Final Project

Abstract

By trying the game that called “Tron Light Cycles”， we found the offensive strategy has a higher advantage,so we adopt the attack mode to design ‘my agent’, occupying a large moving area, force the enemy to take the lead in the "suicide", in which the algorithm is improved according to the snake game.

In summary, we can divide into the processes of attack, pre-judgment, area segmentation and backtracking.Known game map for 20 * 20 grid, on the lower left corner in order to establish the right Angle coordinate system origin of coordinates, each set as A unit, with our enemy is located in the diagonal grid two horns, respectively to the diagonal grid partition, the game USES the A * algorithm to get to the shortest distance and direction of the enemy's head, combined with the security range, factors such as the area of each step to determine our direction, making the final segmentation to our larger area.When the distance between the two

parties in Manhattan is 4 or the region is divided, the backtracking problem is considered, and the inverse A* algorithm is used to determine the longest feasible path in the region, so as to avoid "hanging" before the enemy after the region is divided. In the project, under the conditions given by the title, we set up two other conditions of the opponent: solo drive-agent and wall-agent. In the three conditions, we built our initialization model, found the shortcomings through the analysis of the results, and optimized the model.In the case of barrier free and rock free, we run two models and get the following results:

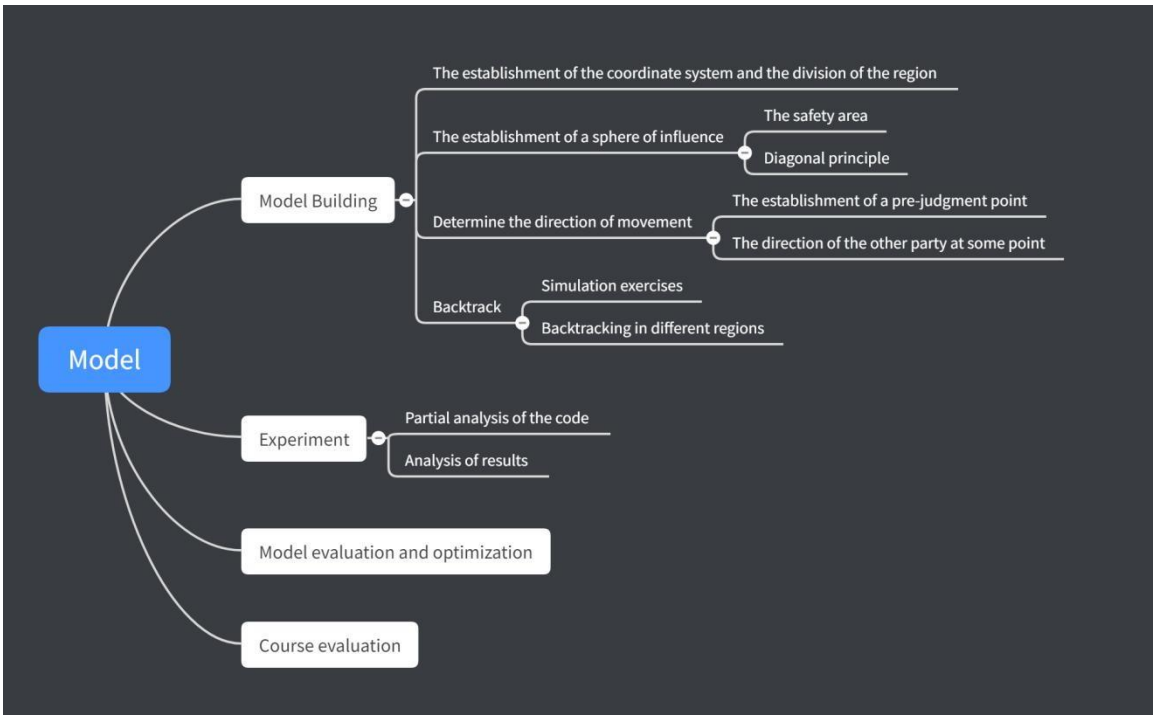
condition:
round:1000
board size:20*20

		non-improved	improved
solo drive-agent	rocky	95.20%	99.70%
	default	96.60%	100.00%
wall -agent	rocky	91.20%	97.80%
	default	94.80%	100.00%
random agent	rocky	98.30%	99.00%
	default	99.40%	100.00%

table1

Key word：A*algorithm、inverse A*algorithm、Manhattan distance

Through the analysis of the problem, the modeling mind map is as follows:



Model building and solving

1.1Model building

Main idea: take the vertex in the lower left corner of the region as the origin to establish a coordinate system with a square as a cell. Meanwhile, divide the region into three parts according to the diagonal: area I, area II and area III. Secondly, we will set the influencing range from our starting point. As we move, the influencing range will expand, and the obstacles in front of us will be detected to ensure our safe movement. According to the current coordinates of the other party's head at each moment and the coordinates of the other party's head at

the previous moment, the direction of the other party's movement at the previous moment is determined. At the same time, according to the region of the other party in the region, the A* algorithm is used to determine the direction of our party's movement at this moment. The final region segmentation is realized by means of attack and containment. For rapid regional segmentation, according to the opponent head's current coordinates and the moment before its coordinates to determine the direction of the opponent in the previous time. Besides, according to the area (I, II or III) it occupies ,we use the A * algorithm to determine our direction at this time, and adopt the method of attack and containment to realize the regional segmentation. Then we will backtrack and tell which part of the segmentation area is larger, later we use the inverse A* algorithm to calculate the longest path to the initial area, and win the game for the purpose of the opponent dying before us.

1.2 Attack

1.2.1 establishment of coordinate system and division of areas

To accurately determine the position of the opponent at every moment and judge the direction of its movement, we establish the coordinate system with the lower left corner of the region as the origin,

the horizontal axis as the row, the vertical axis as the column, and a single squire as a cell. In addition, to better judge the direction of our movement at each moment, we divide the area into three parts I, II and III as figure 1.

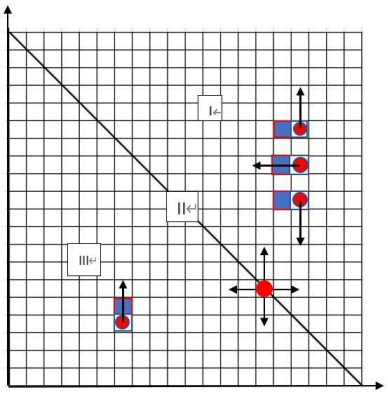


Figure 1

1.2.2 Establishment of influencing range

a. Possible state

Assuming we have gone three steps, the numbers in figure 2 are the possible state for each step, and the area occupied by numbers is our available area.

b. Diagonal principle

Since initial positions of mine and the enemy are on the diagonal of the area, moving along the diagonal as much as possible ensures that we attack and defend as quickly as possible, which has excellent foresight.

c. The determination of the influencing scope

In order to ensure that we can attack and encircle at the fastest speed , we combine every step available area with the diagonal principle to determine our influence range (as shadow area shown in figure 2).

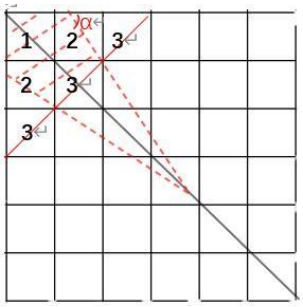


Figure2

1.2.3 Judge moving direction

In order to judge our moving direction at each moment, two factors are considered: the direction of the opponent's movement at this moment, and the opponent's next possible state which is closest to our influencing range.

A certain moment in the game was marked as , the opponent's current position was marked as (r_n,c_n), the previous moment was marked as -1, and the opponent's position was marked as (rn-1,cn-1). Therefore, the direction vector of the opponent's movement at that time was determined to be $(r_n-r_{n-1}, c_n-c_{n-1})$.

In addition, according to the coordinates of the opponent's position at that moment, the part (I, II or III) opponent located is determined. Combined with FIG. 1 and FIG. 2, according to different regions, we can

determine which state is nearest to our influencing range, after opponent finishing the next step, and it's our current goal. It can be divided into the following three categories:

A. When the head of the opponent is in zone I :(r_n, c_n-1)

B. When the head of the opponent is in zone:(r_n, c_n)

C. When the head of the opponent is in zone :(r_n-1, c_n)

Combining these two factors with A* algorithm, we could find the shortest path to the goal and ensure the direction of our next step.

1.3 Preliminary judgment

Initial location area: the backtracking area is a squire whose length of side is 0.3 times of the board's, according to the experience.

In order to avoid my agent "suicides", when the distance between my agent's head and the opponent's head is not greater than Manhattan distance 4, and we can still find a complete secure path towards the opponent's head, we start to predict the possible situation if we finished the next step, and test whether my agent can safely feedback to the initial position area after that step, if it could, my agent can take this step, otherwise, using DFS algorithm to backtrack now. Besides, if we cannot find a complete and safe path to the opponent's head position, we will directly use DFS algorithm to realize the backtracking process.

1.4 Area division

When my agent's head is close to the opponent's, we make preliminary judgment, meanwhile, to judge whether the next state can reach the two vertices of the auxiliary diagonal safely and without obstacles. If it can, continue to move; otherwise, it indicates that the division of the region has been completed.

1.5 Feedback process

After we finished the division of the region, we will choose the larger area to backtrack, and use the inverse A* algorithm to find the longest path to return to the initial region.

Model evaluation and optimization

3.1Advantages of the model

A. This model my agent has a high winning rate. This model USES A * algorithm, DFS, etc. to make the agent become very intelligent.

B. This model is forward-looking. By analyzing the positions of both sides and the routes they have taken, it can predict the future situation and adjust the direction to improve our winning rate.

3.2Deficiencies and optimizations of the model

3.2.1

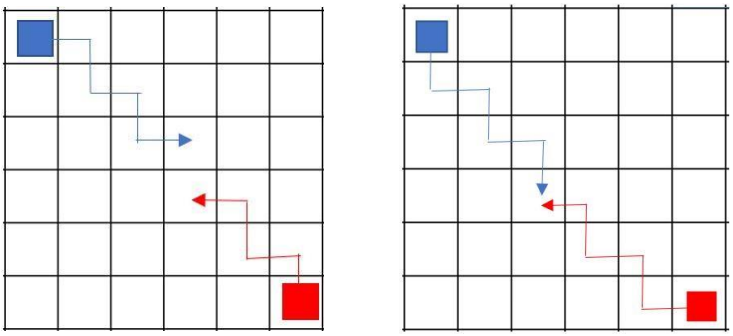
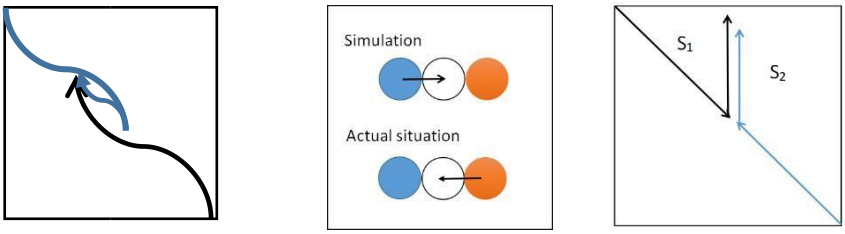


Figure 3

As shown in the figure, when both sides adopt the same strategy as above, the Nash equilibrium will be reached, and the first one will fail.

3.2.2



a b c

Figure 4

(1) Since we have been aiming at the head of the enemy or near the head of the enemy, when the two sides are very close to each other, we will not choose to turn to the right, but follow the enemy forward to make ourselves the first to "hang".

Improvement: in order to prevent the above situation and save time, will be put forward in the 3 * 3 grid security area(1.3) according to the

diagonal into triangle and triangle, when must carry on the back, to determine the two sides in the area I or II, when located in the area I, the triangle as the safety area, when located in zone II, above the triangle as the safety area, to determine the safety area, after using A * and inverse A * algorithm to carry on the back, determine the longest possible path, when there is no A safe path can lead to upper triangular or lower triangular, using the random function randomly selected the next step.

(2) As shown in Figure 4 (b), there is only one space between the two sides. When our longest safe path needs to go through this space, we do not know the next step of the enemy.

Improvement: when there is only one space between the two sides, store all possible positions of the enemy in the close list, and then make prediction to judge whether our subsequent path will be blocked. If so, give up the step and backtrack according to the improvement scheme in (1).

(3)As shown in figure 4 (c), the two sides adopt the same strategy and move along the diagonal. When the two sides are very close, the two sides will follow each other, and the black side is one grid ahead of the blue side. The black side will be the first to "hang itself".

Improvement: in case of (c), the next step, S2, goes first and goes up. At this time, S1 goes down to avoid the two sides sticking forward.

Curriculum evaluation

Although in the special period, the form of online course replaces the traditional teaching mode, we still have some understanding and learning of this course in a short period of time.

In the course, we learned BFS, DFS, A*, CNN, HMM and other algorithms and knowledge, and learned to use python language to deal with simple problems.In terms of homework and examination, considering the professional direction of most of us, the teacher had reasonable requirements on us. Meanwhile, we also learned teamwork in the homework and examination.

Three weeks passed in a flash. Here, all the members of our group would like to thank the teacher.