



Carnegie Mellon University

CMU Defense

Team Name: Snow white and dwarves

Team member:

Bohan Hao

Hang Liu

Guoguo Lu

Guming Lun

Xuan Tang

Yihuang Zhang

You Zhou

Product Idea--Tower defense

- Goal:

Stop enemies from crossing a map by building towers which shoot at them as they pass.

- Rules:

- When an enemy is defeated, the player earns money.
- Enemies and towers usually have varied abilities, costs, and ability costs.
- Winning Condition: if all the CMU survives all monsters' attacks, the player wins.
- Lose Condition: if certain number of monster enter the CMU, CMU is devastated and the player lose.

Roles of members

- Missile Class: Yihua Zhang
- Tower Class: Xuan Tang
- Astar Class: GuoGuo Lu
- Map Class: Minglun Gu
- GameMenu Class: Hang Liu
- Monster Class: Bohan Hao
- GameControl Class: You Zhou, Guoguo Lu
- Integrating the whole program: All the members

Element: Monster--*Crazy Chicken*

- Four Types (With different speeds, health point(HP))



- Automatically find path to destination
 - Walk around blocks (Wall & Tower) by itself (by Astar algorithm)

Element: Tower--*CMU* construction

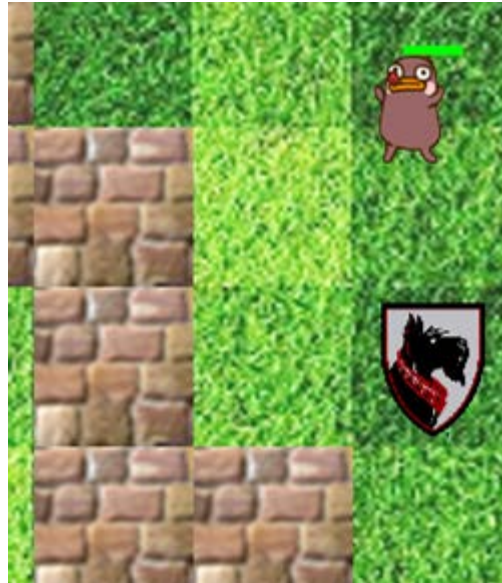
- Four Types (With different power, range, costs)



- Change the map information
 - The location of tower will be detected by monster

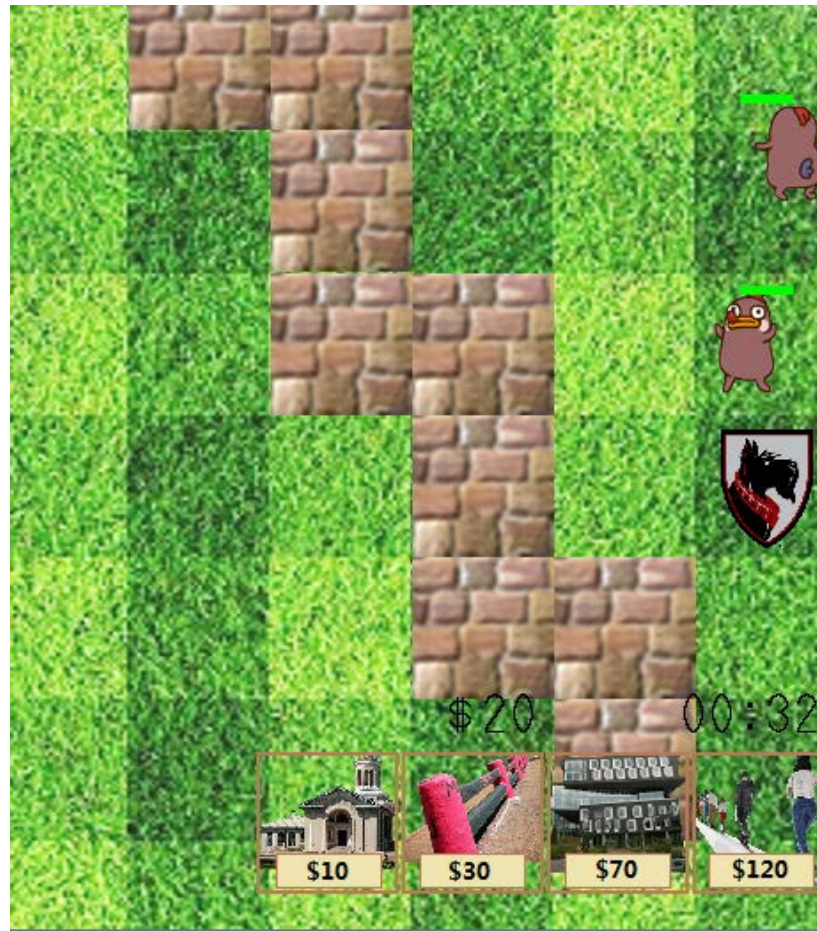
Element: Destination

- Tartan !!!



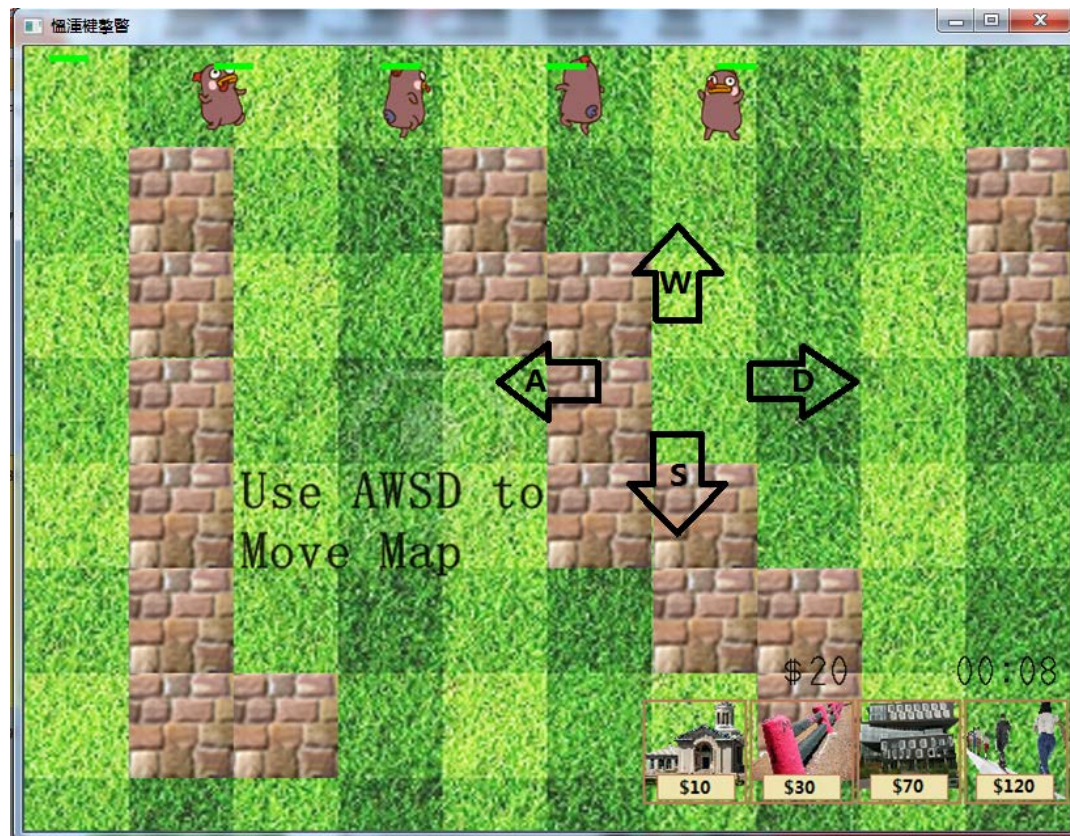
Game process

- Multiple wave monsters come to attack CMU



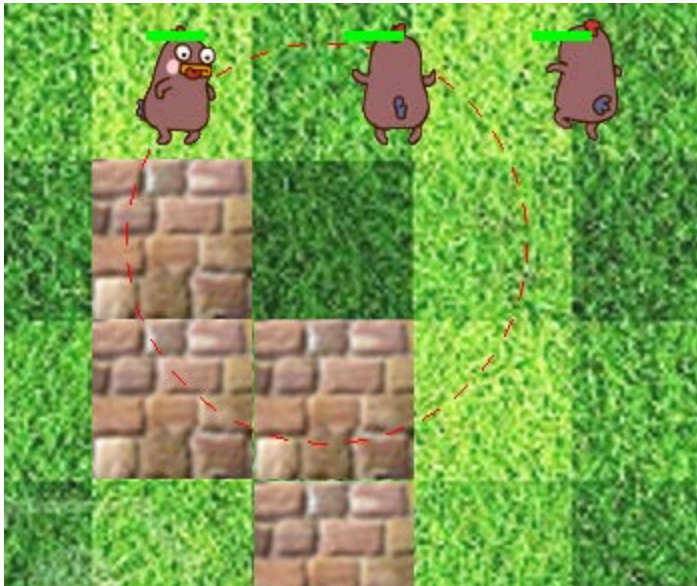
Game process

- Use “A”, “W”, “S”, “D” to move the map
 - Window: 800 * 600: only showed ¼ area of map



Game process

- Choose & build towers
 - Can not be built on walls
 - Block the path of monsters
 - Can not close all the paths
 - Hit monsters



Game process

- Total Money
 - Building tower will cost money
 - Earn money from eliminating monster



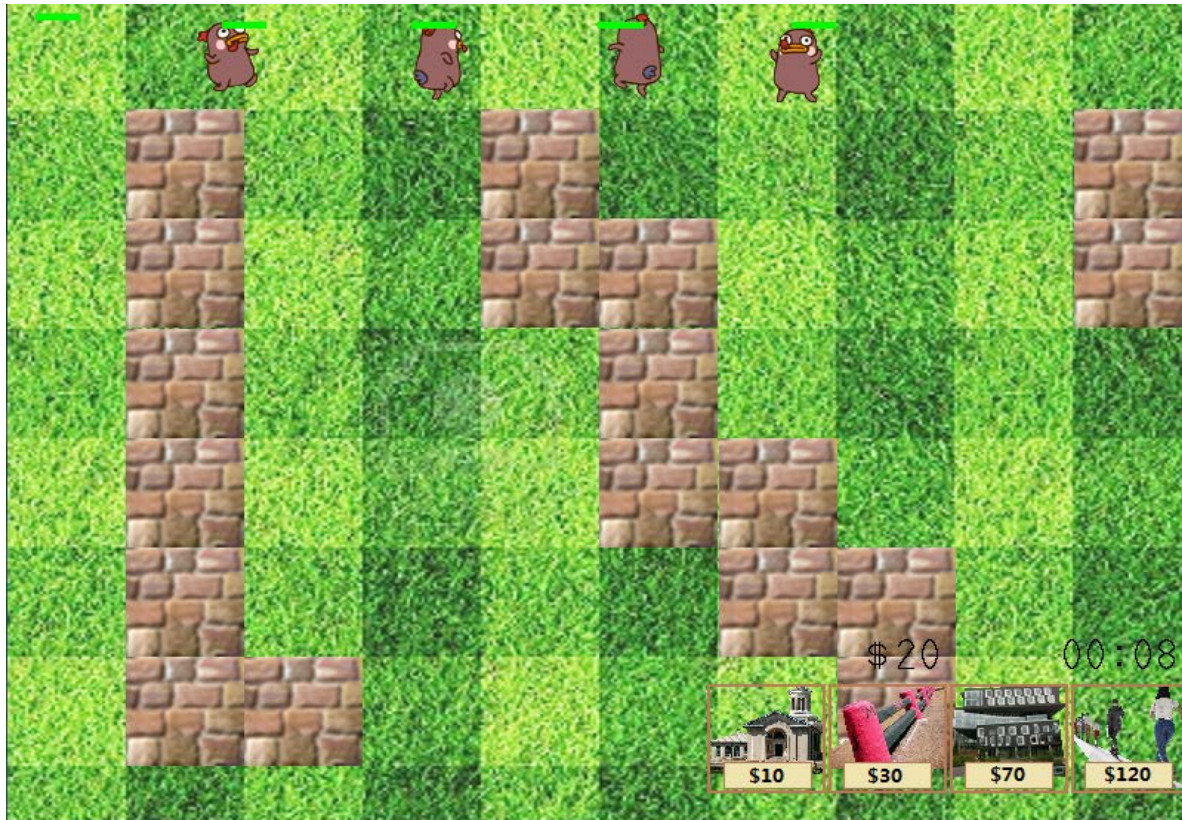
Game process

- Lose condition
 - More than twenty monsters reached Tartan
 - else
 - {Win;}

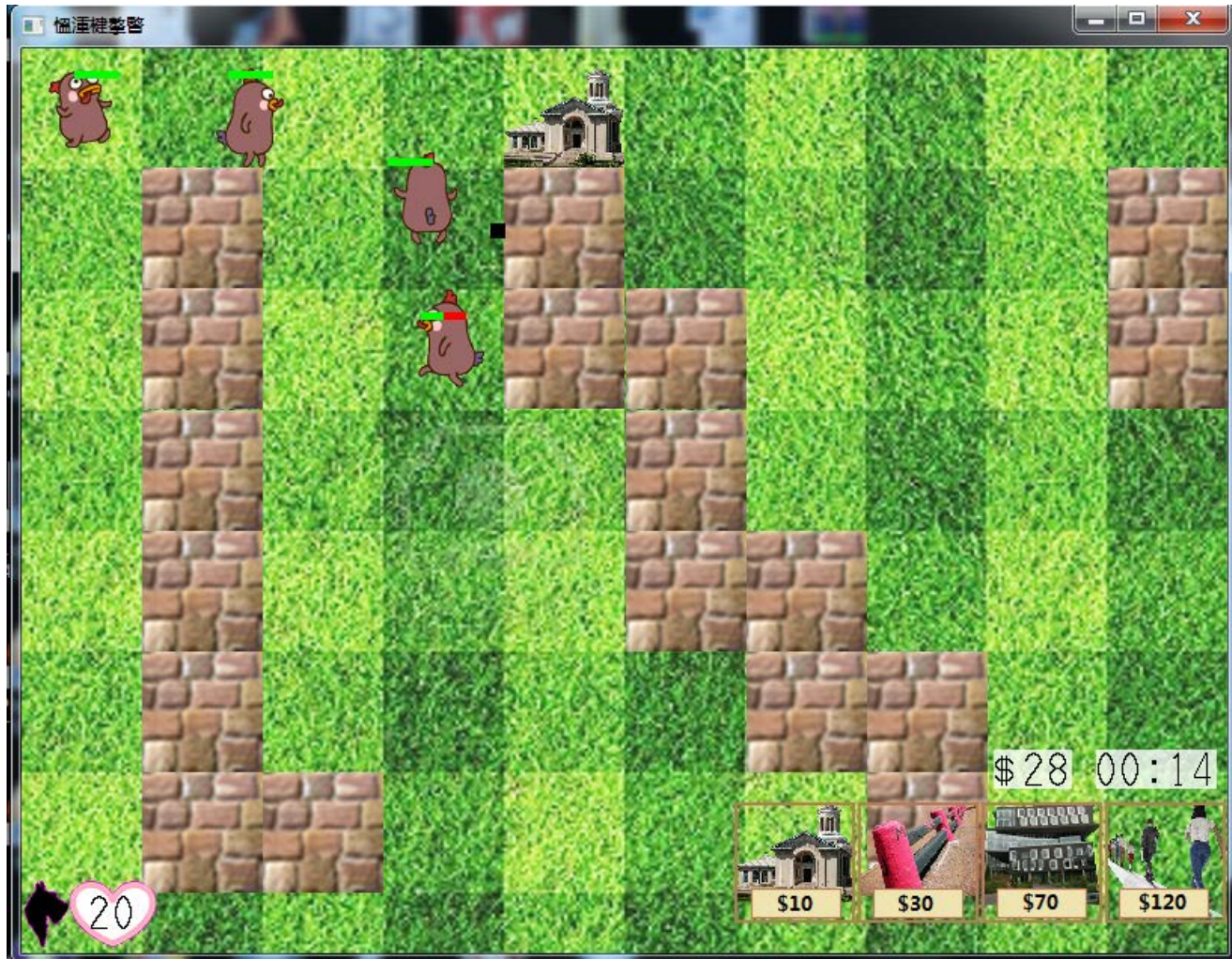


Product Idea—Special Features

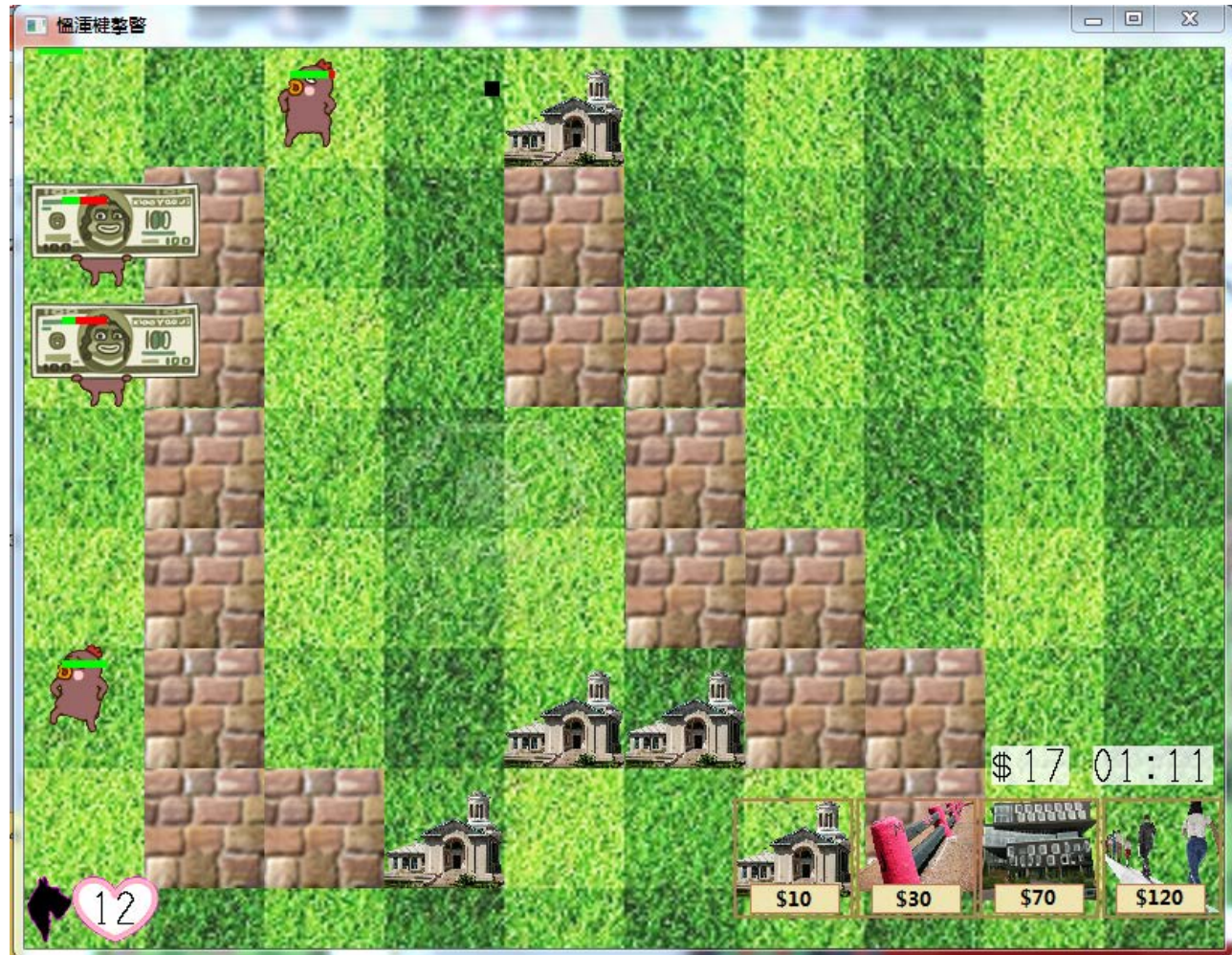
- Unlike other Tower defense Games, in our game the monsters can change the route automatically.



Product Idea—Special Features



Product Idea—Special Features



Challenges

- “Astar” algorithm
- Animation
- Team organization
 - Write code in a standard way is REALLY IMPORTANT!!!

Show game demo



Questions?

Appendix: A-Star-Algorithm

Path Scoring

The key to determining which squares to use when figuring out the path is the following equation:

$$F = G + H$$

where

G = the movement cost to move from the starting point A to a given square on the grid, following the path generated to get there.

H = the estimated movement cost to move from that given square on the grid to the final destination, point B.

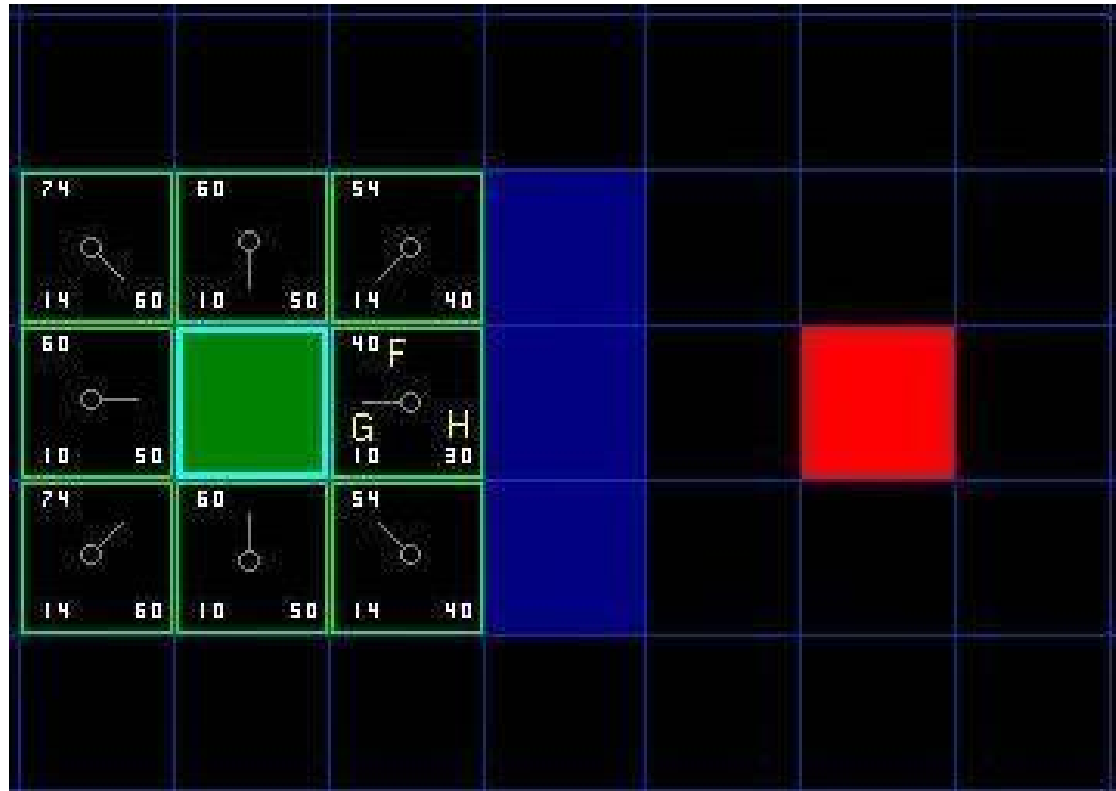
<http://www.policyalmanac.org/games/aStarTutorial.htm>

Appendix: A-Star-Algorithm

- Our path is generated by repeatedly going through our open list and choosing the square with the lowest F score.
- G: A cost of 10 to each horizontal or vertical square moved, and a cost of 14 for a diagonal move
- H can be estimated in a variety of ways. The method we use here is called the Manhattan method, where you calculate the total number of squares moved horizontally and vertically to reach the target square from the current square, ignoring diagonal movement, and ignoring any obstacles that may be in the way.



Appendix: A-Star-Algorithm



<http://www.policyalmanac.org/games/aStarTutorial.htm>

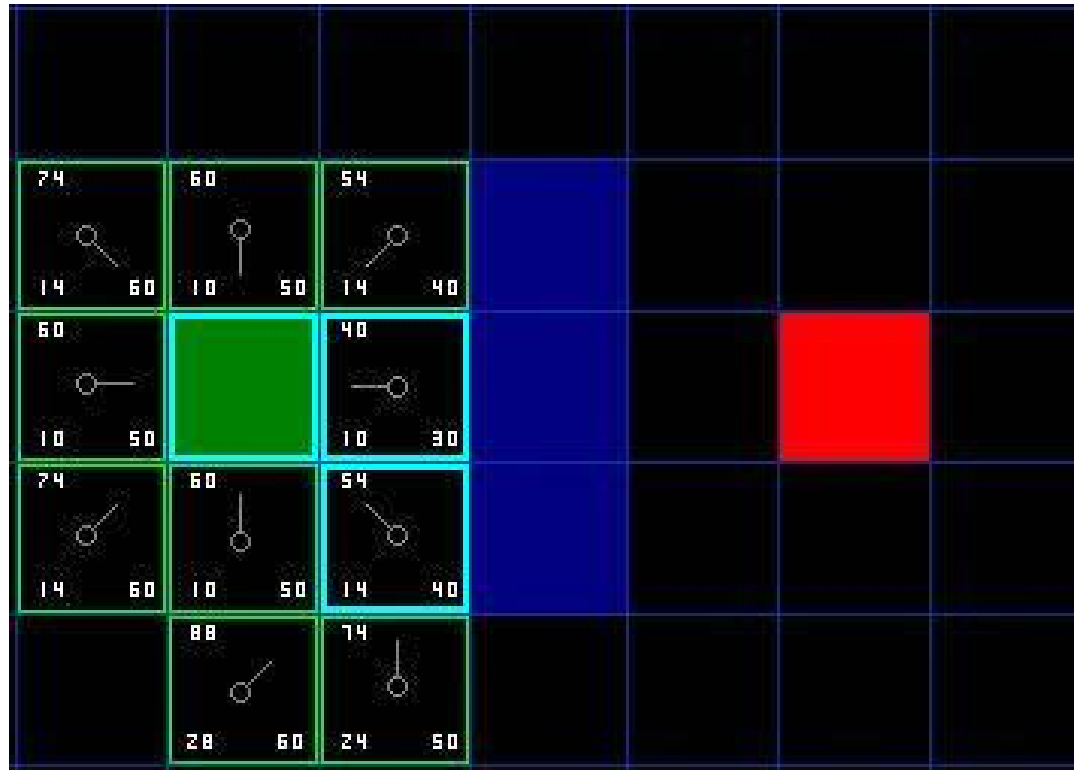
Appendix: A-Star-Algorithm

Choose the lowest F score square from all those that are on the open list.

- Drop it from the open list and add it to the closed list.
- Check all of the adjacent squares. Ignoring those that are on the closed list or unwalkable (terrain with walls, water, or other illegal terrain), add squares to the open list if they are not on the open list already. Make the selected square the “parent” of the new squares.
- If an adjacent square is already on the open list, check to see if this path to that square is a better one. In other words, check to see if the G score for that square is lower if we use the current square to get there. If not, don’t do anything.
- On the other hand, if the G cost of the new path is lower, change the parent of the adjacent square to the selected square (in the diagram above, change the direction of the pointer to point at the selected square). Finally, recalculate both the F and G scores of that square. If this seems confusing, you will see it illustrated below.

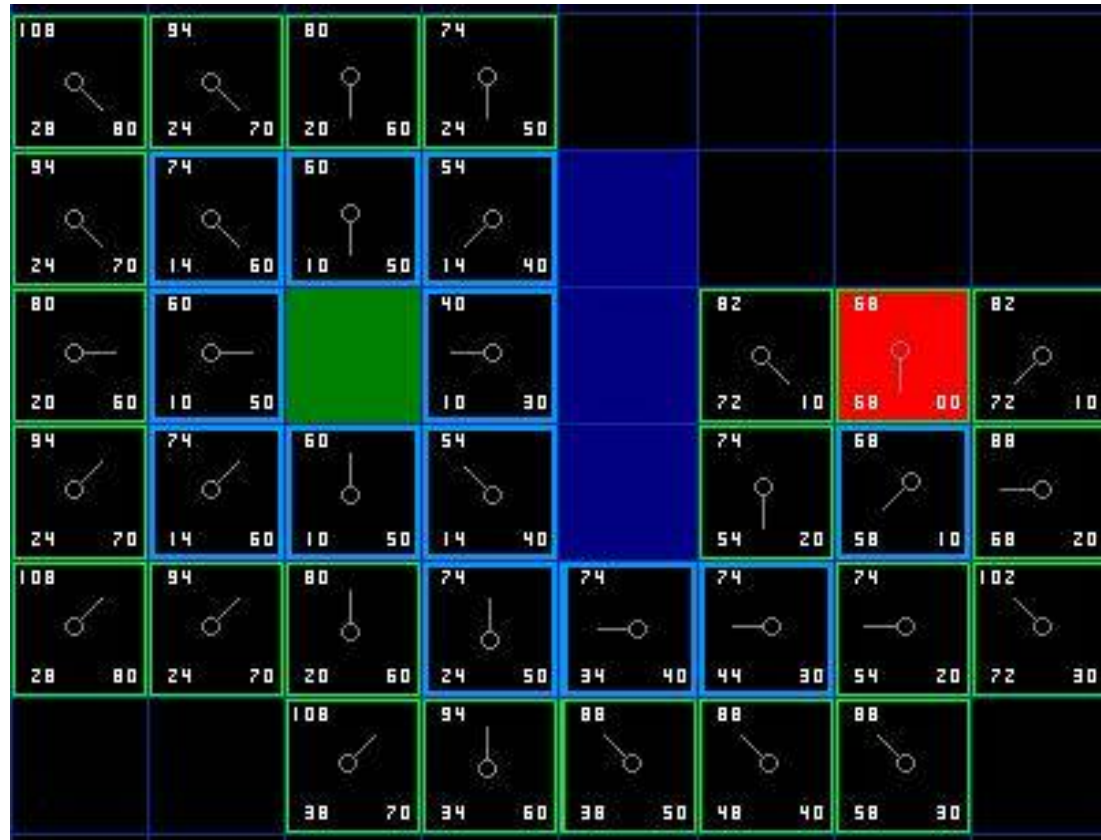
<http://www.policyalmanac.org/games/aStarTutorial.htm>

Appendix: A-Star-Algorithm



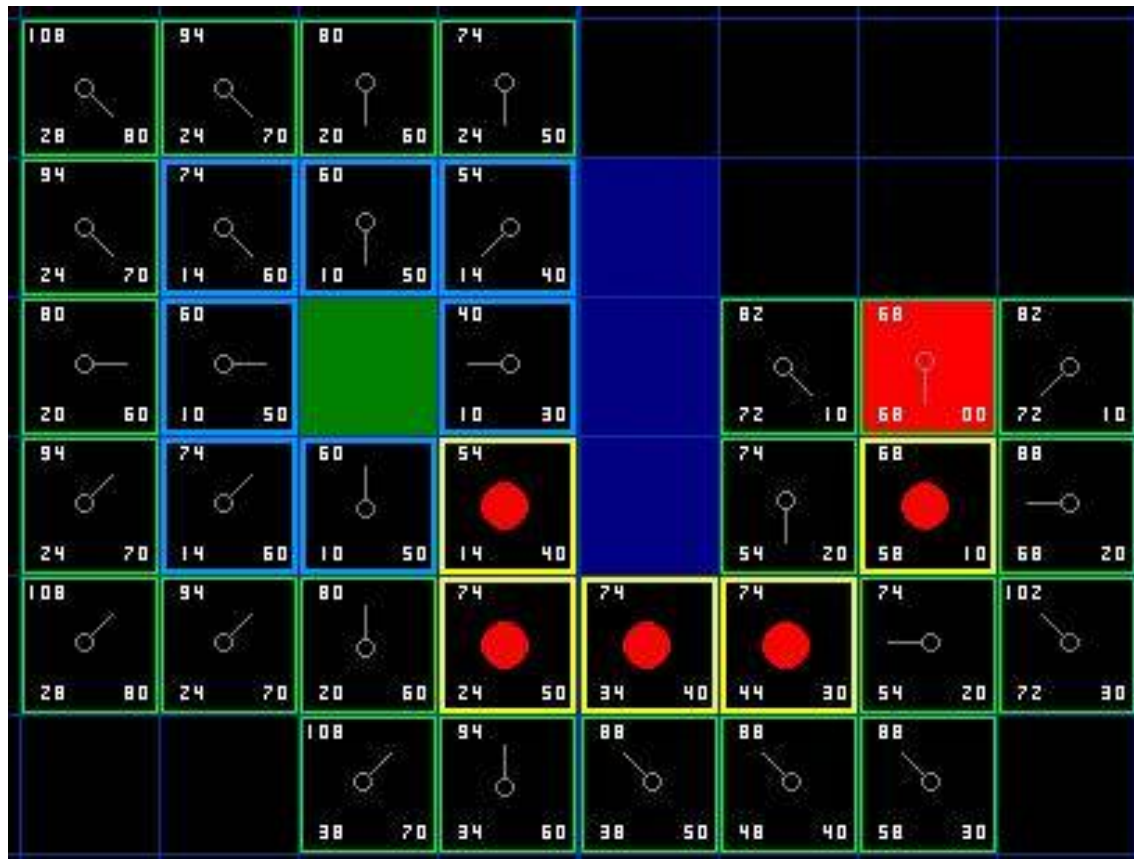
<http://www.policyalmanac.org/games/aStarTutorial.htm>

Appendix: A-Star-Algorithm



<http://www.policyalmanac.org/games/aStarTutorial.htm>

Appendix: A-Star-Algorithm



<http://www.policyalmanac.org/games/aStarTutorial.htm>