1. Abstract

This project is based on the classic tank game – Battle City. For enemy tanks, each tank is an artificial intelligent agent and A Star Algorithm is used for path searching. The target is mutable for enemy tanks, as it could be either the castle or the player. Each enemy tank follows the path generated to its target every time.

2. Introduction and Background

Upon creation, the target for enemy tanks is the castle, and each enemy agent searches the shortest path to destroy the target, Fig.2.1. If the player tank approaches when the enemy tank is moving along its path, the enemy tank will generate a new path from its current position and escape from the player tank. Once the player eats a bonus and the fortress of the castle turns into steel which is bullet proof, the target of the enemy tanks becomes the player. As the player moves, the enemy tanks update their path to the player's position every game-refresh time.

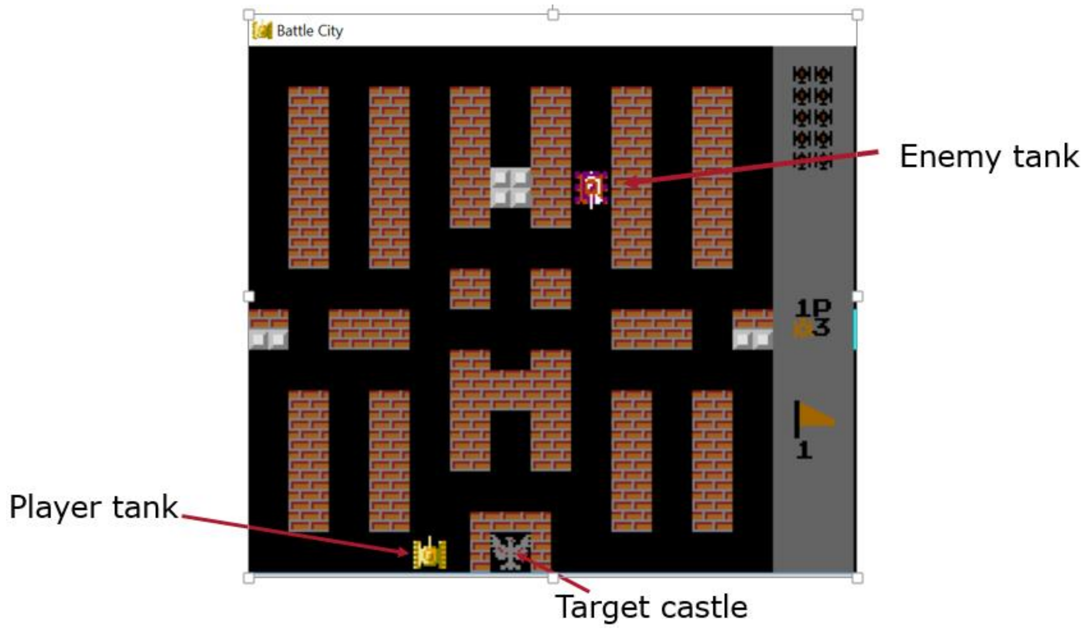
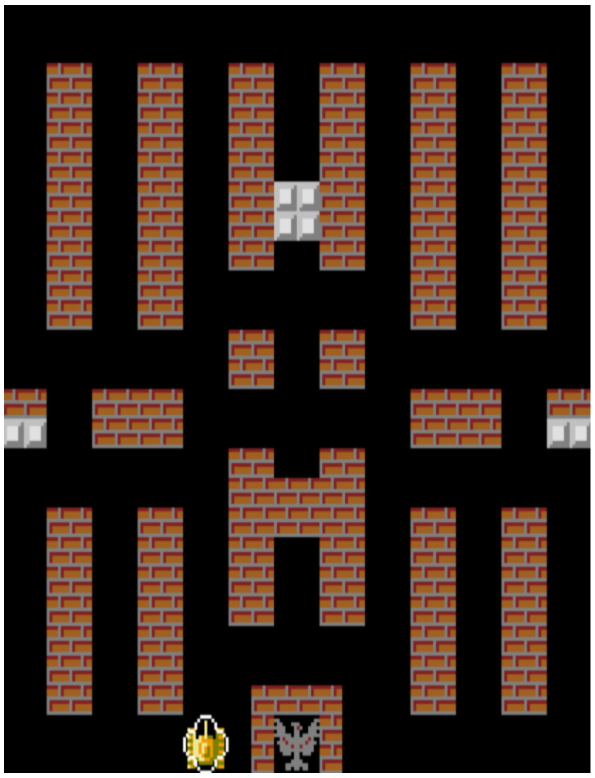


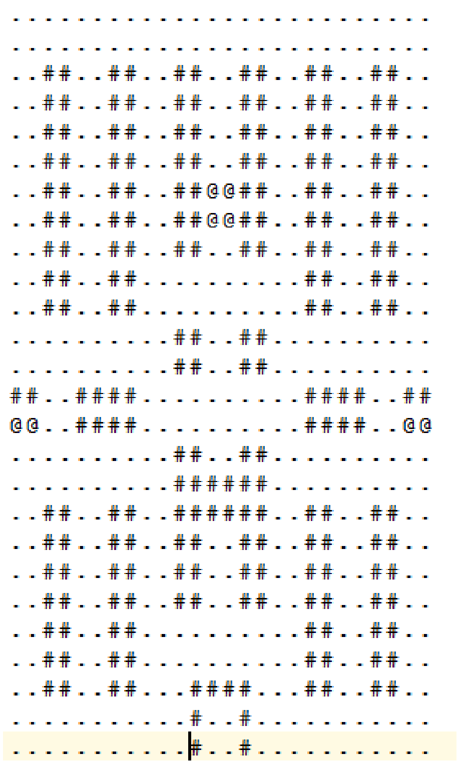
Fig. 2.1 Battle City Game

A star algorithm is used for path searching of enemy agent. A* is an informed search algorithm, or a best-first search, meaning that it solves problems by searching among all possible paths to the solution (goal) for the one that incurs the smallest cost (least distance travelled, shortest time, etc.), and among these paths it first considers the ones that appear to lead most quickly to the solution. In short, A* is a combination of Uniform-cost search which orders by backward cost g(n) and Greedy search which orders by forward cost h(n), and A* orders by the sum: f(n) = g(n) + h(n).

3. Implementation

The map for the game is symbolled in Fig.3.1:





```
..........................
..........................
..##..##..##..##..##..##..
..##..##..##..##..##..##..
..##..##..##..##..##..##..
..##..##..##..##..##..##..
..##..##..##@@##..##..##..
..##..##..##@@##..##..##..
..##..##..##..##..##..##..
..##..##..........##..##..
..##..##..........##..##..
..........##..##..........
..........##..##..........
##..####..........####..##
@@..####..........####..@@
..........##..##..........
..........######..........
..##..##..######..##..##..
..##..##..##..##..##..##..
..##..##..##..##..##..##..
..##..##..##..##..##..##..
..##..##..........##..##..
..##..##..........##..##..
..##..##...####...##..##..
...........#..#...........
...........#..#...........
```

Fig.3.1 A* Map

Bricks are represented by ‘#’, and the path is represented by ‘.’, steels are represented by ‘@’. An A star map is created to represent the value of each position on map, and the policy is as follows:

The value for ‘.’ is ‘1’, the value for ‘#’ is ‘9’, the value for ‘@’ is ‘999’, and the value where the player resides is ‘9999’. The A star map is updated every time a change occurs to the map (i.e. a brick disappears)

The cost at each position is the value of each position on the A star map. The heuristic value is the Manhattan distance from each position to the target position. Our f value is the sum of the both values. We built a priority queue and every time the position with smallest f value is popped out to be expanded. Therefore, at the first place when the fortress is not bullet-proof, the enemy agent searches for the shortest path to its target and will avoid those paths that are close to the player tank because the cost is huge at those positions.
As soon as the player eats the specific bonus and the fortress of castle becomes bullet-proof, the target of enemy tanks becomes the player tank, and the enemy tanks will search for the shortest path to the player tank to attack. Note that the position of the target changes in this situation, and the path is updated every time the target position changes.

4. Proposed Experiments and Outcomes

The experiment is set as the game between A* search algorithm applied enemy tanks with a human controlled player tank. To simplify the action control of the enemy tanks, all enemy tanks will keep a continuous shooting when they are moving. All actions of the player tanks are human controlled by buttons on the keyboard. In order to reduce the complex of play policy and evaluate our algorithm for path searching efficiently, the game was also simplified in some other aspects, such as only bonus “shovel” available.

The goal of enemy tanks is to find out the best path to its target: the castle or the player tank. The path searching is achievable by A* search. Moreover, any enemy tank can automatically change its targets according to different environmental states.

Case 1: Target is the castle
When the castle consists of only bricks, which can be destroyed by normal bullets, the target of enemy tanks would be the castle. Under this situation, the position of the target is fixed. The best path to the castle from any enemy tank will be computed and shown on the right side of our display window, Fig 4.1. The path may change when the enemy tank is approaching the play tank. Since the target of the enemy tanks is destroying the castle and win the game. The enemy tanks will try to find an alternative way to avoid of nearby passes of the player tank, which may cause a head-on collision with the player tank. All these changes on proceeding path of the enemy tanks will be shown synchronously on the right side display window.

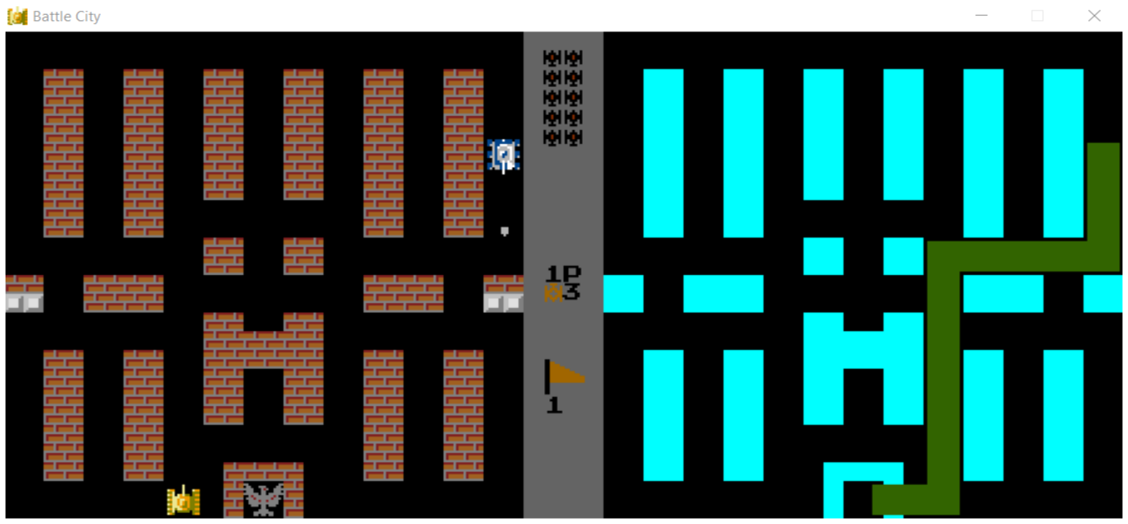


Fig. 4.1 Case 1: Target is the castle

Case 2: Target is the player tank
After the player tank “eating” bonus “shovel”, the castle will be enforced with steel wall, which cannot be punctured by normal bullets. Under this situation, the enemy tank will detect this change and find out it becomes impossible to win the game by destroying the castle. Instead, now the only way the enemy tank can win the game is to destroy the play tank. As a result, the enemy tank will change its target automatically to the play tank. The new path will be generated by A* search algorithm similarly and shown in the right side of display window, Fig 4.2.

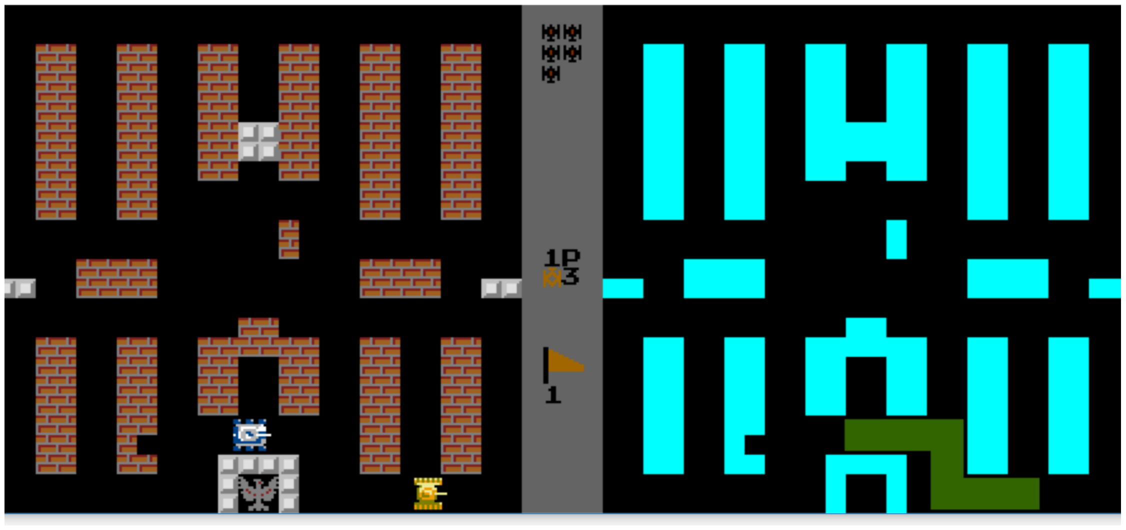


Fig 4.2 Case 2: Target is the player tank

5. Conclusions

By applying A* search algorithm on the enemy tanks in the battle city game, the best path can be found for the enemy tanks with automatically changed targets. Even the game was simplified to assume a continuous shooting from the enemy tanks, the outcomes from our experiments successfully show an automatic target changing and synchronous path updating playing policy of the enemy tanks, which fulfilled our purpose of the enemy tanks' auto-playing.