COMPUTER UNIVERSITY (MANDALAY)

FINAL YEAR PROJECT REPORT

ON

FINDING THE SHORTEST PATH FOR POPULAR PAGODAS OF BAGAN

Bachelor of Computer Science

(B.C.Sc.)

Presented by Group (17)

2014-2015

Acknowledgements

First of all, we would like to express our gratitude to **Dr. Win Aye**, Rector, Computer University (Mandalay).

We would like to offer our special acknowledgement to my supervisor, **Daw Aye Aye Maw**, Assistant Lecturer, Infromation Science Department, Computer University (Mandalay), for their valuable guidance, kind encouragement, and technical assistance.

And then, we also would like to describe our sincere thank **Dr. Aye Aye Chaw**, Associate Professor, Dean of Project, Software Department and Daw Wai Wai Myint, Tutor, English Department, Computer University (Mandalay), for her helpful recommendations and suggestions.

We also would like to special thank **U Thaung Kyaw**, Associate Professor, Head of English Department, Computer University (Mandalay), for editing our project from the English language point of view.

We would like to express our grateful thank to all teachers at Computer University (Mandalay) for their helpful advice.

Finally, we are very grateful to all our teachers, friends and colleagues of Computer University (Mandalay) for their cooperation and help to complete this project successfully.

Group Member List

Sr.No	Name	Roll No.
1	Ma Ei Cho Zin	4CS-11
2	Ma Kyu Kyu Moe	4CS-94
3	Ma Nang Seng Aye	4CS-146
4	Ma Khin Cho Thet	4CS-180

Supervisor

maw 28-9.2015

Name :Daw Aye Aye Maw

Rank :Assistant Lecturer

Department :Information Science Department

Computer University (Mandalay)

Project Schedule

Project Proposal :March

First Seminar :2.6.1015

Second Seminar :7.7.2015

Third Seminar :11.8.2015

Book Submission :September, 2015

Time Schedule	March 2015	May 2015	June 2015	July 2015	August 2015
Project Proposal					
First Seminar					
Second Seminar					
Third Seminar					
Book Submission					

ABSTRACT

Artificial Intelligence is defined generally as the attempt to construct mechanism performed by humans. The knowledge-base contains knowledge necessary for understanding, formulating, and solving problems. This system is intended to present Optimal Route finding system for Bagan. It is used to operate in complicated mapping situations and new unknown environments. This technique can be implemented to find the optimal path from source to destination in region of map. This system uses SMA* algorithms to assist optimal BAGAN route finding and give the optimal route. It will find the shortest path of Bagan route by SMA* algorithms. This system is intended to help visitors by finding the shortest path and displaying the attribute information. The optimal route based on distance or time or cost. This algorithm is complete and optimal. This system is implemented by using Java Programming Language.

List of Figures

List of Tables

CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Introduction

Artificial Intelligence (AI) is a subdivision of computer science devoted to create computer software and hardware that attempt to produce results such as those produced by people. Computer already emulates some of the simple activities of the human mind. This can perform mathematical calculation, manipulate numbers and letters, make simple decision, and perform various storage and retrieval function. Artificial Intelligence (AI) combines ideas from philosophy, anthropology, psychology and neurobiology with computer-based tools and formal mathematical methods in order to come up with machines that deal with more and more complex problems. Many important tasks, such as planning, parsing, and sequence alignment, can be represented as shortest-path problems.

Therefore, this system solve by finding the sequence of actions that lead to desirable goal. Searching methods are used in finding the path from a given start point to some goal / target node. Each visitor needs an optimal route of how to most efficiently visit each famous in Bagan. A* search evaluates nodes by combining g(n), the cost to reach the node, and h(n), estimated cost to the goal from n. f(n)=g(n)+h(n).f(n),estimated total cost of path through n to the goal. SMA* is well and simple, proceeds just like A*, expanding the best leaf until memory is full. At this point, it cannot add a new node to the search tree without dropping an old one. SMA* method is used to find the best solution in this system and also evaluated the performance of this method.

1.2 Objectives of the project

Objectives of the project are:

- To study and understand SMA* method.
- To find the optimal route for Bagan Pagoda by using SMA* method.
- To travel within the short time and reduce the cost.
- To reach the target place easily.

1.3 Project Requirements

1.3.1 Hardware Requirements

1. Four PC and one Server Computer,
2. Intel(R Core(TM) i5 or i7-3230M CPU @2.6 GHz,
3. 32 bit Operating System
4. 4.00 GB memory (RAM)
5. 500 or 1000 GB Hard disk drive,
6. Gigabit Ethernet Controller for Network Adapter,
7. Display devices: Monitor and Cannon Printer and Input Devices

1.3.2 Software Requirements

- Java Software Developer Kid
- Eclipse (Indigo)
- Microsoft SQL Server
- Apache Tomcat

CHAPTER 2
THEORY BACKGROUND

This section describes the theory used to apply the finding the shortest path of the popular places of Bagan by applying SMA* method.

2.1 Artificial intelligence

Artificial Intelligence is the science and engineering of making intelligent machines, especially intelligent computer programs. The computer is exceptional and usually exceeds humans in performance. Computers greatly speed up and simplify some aspects of the human thought process. AI technologies are being integrated with other computer-based information systems so that the capabilities and applicability of computers are greatly increased.

2.2 Problem solving

Problem solving is often associated with the performance output of thinking creatures. It is a mental activity of finding a solution to a problem The term problem solving was introduced by mathematicians. Six major steps are generally observable in the process: problem identification and definition, identification of criteria for finding a solution, generation of alternatives, search for solution and evaluation, choice and recommendation, and implementation

2.3 Searching methods for finding optimal path

2.3.1 A* Searching method

The most widely-known form of best-first search is called A* search. It evaluates nodes by combining g(n),the cost to reach the node, and h(n),estimated cost to the goal from n. f(n)=g(n)+h(n).To find the cheapest solution, reasonable thing to try first is the node with the lowest value of g(n)+h(n). A* is both complete and optimal.

2.3.2 SMA* Searching method

SMA* always drops the worst leaf node- the one with the highest f-value. Like recursive best-first search(RBFS) ,SMA* then backs up the value of the forgotten node to its parents .In this way, the ancestor of a forgotten sub-tree knows the quality of the best path in that sub-tree. With this information, SMA* regenerates the sub-tree only when all other paths have been shown to look worse than the path it has forgotten. SMA* expands the best leaf and deletes the worst leaf.

2.4 Searching Strategies

Searches are broken into two main categories: uninformed searches (brute-force, blind), and informed (heuristic, directed) searches. Uninformed searches are done when there is no information about a preferred search path. Informed searches have some information to help pick search paths; usually a rule of thumb is used to reduce the search area.

2.4.1 Uninformed search

These search strategies that come under the heading of uninformed search are also called blind search. The term means that they have no additional information about states beyond provided in the problem definition. These strategies can generate successors and distinguish a goal state from a non-goal state. All search strategies are distinguished by the order in which nodes are expanded.

2.4.2 Informed (heuristic) search

Heuristic search is a useful tool in solving planning problems. Reducing a planning problem into a graph search problem, the basic aim of heuristic search is to simplify the process of solving the problem by subtracting some unnecessary yet time-consuming computations, usually in the form of a state expansion. Informed search is the one that uses problem-specific knowledge beyond the problem definition. It can find solution more efficiently than the uninformed search.

2.5 Path Finding Problem

It is based on search method. Search methods aren't the perfect solution for every problem, but with creative applications it can solve many. There are various search methods. These methods can use to solve path-finding problem. Path-finding problem needs to search a path or away from the start point to the goal point through some constraint may exist [6]. This system provides the users the option of selecting their origin or destination on the map of Bagan, manually entering an address or selecting a land map from pull-down menu. The routing algorithms finds the optimum path and output is presented to the user both in text and mark on the map.

2.5.1 SMA* (simplified memory bounded A*) algorithm

It uses all available memory to carry out the search. SMA* expands nodes in the some order as A* until it runs out of memory. At this point, the least promising node is deleted and its f-value is backed up in its parent .SMA* is guaranteed to generate the some node as A* and in the same order. However, it may generate some intermediate nodes multiple times as it "forgets "and "remember" portions of the tree. Searching with SMA* starts just like A*, and continues like A* until the available memory is full, then the node with the worst f-value is removed. If all nodes have equally good f-value, the oldest is removed and the newest is expanded [7]. When removing the node, the values of node are stored in its parent, so that the quality of that path will still be known even though the exact path is not known. SMA* 'algorithm is given below.

```
Function: SMA*(problem) return a solution sequence
Input: problem, a problem
Static: Queue, a queue of nodes ordered by f-cost m
Queue: MAKE_QUEUE ({MAKE_NODE (INITIAL_STATE
problem])})
Loop do
      If Queue is empty then return failure
      n ←— deepest least f-cost node in Queue
   if GOAL_TEST(n)then return success
      s ←— NEXT-SUCCESSOR(n)
      if s is not a goal and is at maximum depth then
      f(s)← infinity
      else
      f(s)←_ MAX(f(n),g(s)+h(s))
      if all of n's successors have been generated then
        update n's f-cost and those of its ancestors if necessary
      if SUCSSSORS(n) all in memory then
        remove n from Queue
      if memory is full then
        delete shallowest highest f-cost node in Queue
        remove it from its parent's successor list
```

insert its parent on Queue if necessary

inserts on Queue

end

2.6 Properties of SMA* method

SMA* has the following properties. It will utilize whatever memory is made available to it. It avoids repeated states as far as its memory allows. It is complete if the available memory is sufficient to store the shallowest optimal solution path. It is optimal if enough memory is available to store the shallowest optimal solution path. Otherwise, it returns the best solution that can be reached with the available memory. When enough memory is available for the entire search tree, the search is optimally efficient.

2.7 Complexity of SMA* method

The SMA* algorithm is optimal and complete if enough memory is available, it returns the best solution that can be found using the given memory variable, MAX nodes is used to represent the maximum number of nodes fit into memory. If there is no memory left, and the algorithm still needs to drops the highest f-value [4].

The space complexity of SMA* is *O(M)*. Make use of all available memory *M* to carry out A*.*M* is the given memory limit. SMA* is used all memory available. The time complexity of SMA* is the worse case of $O(b^d)$. The symbol b is the maximum number of successors of any node; d, is the depth of the shallowest goal node. It is complete and optimal if there is a solution reachable. It might be the best general-purpose algorithm for finding optimal solutions.

2.8 Implementation of SMA* method

SMA* expands nodes in the same order as A* until it runs out of memory. At this point, the least promising node is deleted and its f-value is backed up in its parent .SMA* is guaranteed to generate the some node as A* and in the same order. However, it may generate some intermediate nodes multiple times as it "forgets "and "remember" portions of the tree.

Searching with SMA* starts just like A*, and continues like A* until the available memory is full, then the node with the worst f-value is removed. If all nodes have equally good f-value, the oldest is removed and the newest is expanded. When removing the node, the values of it are stored in its parent, so that the quality of that path will still be known even though the exact path is not known.

One important things to consider with SMA* is that a solution can be unreachable; For example, if available memory is three nodes and the shortest path is five nodes then it will be impossible for the algorithm to find a solution. The fact that SMA* removes nodes can result in the need to revisit nodes. It generates the sub tree only when all other paths have been shown to look worse than the path it has forgotten. It complete and optimal if there is a solution reachable.

Using more memory can only improve search efficiency – one could always ignore the additional space, but usually it is better to remember a node than to have to regenerate it when needed.

CHAPTER 3
DESIGN AND IMPLEMENTATION

3.1 System Overview

This system is a computer system capable of capturing, analyzing, and displaying geographically referenced information of interesting places in Bagan; that is, data identified according to location. This technology integrates common database operations such as query and statistical analysis with the unique visualization and geographic analysis benefits offered by Bagan Map. In this system, user can select source and destination of place on the Bagan Map. The system drawn the shortest route on the Bagan Map and evaluates based on distance, time and cost. SMA* are optimally efficient for any given heuristic function. Therefore, this system used SMA* algorithm for the shortest path problem.

3.2 System Flow Diagram

The system searches the shortest path from the source to the destination. And then, the system shows the possible ways and shortest path from the source to the destination. The system also shows the information of Bagan pagodas.

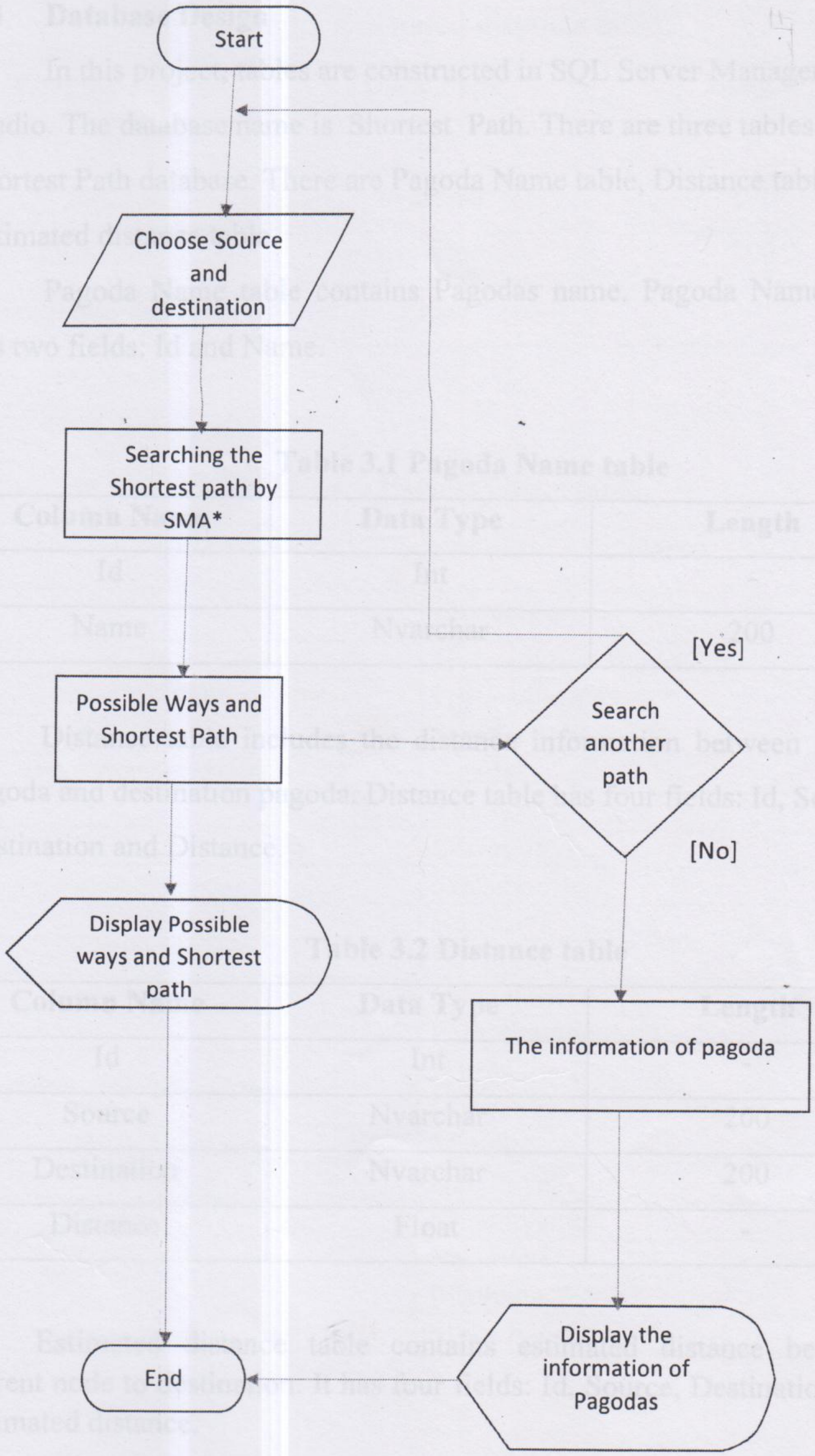


Figure (3.1) System Flow Diagram

3.3 Database Design

In this project, tables are constructed in SQL Server Management Studio. The database name is Shortest Path. There are three tables in the Shortest Path database. There are Pagoda Name table, Distance table and Estimated distance table.

Pagoda Name table contains Pagodas name. Pagoda Name table has two fields: Id and Name.

Table 3.1 Pagoda Name table

Column Name	**Data Type**	**Length**
Id	Int	-
Name	Nvarchar	200

Distance table includes the distance information between source pagoda and destination pagoda. Distance table has four fields: Id, Source , Destination and Distance.

Table 3.2 Distance table

Column Name	**Data Type**	**Length**
Id	Int	-
Source	Nvarchar	200
Destination	Nvarchar	200
Distance	Float	-

Estimated distance table contains estimated distance between current node to destination. It has four fields: Id, Source, Destination and Estimated distance.

Table 3.3 Estimated distance table

Column Name	Data Type	Length
Id	Int	-
Source	nvarchar	200
Destination	nvarchar	200
Estimated distance	Float	-

3.4 Data Set Tables

Table 3.4 Pagoda Name table

Id	Name
1	Shwezigon
2	Gubyauknge Wetkyiinn
3	Htilo Minlo
4	Upali Thein
5	Anada
6	Thabyinyu
7	Shwegugyi
8	Mahabodi
9	Bu Phaya
10	Gawdawpalin
11	Mingalarzadi
12	Pahtothamya
13	Nathlaung Kyaung
14	Gubyaukgyi Myinkabar
15	Manuha
16	Nan Paya
17	Dhamayangyi
18	Sulamani
19	Pyathada
20	Shwe San Daw

Table 3.4 The Distance from Shwezigon to Mahabodi Table

1	Shwezigon	Gubyauknge Wetkyiinn	4
2	Shwezigon	Htilo Minlo	6
3	Gubyauknge Wetkyiinn	Shwezigon	4
4	Gubyauknge Wetkyiinn	Htilo Minlo	2.7
5	Gubyauknge Wetkyiinn	Sulamani	6.1
6	Htilo Minlo	Shwezigon	6
7	Htilo Minlo	Gubyauknge Wetkyiinn	2.7
8	Htilo Minlo	Upali Thein	0.6
9	Htilo Minlo	Sulamani	4.5
10	Upali Thein	Htilo Minlo	0.6
11	Upali Thein	Shwegugyi	5
12	Upali Thein	Anada	2.5
13	Anada	Upali Thein	2.5
14	Anada	Thabyinyu	2.1
15	Anada	Dhamayangyi	5
16	Thabyinyu	Anada	2.1
17	Thabyinyu	Shwegugyi	0.8
18	Thabyinyu	Mingalarzadi	1.7
19	Shwegugyi	Mahabodi	1.6
20	Shwegugyi	Upali Thein	5
21	Shwegugyi	Thabyinyu	0.8
22	Shwegugyi	Mingalarzadi	1.2
23	Mahabodi	Bu Phaya	0.8
24	Mahabodi	Gawdawpalin	1.7

Table 3.4 The Distance from Mahabodi to Shwe San Daw Table

25	Mahabodi	Shwegugyi	1.6
26	Bu Phaya	Gawdawpalin	1.3
27	Bu Phaya	Mahabodi	0.8
28	Gawdawpalin	Bu Phaya	1.3
29	Gawdawpalin	Mahabodi	1.7
30	Mingalarzadi	Shwegugyi	1.2
31	Mingalarzadi	Thabyinyu	1.7
32	Mingalarzadi	Pahtothamya	0.7
33	Pahtothamya	Mingalarzadi	0.7
34	Pahtothamya	Nathlaung Kyaung	1.9
35	Pahtothamya	Gubyaukgyi Myinkabar	2.2
36	Nathlaung Kyaung	Pahtothamya	1.9
37	Nathlaung Kyaung	Shwe San Daw	1.3
38	Gubyaukgyi Myinkabar	Pahtothamya	2.2
39	Gubyaukgyi Myinkabar	Shwe San Daw	2.4
40	Gubyaukgyi Myinkabar	Manuha	1.8
41	Manuha	Gubyaukgyi Myinkabar	1.8
42	Manuha	Nan Paya	0.5
43	Nan Paya	Manuha	0.5
44	Dhamayangyi	Ananda	5
45	Dhamayangyi	Pyathada	3.5
46	Sulamani	Gubyauknge Wetkyiinn	6.1
47	Sulamani	Htilo Minlo	4.5
48	Sulamani	Pyathada	2
49	Pyathada	Sulamani	2
50	Pyathada	Dhamayangyi	3.5
51	Shwe San Daw	Nathlaung Kyaung	1.3
52	Shwe San Daw	Gubyaukgyi Myinkabar	2.4

Table 3.5 Estimated Distance Table From Shwezigon to Pagodas

Id	Source	Destination	Distance
1	Shwezigon	Gubyauknge Wetkyiinn	4
2	Shwezigon	Htilo Minlo	6
3	Shwezigon	Upali Thein	6.5
4	Shwezigon	Anada	9
5	Shwezigon	Thabyinyu	11.5
6	Shwezigon	Shwegugyi	12
7	Shwezigon	Mahabodi	11.2
8	Shwezigon	Bu Phaya	11.9
9	Shwezigon	Gawdawpalin	13
10	Shwezigon	Mingalarzadi	13.5
	.....		
	....		
	.....		
	.....		
15	Shwezigon	Nan Paya	18.5
16	Shwezigon	Dhamayangyi	13.6
17	Shwezigon	Sulamani	10.5
18	Shwezigon	Pyathada	11.8
19	Shwezigon	Shwe San Daw	1

Table 3.5 Estimated Distance Table From Gubyaukng to Pagodas

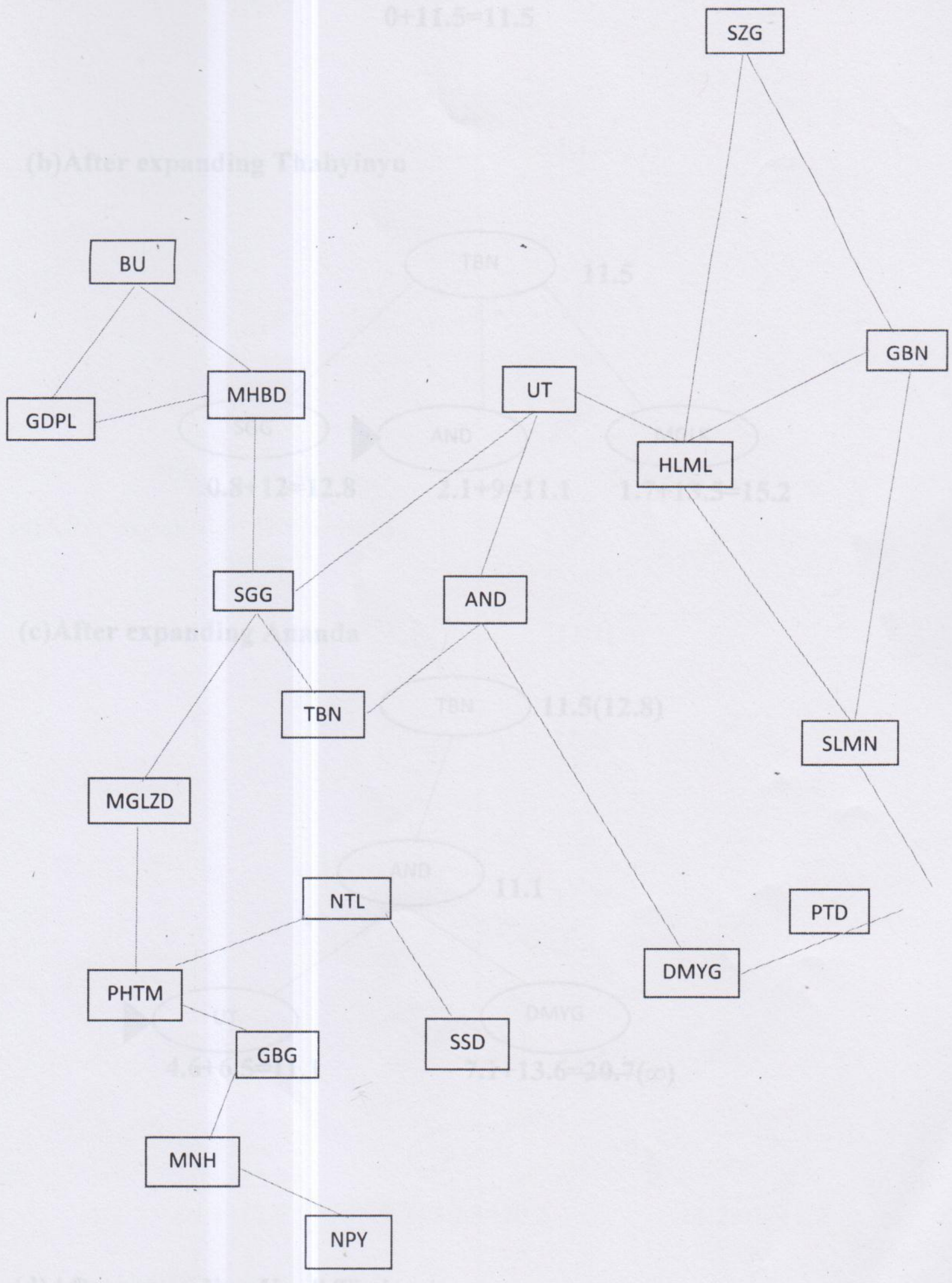
1	Gubyauknge Wetkyiinn	Htilo Minlo	2.7
2	Gubyauknge Wetkyiinn	Upali Thein	3.5
3	Gubyauknge Wetkyiinn	Anada	5.5
4	Gubyauknge Wetkyiinn	Thabyinyu	8.2
5	Gubyauknge Wetkyiinn	Shwegugyi	9
6	Gubyauknge Wetkyiinn	Mahabodi	8.5
7	Gubyauknge Wetkyiinn	Bu Phaya	8.9
8	Gubyauknge Wetkyiinn	Gawdawpalin	10
9	Gubyauknge Wetkyiinn	Mingalarzadi	10.1
10	Gubyauknge Wetkyiinn	Pahtothamya	10.2
11	Gubyauknge Wetkyiinn	Nathlaung Kyaung	9
12	Gubyauknge Wetkyiinn	Gubyaukgyi Myinkabar	11.5
13	Gubyauknge Wetkyiinn	Manuha	13.4
14	Gubyauknge Wetkyiinn	Nan Paya	14.3
15	Gubyauknge Wetkyiinn	Dhamayangyi	9.2
16	Gubyauknge Wetkyiinn	Sulamani	6.1
17	Gubyauknge Wetkyiinn	Pyathada	7.8
18	Gubyauknge Wetkyiinn	Shwe San Daw	9.6

Table 3.5 Estimated Distance Table

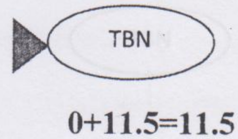
1	Htilo Minlo	Upali Thein	0.6
2	Htilo Minlo	Anada	2.5
3	Upali Thein	Anada	2.5
4	Upali Thein	Thabyinyu	2.9
5	Anada	Thabyinyu	2.1
6	Anada	Shwegugyi	3.4
7	Thabyinyu	Shwegugyi	0.8
8	Thabyinyu	Mahabodi	2.5
9	Shwegugyi	Mahabodi	1.6
10	Shwegugyi	Bu Phaya	3
11	Mahabodi	Bu Phaya	0.8
12	Mahabodi	Gawdawpalin	1.7
13	Bu Phaya	Gawdawpalin	1.3
14	Bu Phaya	Mingalarzadi	4
15	Gawdawpalin	Mingalarzadi	2.5
16	Gawdawpalin	Pahtothamya	3.5
17	....		
18	....		
19	Dhamayangyi	Shwe San Daw	2
20	Sulamani	Pyathada	2
21	Sulamani	Shwe San Daw	4.5
22	Pyathada	Shwe San Daw	6.3

3.5 Case Study

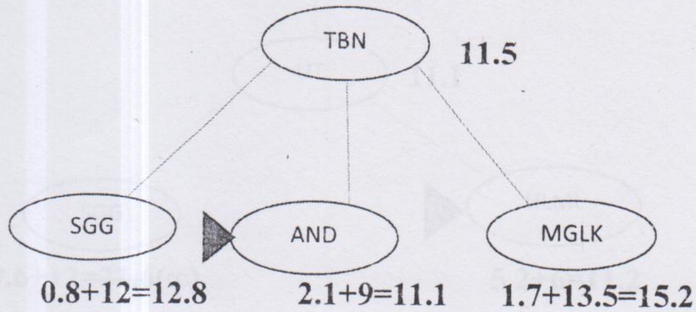
The user go from TBN to SZG,



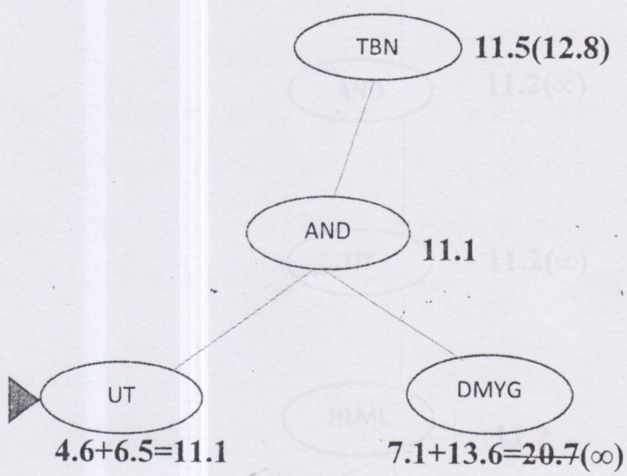
(a)The initial state



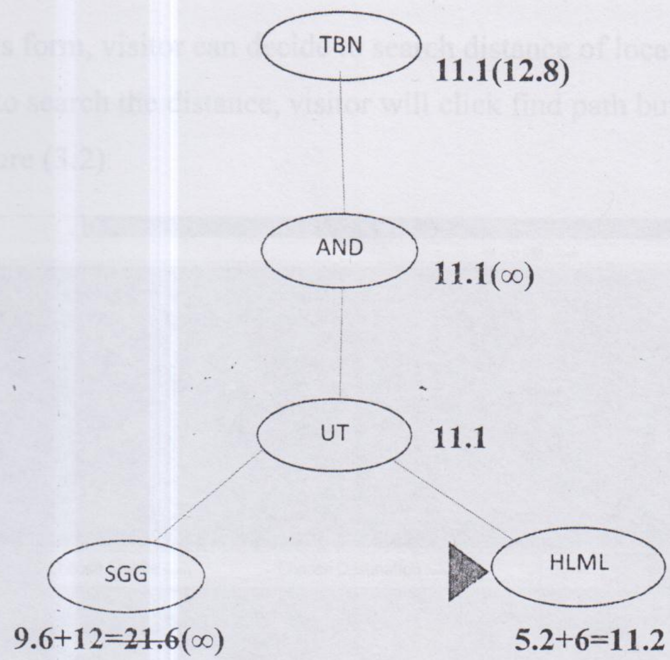
(b)After expanding Thabyinyu



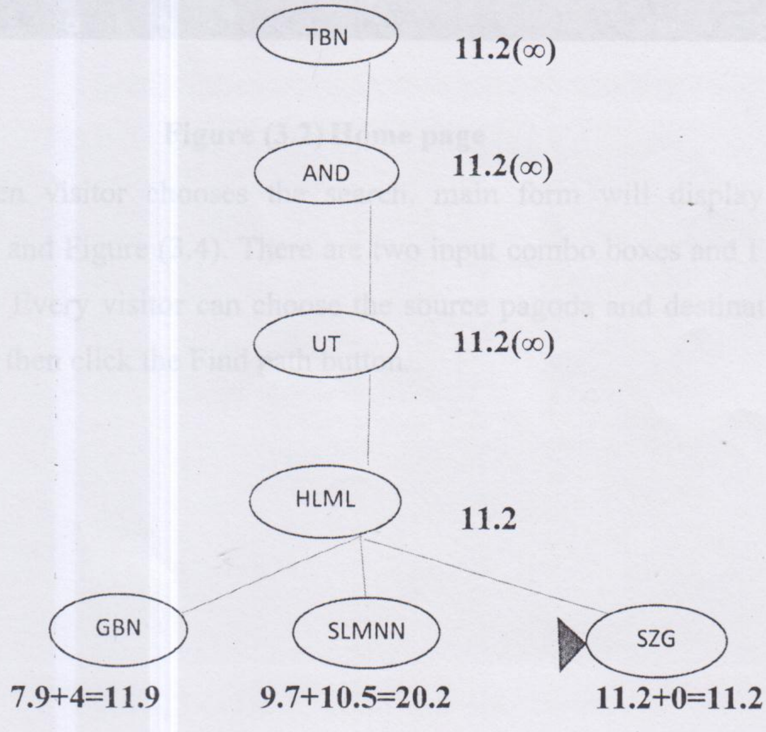
(c)After expanding Ananda



(d)After expanding Upali Thein



(e)After expanding Htilo minlo



3.6 Implementation of System

In this form, visitor can decide to search distance of location. If user wants to search the distance, visitor will click find path button as show in figure (3.2).

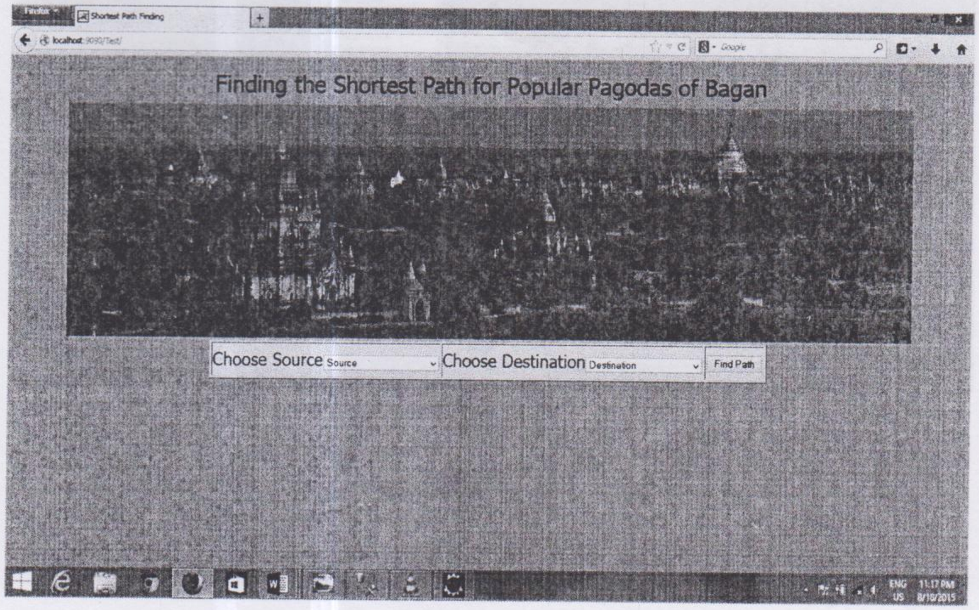


Figure (3.2) Home page

When visitor chooses the search, main form will display in Figure (3.3) and Figure (3.4). There are two input combo boxes and Find Path button. Every visitor can choose the source pagoda and destination pagoda, and then click the Find path button.

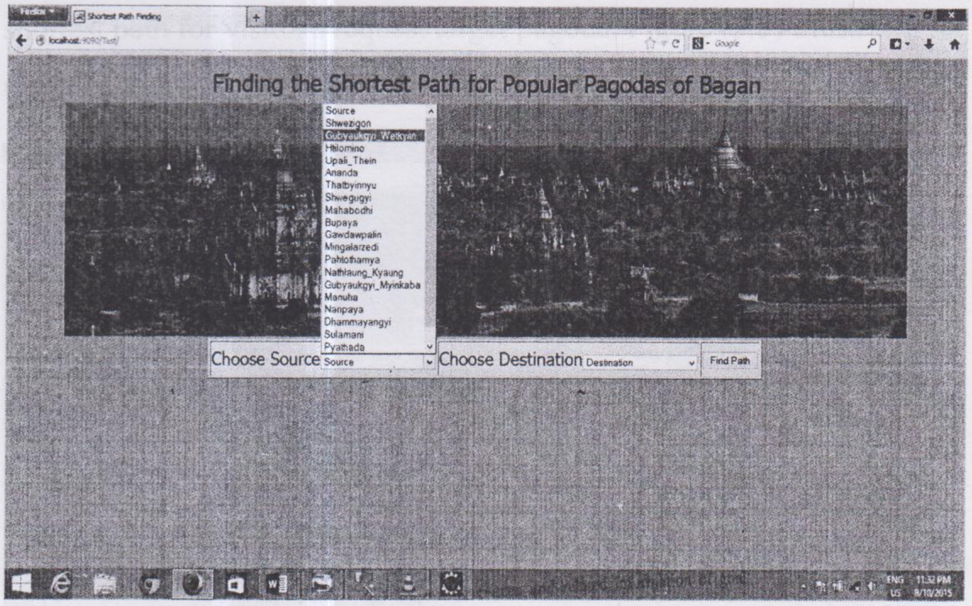


Figure (3.3) Source choosing form

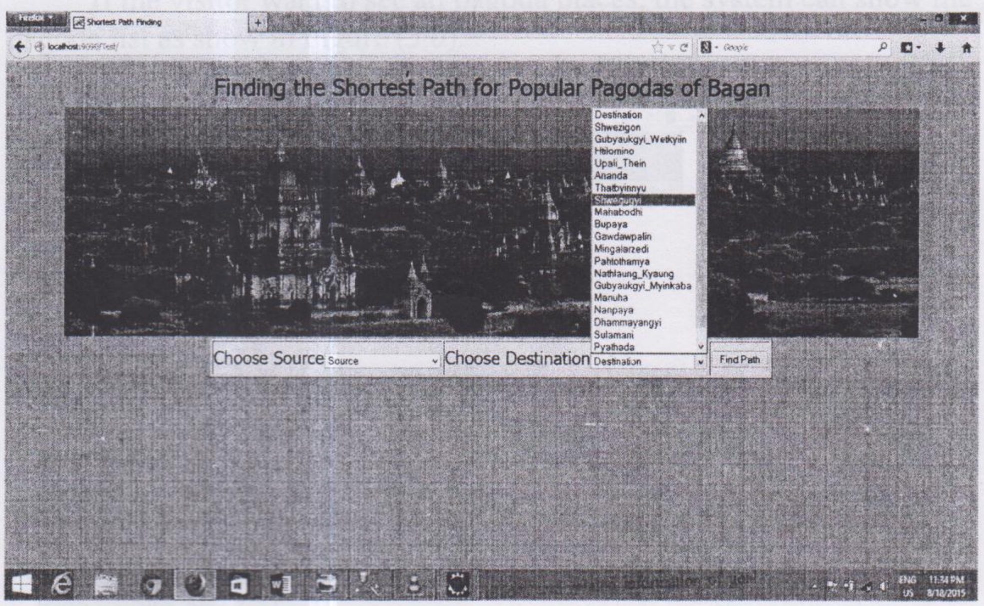


Figure (3.4) Destination choosing form

After searching, the system can display the possible ways and shortest path for visitor's request target pagoda. Figure (3.5) illustrates for result route.

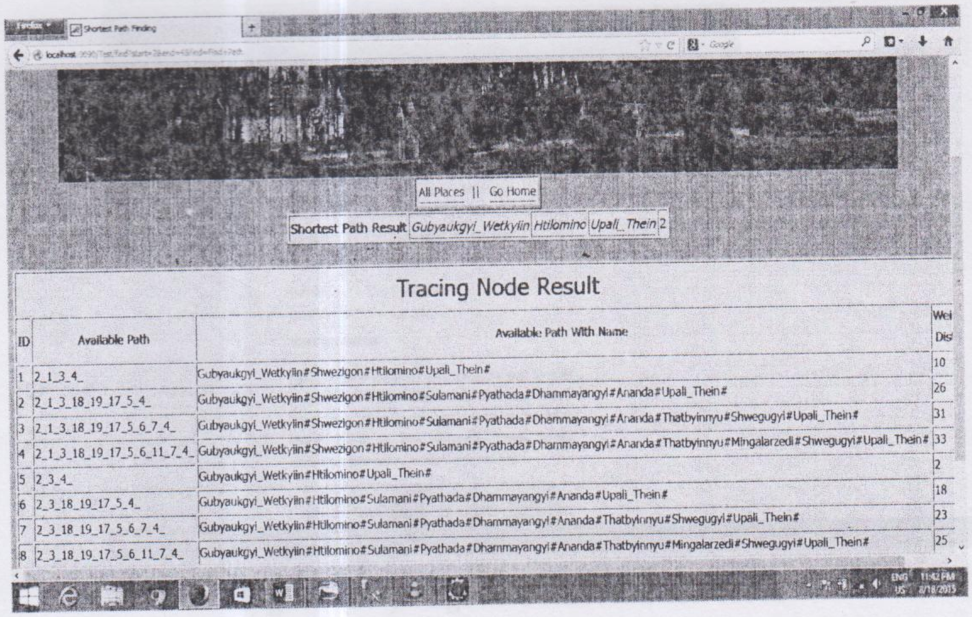


Figure (3.5) Possible ways and shortest path form

If the visitor want to see all pagoda places, the system can show the pagoda list as show in Figure (3.6).

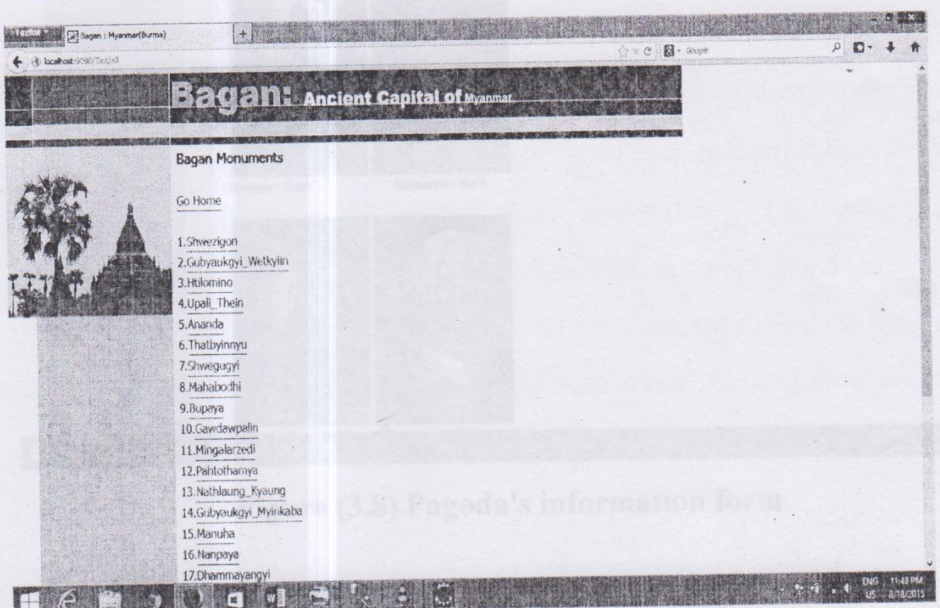


Figure (3.6) Pagodas list form

If the user clicks the pagoda name, the system can also show the pagoda's information as Figure (3.7) and (3.8).

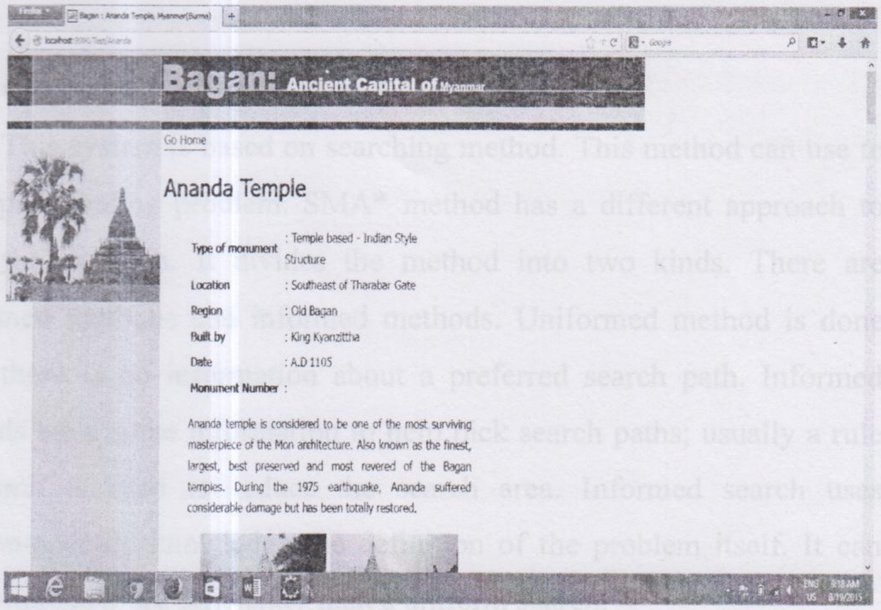


Figure (3.7) Pagoda's information form

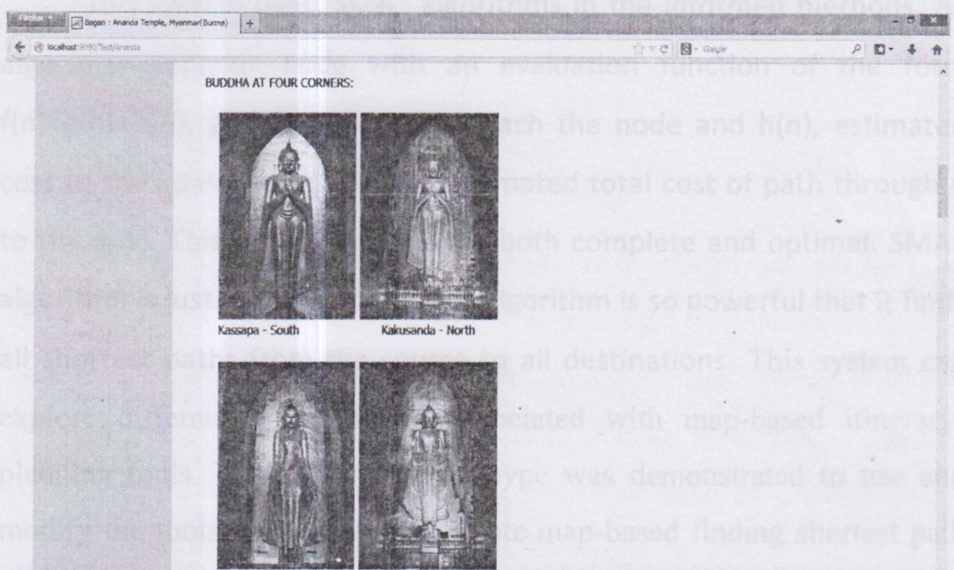


Figure (3.8) Pagoda's information form

CHAPTER 4
CONCLUSION

4.1 Conclusion

This system is based on searching method. This method can use to solve path-finding problem. SMA* method has a different approach to solve the problem. It divides the method into two kinds. There are uniformed methods and informed methods. Uniformed method is done when there is no information about a preferred search path. Informed methods have some information to help pick search paths; usually a rule of thumb is used to reduce the search area. Informed search uses problem-specific knowledge the definition of the problem itself. It can find solutions more efficiently than a uniform search.

This system uses SMA* algorithms in the informed methods. A* algorithm expands node with an evaluation function of the form f(n)=g(n)+h(n), g(n) is the cost to reach the node and h(n), estimated cost to the goal from n and f(n), estimated total cost of path through n to the goal. This search algorithm is both complete and optimal. SMA* algorithm is just like A*. In fact, the algorithm is so powerful that it finds all shortest paths from the source to all destinations. This system can explore different design issues associated with map-based itinerary-planning tools. In this way, a prototype was demonstrated to use and modify the tools and software to create map-based finding shortest path system for Bagan.

4.2 Advantages of the Project

This system can efficiently save the time cost for the visitors in Bagan. This system shows the shortest path from the source and destination. Therefore, the visitor can know the path in immediately. The user can know the cost of estimated traveling time, distance of path from the source to the destination, estimate of traveling cost.

4.3 Limitations and Further Extension

4.3.1 Limitations

This system is implemented by using map-based off line system. This system can develop to online observation system. The limitation of this system is that this system doesn't provide the use of various maps that are small scale map, large scale map and the user cannot change the profile of the system.

4.3.2 Further Extension

In the future work, this system will extend to develop the Geographical Information Systems (GIS) of Bagan and should be applied with small scale and large scale maps. And also, this system extend to solve shortest path problem with Geographical Information Systems (GIS) and should be applied other applications such as internet packet routing, flight reservations and driving directions, mobile.

REFERENCES

[1] AIMI Slike c Stuart Russell and Peter Norving ,1988,BernhandPfahringer 2001-Extra material:informed memory bounded search

[2] Stuart Russell. Peter Norvig, "Artificial Intelligence a Modern Approach", Second Edition, Prentice Hall Series in Artificial Intelligence.

[3] Louis E.Frenzel, Jr. "Expert System and Applied Artificial Intelligence", California State University at Long Beach

[4] Nils J.Nilsson, "Artificial Intelligence", Stanford University, Morgan Kaufmann Publishers, 1998.

[5] Email Kelevedjiev "Computational Approach for Assessment of Critical Infrastructure Network System" Institute of Mathematics and Informatics Bulgarian Academy of Sciences

[6] ThriMon,Mya Than Hinin ,"Comparison of Three Searching method for traveling"

[7] Thae Zon Ei , "Comparison of A* and SMA* "

[8] Pristley, Practical object-oriented design with UML, 2000.

[9] Rational Rose Suite, Rational Rose RT and other software from Rational Software Corporation.

[10] Real-time Studio, ARTiSAN Software Tools, Inc. July 2001.