A context aware application developed for stadium path finding

Bing Sun, Chris Todd

Department of Electronic & Electrical Engineering, University College London.

Abstract: This paper introduced an original context aware concept called “smart ticket”. It assists costumers to find their seats effectively and efficiently at a crowding and confusing stadium environment. The scenario has been set to Wembley Stadium, one of the most important places for London Olympic Games 2012. The main goal of this project is to design, implement the smart ticket application and simulate its network performance under this special scenario.

1. Introduction

For countless times audiences come to stadium and get packed up at the entrance and aisles 10 minutes before the event. Their unfamiliarity to the magnificent stadium may cause unbearable congestion, which result in irritation, turbulence, or even terrorism attacks. Smart ticket is a context aware application which solves all these unpleasant problems. It helps to find the best path for the audiences, by utilizing their location, ID, seat, direction.

Figure 1 shows a brief model of Wembley stadium. Around the stadium, hundreds of sensors are placed which catch the audience locations and other context information. The context information comes to many decentralized points which compute the best path for the audience and get back to the screen of their smart ticket.

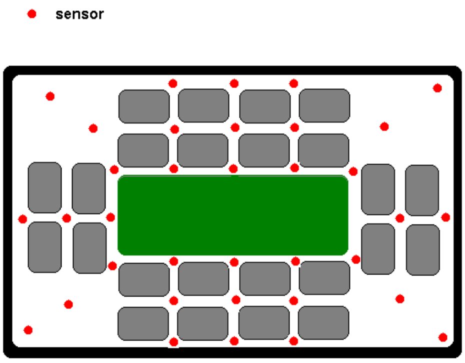

Figure 1 Stadium model

The project, theoretically involve in 3 fields:

Context aware: Computers are not smart as human beings to deal with environmental information, especially interacting with people. So context aware computing tries to help computers become smarter, discovering and utilizing of context information such as audience location, ID, nearby people and traffic. At the primal stage of the project, a handy electronic device (Figure 2) is distributed to every customer at entrance and assists him finding his seat. Later, voice, video and many other multimedia functions are to be realized in the platform to make it more versatile.

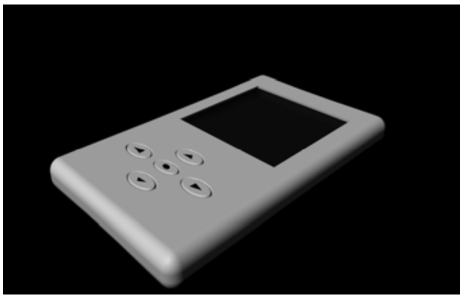

Figure 2: Smart ticket device

Localization: The pivotal information in this scenario is to know where the audience is and where he is going to at the time. The system will make different individual decisions for the audiences, depending where he is and what direction he is heading for.

Path finding: there are various path finding algorithms like A*, Dijkstra's Algorithm, and Best-First-Search. Their advantages fit in different scenarios. A* algorithms is adopted for its fast response and intelligence.

2. The path finding principles

The idea of A* algorithms is not complicated but quite powerful. It takes the essence of both the Dijkstra's Algorithm and Best-First-Search. Figure 3 is a demo version to show how the algorithms fit in the scenario. The map is assumed to be composed of grids.

There are mainly 4 kinds of grid:
The start grid: where the customer stands.
The end grid: the yellow seat he tries to reach.
The walkable grid: the black grid.
The unwalkable grid: the white seat area. It could be walls, stairs in reality.

The task is to find the shortest path from the start grid to the end grid through the walkable areas, in other word, find each suitable grid along the best path. By realizing this, each walkable grid is given two values:

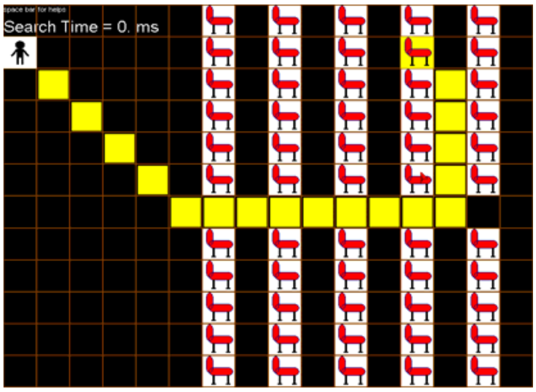

Figure3. Smart ticket demo

G =the route cost from start grid to the current grid considering obstacles. It means how far the customer should walk to the current point.

H =the expected cost between the current and end grid ignoring obstacles. It means the direct distance from the current grid to the end point. This expectation value helps to lead the calculation towards destination not all directions.

F= G + H. F stands for the lowest cost from the source to the sink in a certain sense. Here cost is mainly taken for the length of the route for simple calculation, let alone the terrain factor. Stepping from a lowest cost grid to the next lowest cost grid, then the shortest path is found.

Take the first step for example, the left grid to the person is chosen temporarily for the lowest F value. Then try the lowest cost grid next to it. If a grid is unwalkable, it will go back to the previous grid and try the second lowest grid and compare the cost with the previous route. Finally the shortest path is decided.

The demo version in Figure 3, show how the algorithms works, it marks the shortest path to the yellow seat wherever it is chosen. A timeGetTime( ) function is used before and after the calculation, and the time difference is the service time.

3. Smart ticket applications

The application (figure 4) based on A* path finding algorithms, uses C++, and some developed modules in DirectX.

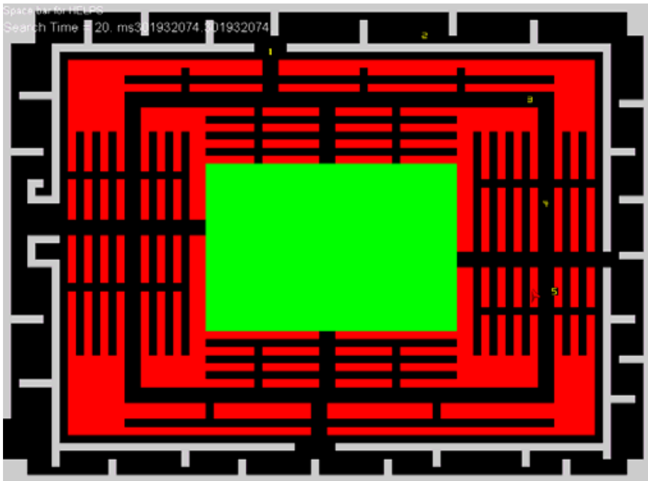

Figure 4. Smart ticket

The map definition is set 80*60, which can be easily implemented into a handset screen and this definition can help to reach an ideal searching speed. (The relationship between map scale and service time is discussed later)

The map can zoom in or out to fit in the smart ticket device. The black area in the map is walkable, and the red seat and white wall is not. Entrance is set at the left bottom. Basically, the customers come from the entrance and are lead to their seat thereafter.

Destination, speed of customer, can be set up before simulation, which can be, in reality, acquired through context information. Although it is complex and interesting, there is no collision factor in this version. For example, when the system expected a collision would happen and informed the audience, it might disappear when the he walked there. So this factor is left for future work.

4. Performance Simulation

According to vast investigations over context aware applications, the crucial attribute of QoS is the end to end delay, which is significantly important at this scenario. If the delay is unbearable, for example, over 1s, customer probably would not follow the instruction and walk their own way. End to end delay mainly includes transmission delay, queuing delay, and application service time. In this paper, the service time is simulated.

4.1 <u>variation of enquiry location</u>

The simulation is performed by the smart ticket application based on the following assumptions:

1). 5 people are set and marked from number 1-5.

2). Start and end point is randomly chosen.

3). Enquiries are responded in a serial FIFO queue.

4). Result is collected averagely through 10 times running.

Results:

If the 5 people starts the same place but ends different, the service time is average 10ms.

If the 5 people starts from different place but ends same, the service time is average10ms.

If the 5 people starts from different place and ends different, the service time is average 50ms.

Above results show service time is very relative to the location of the enquiry. A decentralized networking is adopted in the system. Decentralized servers only serve the customer in certain region, through which service time can be largely reduced, because people enquire from nearly the same place. In other words, calculation loads are much lessened.

4.2 variation of map scale

From above discussion, a conclusion can be drawn that service time is very much related to how the enquiry is initiated. However, Attention should also be drawn to the grid size which is also much related. A similar experiment is done to test the service time under different map scale.

A model is build in the demo version, with start point set to the left top, end point to the right bottom, and the unwalkable area lying in the middle. In this situation, the service time is collected for the worst situation because the system needs to calculate the farthest distance.

Results：

Scale	16*12	40*30	80*60	160*120	200*150	400*300	800*600
Service time (ms)	1	5-10	20	100	180	980	5890

The results above show that service time grows much faster than the scale size, which means proper selection of map scale is very important to the system performance. The service time time is under 20ms when the map definition is lower than 80*60 which is proper for the current system. For a future upgrading concern, if a map needs to reach higher definition like XGA or even UXGA, a more powerful server is needed or a grading computing is needed, which means a coarse computing is done at the beginning, followed by an accurate regional computing.

5. Conclusion

This paper introduced an innovative context aware concept for Wembley stadium, "smart ticket". It developed, designed and analyzed the application from the technical part of view. Through introducing A* algorithms, the service time is controlled within 10ms. And a performance simulation is done and the results offered some valuable evidence to prove the service time is much related with both the enquiry location and the map scale.

References

[1] Mari Korkea-Aho, "Context-Aware Applications Survey", 25/04/2000

[2] S. Reddy, K. Lam, Y. Malidor, C. Martineau, G. Monghal, K. Mohan, "Location information and handover optimization in WLAN/WPAN", 9th semester thesis of Mobile Communication, Aalborg University, autumn 2004

[3] Dey, A.K. and Abowd, G.D. (1999). Toward a better understanding of context and contextawareness, GVU Technical Report GIT-GVU-99-22, College of Computing, Georgia Institute of Technology

[4] D. Wu, "Dijkstra's shortest path algorithm", Hong Kong University of Science & Technology,

[5]Thomas. “H.323: The Multimedia Communications Standard for Local Area Networks”. IEEE Communication Magazine, 1996