARIS-2005