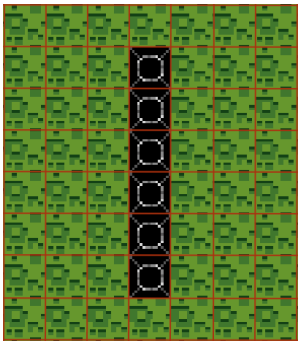
Jesse Ozog / Apr 2016
SmashRiot.com/stem/pf

AI: Pathfinding

In games, behavior of non-player entities is controlled by an artificial intelligence (AI), and an important component of any game's AI is Pathfinding. Pathfinding allows for the AI controlled entities to efficiently find the shortest path to for one place to another. One common Pathfinding algorithm is named A* (A-Star). This talk will introduce and discuss the A* Pathfinding algorithm. The students will utilize an example project to experiment with A* Pathfinding to direct and alter the A* Pathfinding's behavior to meet a particular goal.

Grid:



The level is first divided into a grid of squares in order to simplify the search area. The size of the grid should be small enough to show all the details, but large enough to remain efficient. Want to keep the total number of squares in the grid as low as possible. In this case, the grid is represented as a two dimensional array.

Nodes:

Each square in the grid is called a node. A node is a unit of the grid. Each node has attributes such as walkable, not walkable, costs, and parent node.

Lists:

The A* Pathfinding uses two lists to keep track of the nodes: Open and Closed

- **Open list:** a list of nodes that need to be looked at
- **Closed list:** a list of nodes that have already been looked at

Costs:

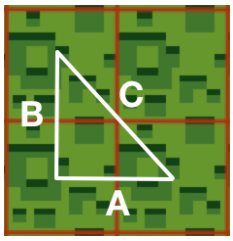
Adjacent node: node directly above, below, left or right of current node (N, S, E, W)
Diagonal node: node at a corner of the current node (NW, NE, SE, SW)

Determine a cost for moving to a neighboring node:

- Move to an adjacent node, the cost is 10
- Move to a diagonal node, the cost is 14

Why are diagonal costs = 14?
Pythagorean theorem to find distance of a diagonal: $A^2+B^2=C^2$

- If costs 10 to move West (A) and 10 to move North (B), then diagonal cost (C) is: (10*10) + (10*10) = 200 = C^2
- C = SqRt(200) = 14.14



Absolute Cost: Actual cost to get to this node (sum of all parent costs)
Estimated Cost: Estimated cost to the destination node (Manhattan distance):

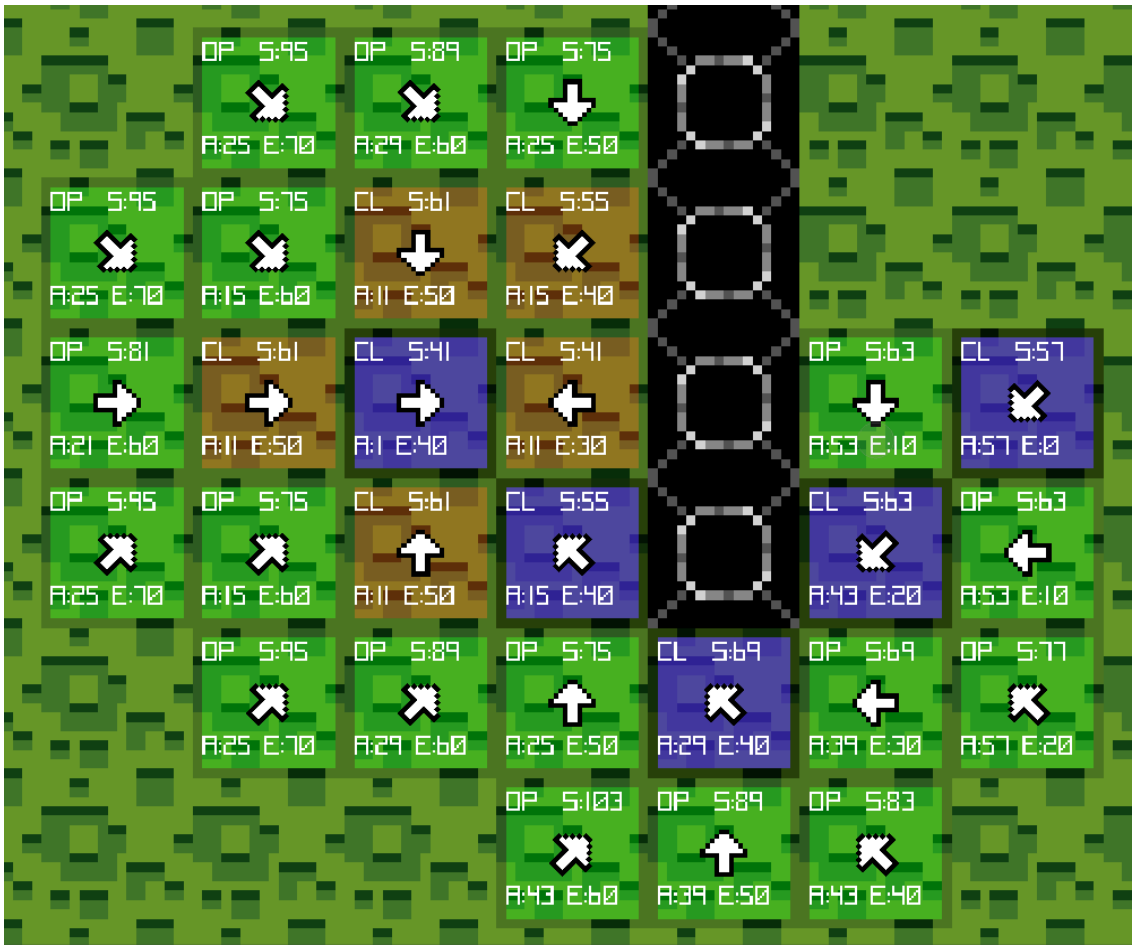
- Abs(Start.x - End.x) + Abs(Start.y - End.y)
- Example: Start=0,0 End=10,10: Abs(0-10) + Abs(0-10) = 10+10 = 20

Jesse Ozog / Apr 2016
SmashRiot.com/stem/pf

A* Algorithm:

Add the Starting node to the open list and calculate starting costs
Repeat 1-3 until Ending node is found OR run out of nodes to check

1. Move current node from Open list to Closed list
2. Check each of the 8 neighbor nodes around the current node
    a. Neighbor node is walkable
        i. Neighbor node is NOT on the closed list
            1. Neighbor is NOT on the open list
                a. Calculate ABS and EST cost for neighbor node
                b. Make current node the parent of the neighbor node
                c. Add to open list
            2. Neighbor is on the open list
                a. If neighbor's ABS cost is better than current node parent's ABS cost
                    i. Make neighbor node the new parent node for current node
                    ii. Update ABS costs for current node
3. Find lowest scoring node on open list and repeat 1-3 using new node



More information available at: http://smashriot.com/stem/pf/