International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 11, November-2019

Improved Virtual Digital Trainer Kit using Mobile Computing

Virtual Digital Trainer Kit

Prof. Krishna Kulkarni
CSE
ICOER JSPM, Pune

Mr. Pravin Choudhari
CSE
ICOER JSPM, Pune

Mr. Navneet Kashyap
CSE
ICOER JSPM, Pune

Mr. Hrishikesh Warghade
CSE
ICOER JSPM, Pune

Mr. Atul Chougule
CSE
ICOER JSPM, Pune

***Abstract*— Digital training kit is used for the design of combinational and sequential logic circuits. There are a number of sources which provide virtual digital training kit but there are issues with the sources. Mobile has more computing power than computer which makes mobile a better platform for such applications. To increase the efficiency of the application we will be using improved A* algorithm and to make the GUI simple for the end user to understand we will be dividing the screen into square grids and to decide the next grid we will use heuristic value.**

Keywords— ***(1)Digital training kit (2)Virtual digital training kit (3) Improved A* algorithm (4) Square grid (5) Heuristic value (6)Virtual system (7) mobile computing***

I.INTRODUCTION

In the field of research digital training kit is very important for circuit making. It helps the researcher to build combinational and sequential circuits. The digital training kit is very important for the academics of the young engineers. Many colleges cannot provide the digital training kit 24/7 to the innovative minds of the young engineers to work on their ideas and also to practice at home. As the digital training kits are expensive students cannot afford to buy one and issuing it from college is not the best alternative as the kits can be damaged. This is where the idea of virtual digital training kit came. There are n number of software’s which were created but there are some major issues, these software’s are platform dependent, they have some specification requirements and need some previous knowledge to operate. To overcome the drawbacks of these software’s some researchers created digital training kit on websites. These websites which are providing virtual digital training kits are not platform dependent and does not require any previous knowledge for the implementation but there are issues with websites as well as the computation power of websites is less because of which many complex algorithms cannot be implemented which requires high computation power. This will eventually reduce the use of that particular website as many surveys have shown that user switches the website if it takes more time to response.

The current situation of computation field is that mobile phones have more computing power than the computers and there are more mobile users than the computers as the mobile applications are easy to handle as compared to the computer software. Here we will be providing a mobile application which will provide a virtual digital training application and this application will overcome the drawbacks of the previous technologies available in the market. The technologies used is Heuristic value, Improved A* algorithm and square grid.

II. IMPROVED A* ALGORITHM

A. The pseudocode of A algorithm is as follows:*

```
function find_path (Source, Destination) {
  OPEN_List = [Source];
  CLOSED_List = [];
  Current_Node = null;
  Neighbours = [];  Path = [];
  Source.Parent = null;
  while ( OPEN_List is not Empty ) {
        Current_Node = OPEN_List.remove_least_node();
        if (Current_Node == Destination)
        break;
        CLOSED_List.add (Current_Node);
        Neighbours = Map.search_neighbours( Current_Node );
        foreach (n in Neighbours) {
                examine(n);
                n.Parent = Current_Node;
                OPEN_List.add(n);
```

```
            }
        }
while( Current_Node is not null ) {
        Path.add( Current_Node );
        Current_Node = Current_Node.Parent;
        }
}
```

The A* algorithm have two lists i.e. OPEN List and CLOSED List to keep track of examined or traversed nodes. The OPEN list contains the nodes, which are discovered but not examined. On the other hand , the CLOSED list contains the nodes, which are discovered and also examined. The nodes are kept in sorted manner with respect to cost value of the node . While finding a path, the A* algorithm adds the Source Node to OPEN list, and then repeatedly removes the node having least cost value from the OPEN list and adds the walkable neighbor nodes to the OPEN list, until the destination node is finally removed from the OPEN list. As the nodes are sorted in the OPEN list, removing the node with least cost value become very much fast and optimal. The removed node from OPEN list is then examined. The A* algorithm gets terminated when the removed node is the destination node. Otherwise, the walkable neighbour nodes of the current node are discovered and then added to OPEN list after calculating cost value of each neighbour node by cost function of A* algorithm. And after that walkable neighbour node is added to OPEN list, and the current node is added to CLOSED list

In order to reduce the number of rotations taken by the A* algorithm, the cost function needs to be adjusted. As the heuristic cost function does not any obstacle in consideration between the current node and destination node, it may consider any rotation or may not. But in actual cost function, it does take care of the cost of moving from source to current nodes. And by adding a cost to actual cost value for each turn taken by that path, the more accurate result i.e. path with minimum turns can be achieved. The actual cost function g(n) of the A* algorithm g(n) = g(parent) + movement cost can be modified as follows:

$$g(n) = g(parent) + movement\ cost + R$$

Where, R is the cost of changing the direction i.e. the cost of rotating left or rotating right. The value of R can be defined as follows:

If turn is detected then R = CR
Otherwise R = 0

Where, CR is the cost of rotation. The Cost of Rotation depends on the size of Grid. The larger the grid size, the higher value of rotation cost required. Due to this, the cost value of the node will be increased after each turn. And hence, as algorithm chooses the nodes with least cost value from OPEN list, it will also select the nodes with a smaller number of turns, which may result into the path with a smaller number of turns.

To find the path from the Source to the Destination the improved version of A* algorithm is applied. The following results were obtained (Figs. 1, 2 and 3).

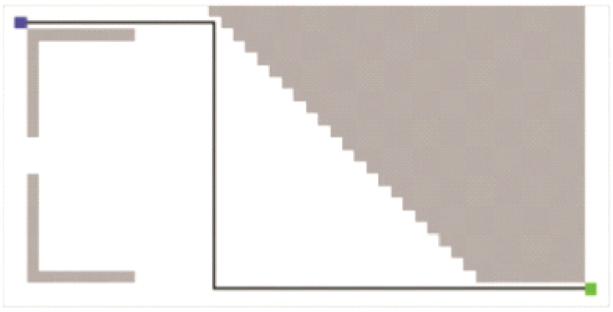


Fig 1. Grid 1

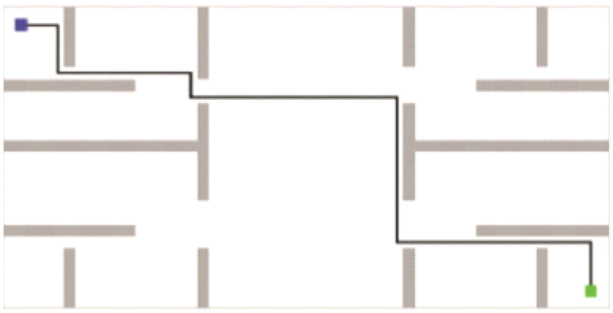


Fig 2. Grid 2

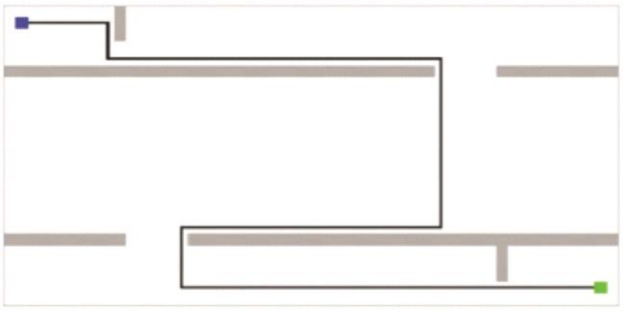


Fig 3. Grid 3

Table 1: Result of improved A* Algorithm.

Grid	Time required (in milliseconds)	Path length	Number of rotations
Grid 1	0.3082	70	2
Grid 2	0.4852	70	7
Grid 3	0.5446	112	6

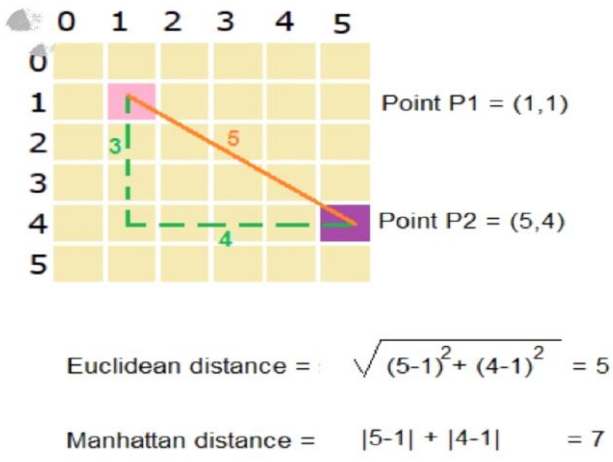
The traditional A* algorithm takes less time (in milliseconds) to find the path, but the number of turns in path are more as compared to Improve A* algorithm, whereas the improved A* algorithm takes slightly more time (few more milliseconds) but gives less number of turns. As the rotation cost increases gradually, the value of cost function of A* algorithm also increases, as a result, the accuracy of algorithm increases, i.e. number of turns taken reduces. In improved A* algorithm we discover few new nodes, which increases the time, hence the time to execute improved A* algorithm is slightly more than traditional A* algorithm.

III. HEURISTIC FUNCTION

Heuristics search refers to a search strategy that attempts to optimize a given problem by iteratively improving the solution based on a given heuristics functions or a cost measure.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 11, November-2019

Algorithms behavior based on heuristic and cost functions can be very useful in the game.
Speed as well as accuracy can be exploited to make your game faster. Construct an exact
Heuristic to precompute the length of the shortest path between every pair of prints


0 1 2 3 4 5
0 1 2 3 4 5
3
5
4
Point P1 = (1,1)
Point P2 = (5,4)
Euclidean distance = $\sqrt{(5-1)^2 + (4-1)^2}$ = 5
Manhattan distance = |5-1| + |4-1| = 7

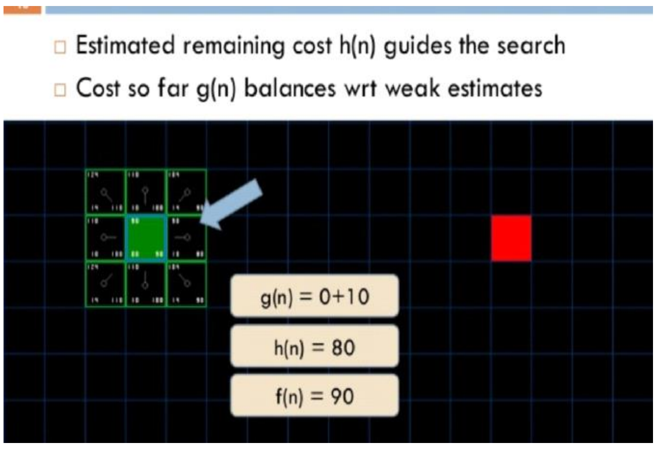
The function of heuristics is most important:
Function
Heuristics function is a function that estimates the cost of getting from one place to another place. (from the current state to the goal state) Also called as a heuristic. Used in a given decision process to try to make the best choice of list of possibilities (to choose the move more likely to lead to the goal state)

A. *This is not feasible for most game maps.so there are ways to approximate this heuristic.*

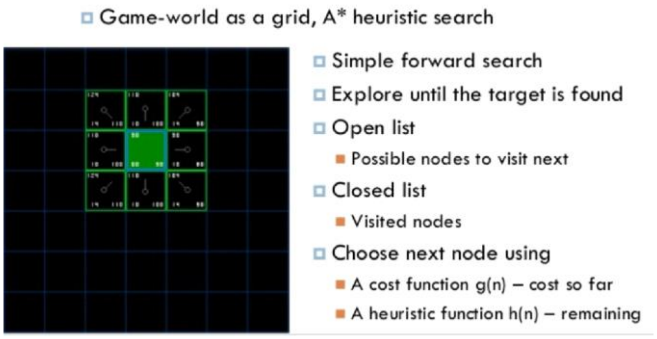
1) Fit a coarse grid on top of the fine grid. Precompute the shortage path between any pair of coarse grid location.
2) Precompute the shortest path between any pair of given waypoints. This concept is a generalization of the coarse grid approach.
3) If there is no obstacles and no slow terrain, then the shortest path from the starting point to the goal should be straight line. On a grid there are well known heuristic function to use.

B. *Important of heuristic search:*

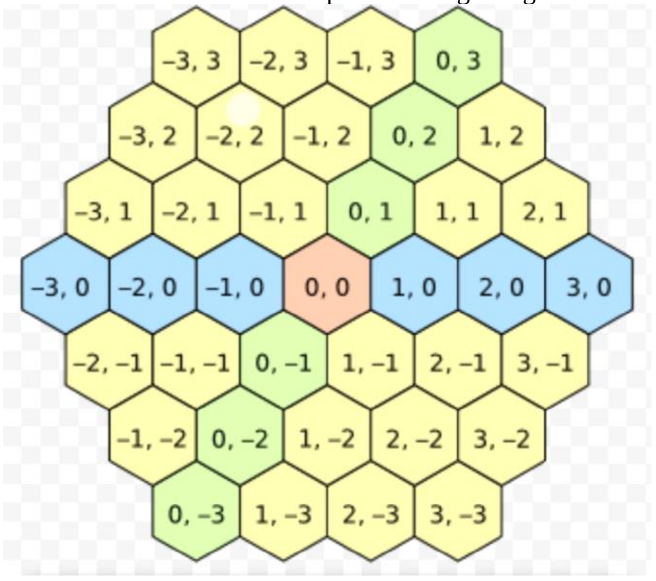
Heuristics reduce the mental effort required to make choices and decisions.
Fast and Frugal: Still other theories argue that heuristics are actually more accurate than they are biased. In other words, we use heuristics because they are fast and usually correct.


- Estimated remaining cost h(n) guides the search
- Cost so far g(n) balances wrt weak estimates

C *Heuristic function to use in :*
On a square grid that allows 4 directions of movement,Manhattan distance.
The Manhattan distance is an admissible heuristic in this case because every tile Will have to be moved at least the number of spots in between itself and its correct position. The subscripts shoe the Manhattan distance for each tile.


- Game-world as a grid, A* heuristic search
- Simple forward search
- Explore until the target is found
- Open list
  - Possible nodes to visit next
- Closed list
  - Visited nodes
- Choose next node using
  - A cost function g(n) – cost so far
  - A heuristic function h(n) – remaining

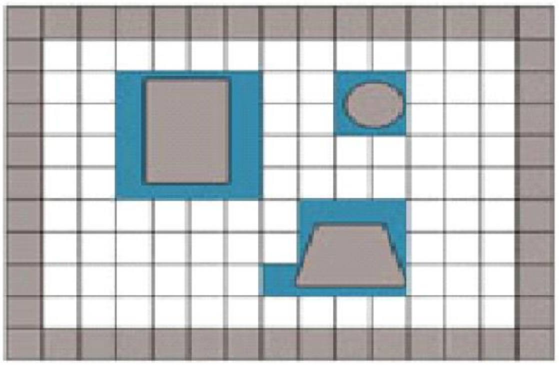
1) On a square grid allows 8 directions of movement, diagonal distance.
2) On a square grid that allows any direction of movement, might or might not want Euclidean distance
3) On a hexagon grid that allows 6 direction of movement, uses a Manhattan distance adapted to hexagonal grids.



IV. GRID

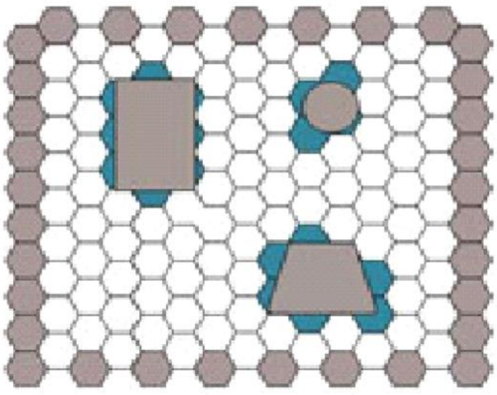
Terrain topology is the foundation of the mobile application and generating the graph is the most important part of it. Here we are studying different graph generating techniques such as 2D square grid (octile), 2D hexagonal grid and 2D triangular grid. A grid contains vertices and edges which connects the vertices. *Study done by Zeyad Abd Algfoor, Mohd Shahrizal Sunar and Hoshang Kolivand gives us the idea about the advantages and disadvantages of 2D square grid, 2D hexagonal grid and 2D triangular grid.*

A. 2D square grid



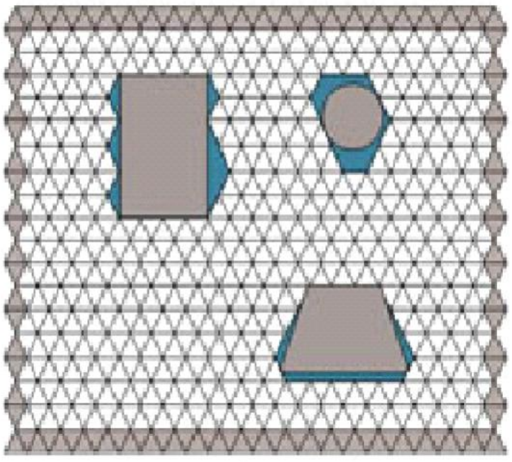
1) Square grid is very popular in computer games and number of algorithms have been proposed for path finding problem. Harabor and Grastien have given the jump point search (JPS) single-agent algorithm. This algorithm was given to solve the well-known problem named "uniform-cost octile grid in static environment." Uras et al published a method to accelerate the path search by generating subgoal graphs. Bnaya et al proposed the Exhaustive and Monte- Carlo Iterative Taxing Algorithms (EITA/MC-ITA) for the multi agent path finding problem. The drawback of these techniques is that it assigns priorities randomly to one or the other agent. Anderson presented an excellent additive heuristic for a single-agent search on a four-connected square grid with dynamic obstacle.Koenig and Sun demonstrated the advantages and disadvantages of real-time and incremental heuristic searches in square grids.

B. 2D hexagonal grid



Bj¨ornsson et al have shown that 2D hexagonal grid have many of the properties of the square grid. Hexagonal grids have smaller search time and memory complexities than grid graphs constructed from squares. The only problem with 2D hexagonal grid is that it takes more time to execute for the large map. Path finding algorithms have mainly been implemented on square grids, and the multi agent problem on a hexagonal grid has not yet been investigated in dynamic or real-time environments.

C. 2D triangular grid



Triangular grids are not widely used as compared to square grid and hexagonal grid but still triangular grid have some desirable properties. There TA* and TRA* algorithm found out to be perfectly working on the large maps.

The study of all three-graph making strategy gets up to a conclusion that for mobile application 2D square grid would be the most suitable.

V. REFERENCES

[1] Zeyad Abd Algfoor, Mohd Shahrizal Sunar and Hoshang Kolivand, "A Comprehensive Study on Pathfinding Techniques for Robotics and VideoGames" *Media and Games Innovation Centre of Excellence (MaGIC-X) UTM-IRDA Digital Media Centre, University Teknologi Malaysia, 81310 Skudai, Johor, Malaysia.*

[2] Geethu Elizebeth Mathew1 gemsmat8@gmail.comII- ME-CSE G.Malathy2 malathi.gurunathan@gmail.com, "DIRECTION BASED HEURISTIC FOR PATHFINDING IN VIDEO GAMES" K.S.R Institute for Engineering and Technology-Tiruchengode, IEEE SPONSORED SECOND INTERNATIONAL CONFERENCE ON ELECTRONICS AND COMMUNICATION SYSTEMS(ICECS '2015

[3] Ashok M. Chaudhari1(✉), Minal R. Apsangi2, and Akshay B. Kudale1, "Improved A-star Algorithm with Least Turn for Robotic Rescue Operations" 1 Information Technology, PVG's COET Pune, Pune, Maharashtra, India ashok78670@gmail.com, akshayak616@gmail.com 2 Faculty, Information Technology, PVG's COET Pune, Pune, Maharashtra, India mra_it@pvgcoet.ac.in

[4] D. P. Gadekar, N. P. Sable, A. H. Raut, "Exploring Data Security Scheme into Cloud Using Encryption Algorithms" International Journal of Recent Technology and Engineering (IJRTE), Published By:Blue Eyes Intelligence Engineering & Sciences Publication, ISSN: 2277-3878, Volume-8 Issue-2, July2019, DOI: 10.35940/ijrte.B2504.078219, SCOPUS Journal.

[5] Sable Nilesh Popat*, Y. P. Singh," Efficient Research on the Relationship Standard Mining Calculations in Data Mining" in *Journal of Advances in Science and Technology | Science & Technology,* Vol. 14, Issue No. 2, September-2017, ISSN 2230-9659.

[6] Sable Nilesh Popat*, Y. P. Singh," Analysis and Study on the Classifier Based Data Mining Methods" in *Journal of Advances in Science and Technology | Science & Technology,* Vol. 14, Issue No. 2, September-2017, ISSN 2230-9659.

[7] Jinli Xu, Fei Hu, Xiaolei Han, "**Realization of Bidirectional A* Algorithm Based on the Hierarchical Thinking During the Process of Path Planning**" Wuhan Univ. of Technol 2008 International Conference on Computer Science and Software Engineering Year: 2008 | Volume: 1 | Conference Paper | Publisher: IEEE

[8] Ji-Xian Xiao, Fang-Ling Lu, "**An improvement of the shortest path algorithm based on Dijkstra algorithm**" 2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE), Year: 2010 | Volume: 2 | Conference Paper | Publisher: IEEE