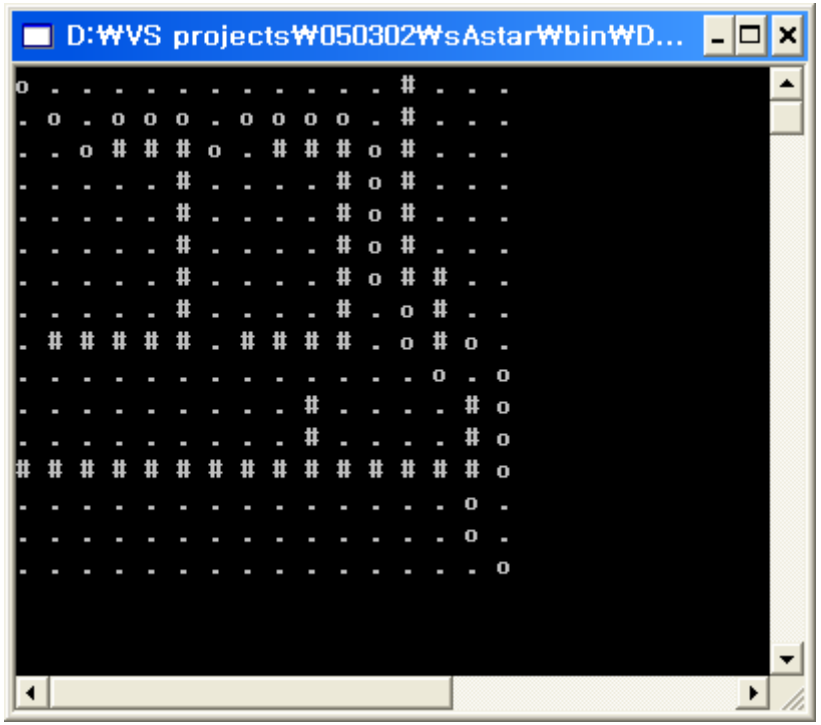
Very simple A* algorithm implementation

seunghyop back, 17 Mar 2005

Simple A* pathfinding algorithm implementation for beginners

Download demo project - 9.01 Kb



Introduction

This is about A* algorithm implementation which is about the way how we can find a best path between two positions. I already know that there are other A* implementations in this codeproject site. They are good, but I bet this is more simple and an easy implementation for beginners to understand.

There is no unnecessary code in this implementation, I just implement the A* algorithm pseudocode step by step in very intuitive ways. As you can see in the picture above, '#' means wall, each dot means available road, and 'o' means path that AI finds.

About A* algorithm

Shorty, A* is the most popular pathfinding algorithm to go from start to goal, based on efficiency of movement cost.

You can visit A* Pathfinding for Beginners(http://www.policyalmanac.org/games/aStarTutorial.htm) to learn how this algorithm works.

You can see pseudocode for A* that I used in this implementation at Justin Heyes-Jones A* tutorial. Visit here (http://www.geocities.com/jheyesjones/pseudocode.html) to see orginal pseudocode. The print-out of this pseudocode will help you a lot to understand this implementation.

Class Design

There are three primary classes in this implementation.

- **Map** - This class represents a map.
- **Node** - This class represents each tile on the map. It has two primary methods.
  - `CompareTo` - This method allows us to decide which node is better. We can say current node is better if this method returns negative number, which means current node has lower cost than the other node being compared.
  - `isMatch` - This allows us to decide whether two node's geomatical positions are same or not.
- **SortedCostNodeList** - This is a list that stores `Node` object list. We need to get off the lowest cost node from the list, so this list is implemented as sorted list order by cost value of the node, and we don't need to examine the costs of all elements in the list every time to pop the node which has the lowest cost because they are already sorted. Just pop one.This list has two primary methods.
  - `push` - add node elements to the list at proper position order by node cost.
  - Node's `CompareTo` method is used internally to sort order by cost.
    `pop` - just returns the lowest cost node from the list and remove it from the list.

Implementation

This is the core loop of the algorithm.

Hide Shrink Copy Code

```
ArrayList SolutionPathList = new ArrayList();

//Create a node containing the goal state node_goal
Node node_goal = new Node(null,null,1,15,15);

//Create a node containing the start state node_start
Node node_start = new Node(null,node_goal,1,0,0);


//Create OPEN and CLOSED list
SortedCostNodeList OPEN = new SortedCostNodeList ();
SortedCostNodeList CLOSED = new SortedCostNodeList ();


//Put node_start on the OPEN list
OPEN.push (node_start);

//while the OPEN list is not empty
while (OPEN.Count>0)
{
 //Get the node off the open list
 //with the lowest f and call it node_current
 Node node_current = OPEN.pop ();

 //if node_current is the same state as node_goal we
```

```
//have found the solution;
//break from the while loop;
if (node_current.isMatch (node_goal))
{
  node_goal.parentNode = node_current.parentNode ;
  break;
}

//Generate each state node_successor that can come after node_current
ArrayList successors = node_current.GetSuccessors ();

//for each node_successor or node_current
foreach (Node node_successor in successors)
{
  //Set the cost of node_successor to be the cost of node_current plus
  //the cost to get to node_successor from node_current
  //--> already set while we were getting successors

  //find node_successor on the OPEN list
  int oFound = OPEN.IndexOf (node_successor);

  //if node_successor is on the OPEN list but the existing one is as good
  //or better then discard this successor and continue

  if (oFound>0)
  {
    Node existing_node = OPEN.NodeAt (oFound);
    if (existing_node.CompareTo (node_current) <= 0)
      continue;
  }


  //find node_successor on the CLOSED list
  int cFound = CLOSED.IndexOf (node_successor);

  //if node_successor is on the CLOSED list
  //but the existing one is as good
  //or better then discard this successor and continue;
  if (cFound>0)
  {
    Node existing_node = CLOSED.NodeAt (cFound);
    if (existing_node.CompareTo (node_current) <= 0 )
      continue;
  }

  //Remove occurences of node_successor from OPEN and CLOSED
  if (oFound!=-1)
     OPEN.RemoveAt (oFound);
  if (cFound!=-1)
     CLOSED.RemoveAt (cFound);
```

```
   //Set the parent of node_successor to node_current;
   //--> already set while we were getting successors

   //Set h to be the estimated distance to node_goal
   //(Using heuristic function)
   //--> already set while we were getting successors

   //Add node_successor to the OPEN list
   OPEN.push (node_successor);

  }
  //Add node_current to the CLOSED list
  CLOSED.push (node_current);
}
```

Once we get to the goal, follow parent nodes to find the solution path.

Hide Copy Code

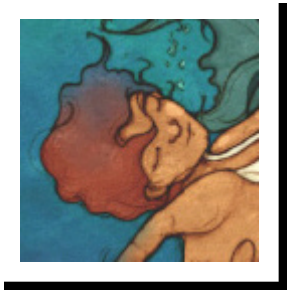
```
//follow the parentNode from goal to start node
//to find solution
Node p = node_goal;
while(p != null)
{
 SolutionPathList.Insert(0,p);
 p = p.parentNode ;
}
```

This A* implementation is very simple and good for beginners who want to know how A* algorithm works. Change map data in variety of ways, and check out how AI is smart to find the good path. Enjoy your programming.

License

Share

About the Author

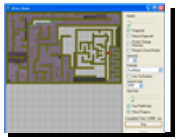


seunghyop back

Korea (Republic of)

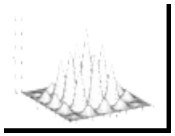
Working in SK Communications,South Korea as a web site developer.

You may also be interested in...



A* algorithm implementation in C#

Optimizing Traffic for Emergency Vehicles using IOT and Mobile Edge Computing



A Simple C# Genetic Algorithm

Build an Autonomous Mobile Robot with the Intel® RealSense™ Camera, ROS, and SAWR

Flexible New IoT Platform Empowers Enterprise Applications

SAPrefs - Netscape-like Preferences Dialog

Comments and Discussions

You must Sign In to use this message board.

Search Comments

Spacing Relaxed Layout Open All Per page 25 Update

First Prev Next

Diagonals

Member 11706703 30-Sep-15 0:53

Excellent work man, thanks!
But how I can modify the search path to ignore diagonals?

Edit: found.
Maybe not the best solution, but i just added to Node.GetSuccessors this:

Hide Copy Code

```
if (Math.Abs(xd)+Mathf.Abs(yd)==2) continue;
```

modified 30-Sep-15 6:01am.

Sign In · View Thread

Freeze

Member 10665704 23-Feb-15 8:32

The code seems to freeze sometimes when my character is trapped in a small room.

I've ported the code, but to be honest, I don't understand it much at all. I'd like the code to stop looking when there is no exit. Any hint on where I should make a modification?

Sign In · View Thread

Re: Freeze

Member 10665704 23-Feb-15 8:35

Also, it sometime uses one more move than it should.

Here is an example:
http://oi59.tinypic.com/okpgdi.jpg[^]

Sign In · View Thread

Thank you!

Member 8793129 30-Jul-14 19:07

Hey, seunghyop back!

Many thanks for your well-written article! It made it very easy for me to grasp the mechanics of the algorithm and provided me with a very good start on A*. Your article focuses on the essentials and your implementation reflects the conceptual structure of A* in a very crisp manner.

I have written an article in response to this implementation. Please take a look and let me know what you think. If you are unhappy with it, please let me know and I will either take it down or amend it.

A*: In Gratitude for Seunghyop Back's Introduction

Sign In · View Thread 5.00/5 (1 vote)

Good job

Member 10617067 22-Feb-14 6:27

Just wanted to say thank you. This script helped me a lot and was good programmed. Was easy for me to understand it and make some changes for me.

Sign In · View Thread

[Message Deleted]

Danny Rodriguez 27-Jan-08 10:19

[Message Deleted]

Sign In · View Thread

I realy needed this.Thanks.

toslavdsf3 29-Oct-07 15:22

Hi, I'm student at university, and I had to implement a A*. But all the projects I found were to big, and mixed with a lot of unnecessary code, like classes that implement some visual panels, mazes... And they didn't show the algorithm clearly.
This project was perfect for my needs. I just needed the algorithm A*. Now I can implemented in my way, and needs, because it was really easy to understand.
I will need to implement A* in NS2 for some mobile sensor network.Wish my luck.

seunghyop - Thanks again.

Tosho

Sign In · View Thread

HELP W/ ROGUELIKE

adiikan 9-Oct-07 8:43

I'm trying to implement this code into a turn based RogueLike console app very similar to the map in this example. Could someone give me some help in how to attach this to a robot - stepping through the path via a keystroke - instead of displaying the complete path from the start???

thanks in advance!
Monty

Sign In · View Thread

What about isometric maps?

worldspawn69 13-Jul-07 18:18

Does anyone know a good way to implement this code only using an isometric map? All the coords involved are floats instead of ints.

Sign In · View Thread

Just what I was looking for.

Henize 18-Sep-06 6:27

Iv been trying to impliment my own version of the A* algorithm. I have came close but have failed(3 times). It seems so simple but I just cant get it right Im going to use your design and create a .dll for use in one of my games..

---

static int Sqrt(int x) { if (x<0) throw new ArgumentOutOfRangeException(); int temp, y=0, b=0x8000, bshft=15, v=x; do { if (v>=(temp=(y<<1)+b<<bshft--)) { y+=b; v-=temp; } } while ((b>>=1)>0); return y;

Sign In · View Thread

MapLittle Mistake

Djooneagain 15-Sep-05 23:39

Your algorithm implementation is very detailed! It's good.

There's, I think, a little mistake in the function GetMap of the class Map (an inversion in the coordinates of the pixels).
Here, it's better:

public static int getMap(int x,int y)
{
int xMax =Mapdata.GetUpperBound (0);

```
int yMax =Mapdata.GetUpperBound (1);

if (x<0 || x>xMax || y<0 || y>yMax)
return -1;
else
return Mapdata[x,y];
}
```

Sign In · View Thread

An issue in the algorithm

LynxRaven 30-Mar-05 19:37

I have now seen this issue in two algorithms I have looked at... If someone is walking through a maze, why would they make an unnecessary diagonal step... I believe that a horizontal or vertical step should be worth 1, and a diagonal step should be worth 1.5 (as diagonal would we worth 1.414 more than a horizontal or vertical step).

If you were to give a cost to a diagonal movement of 1.5, there would be no unnecessary diagonal movements unless they were shorter than a vertical or horizontal movement...

Sign In · View Thread 5.00/5 (1 vote)

Re: An issue in the algorithm

Hyrum 2-Dec-05 8:25

Your issue with the algorithm is due to a false assumption about the world where the search takes place. A diagonal move may appear to be at a greater distance than a horizontal or vertical move, but the only thing we care about is the *cost* to make the move. It appears that in this search world the cost to make a diagonal move is the same as a horizontal or vertical move. The reason the result ends up with diagonal moves is the coder's choice to expand successor nodes in an order that happens to give priority to diagonal moves. Visually, it's a little confusing (and a little odd) but logically it works fine.

Truly, a much more interesting implementation of A* would have variable cost-to-go's all throughout the world. That way it wouldn't just be an issue of where are the walls or what's the shortest path but rather a question of which path has the least cost. And really, that's the main function of A*. Using A* in a world where all moves have the same cost is overkill.

Sign In · View Thread

Can't Download

Regina11_2000 18-Mar-05 10:44

I haven't been able to download the demo. Is there something wrong with the link to the demo?

Thanks for the posting though. I'm keen on trying it.

Cheers;

Daver

Sign In · View Thread

Re: Can't Download

Nishant S 20-Mar-05 20:10

Try again, please.

Nish

Sign In · View Thread

Last Visit: 31-Dec-99 19:00 Last Update: 21-Feb-18 19:24 Refresh 1

General News Suggestion Question Bug Answer Joke Praise Rant Admin