Intelligent Systems I - Group Assignment

1st João Santos (76912)
DETI
Universidade de Aveiro
Aveiro, Portugal
santos.martins.joao@ua.pt

***Abstract*—In the proposed assignment it was asked for the students to build a intelligent system capable of taking thoughtful actions that, hopefully, lead an army to defend a base as long as possible. The game is composed by an allied army which the goal is to defend its base while an attacking army does what it can to destroy it. The developed agent was capable of defending the base considerably longer than the default strategy.**

***Index Terms*—army, game, agent, intelligent systems**

I. Introduction

An intelligent agent is anything that observes its surroundings, acts autonomously to achieve goals, and can learn or use knowledge to enhance its performance. In the scope of this assignment, the aim was to developed such an agent that could manage resources and defend a military base by coordinating the allied soldiers against the enemy.

The game happens in turns. In each, first the agent (or allied) moves are applied and then the enemy has its turn. If the agent takes longer than two hundred milliseconds to apply its moves, then they are ignored. Also, it can not take more than five hundred actions. If the agent can survive for two thousand turns, then it wins the game.

In this game, their are four kinds of entities:

1) Base: this base is where the allied soldiers are recruited and the resources are obtained. It can be upgraded so, at each turn, more resources are gained;
2) Allied Melee: these soldiers can only attack the enemy when they are on the same cell. Their elimination ratio is 1:1, this is, on melee can only attack and eliminate one enemy. Each one costs fifty resources;
3) Allied Ranged: these soldiers can attack an enemy when it is within a three cell (Manhattan) distance. Their elimination ratio is 4:1, this is, a single ranged soldier can eliminate one enemy at three, two or one cells of distance plus another one when in the same cell. Each one costs one hundred fifty resources;
4) Enemy Melee: these are the soldiers that will try to attack the base or any of the allied soldiers. They move towards the closest goal.

Also, two game difficulties exist for the agents to be tested. On the easier difficulty 0, the enemies are spawn in a random position, always along the last column of the board. The base cost for upgrading the base is five hundred while the base production is two hundred.

On the harder difficulty 1, the enemies spawn on a backslash formation. The base upgrading cost eight hundred and the base production is still two hundred.

These base values are what define how much it costs to upgrade the base throughout the game and how much it will produce at each level.

II. General Architecture

In a macro point of view, the developed agent differentiates its behaviour (from now on called macro strategy) accordingly to the known difficulty of the game. Mainly, the agent loads a set of parameters and chooses the accordingly individual strategy (called micro strategy) for each cell with any allied soldier (melee, ranged or base). Yet, both macro strategies have big similarities.

III. Main Algorithms and Parameters

Both macro strategies consist on building a defence block composed of ranged soldiers, while the melee soldiers try to attack the enemy by reaching the last column of the board. While this happens, the base upgrades its level in steps, this is, given a set of conditions that will be detailed in the coming paragraphs, it stops spawning any soldier so all the generated resources are invested in the upgrading process. This conduct switches back to the default buy strategy when a desired level is reached.

A. Difficulty 0

Before going any further, we need to define the concept of battle front. In the scope of this project, the battle front if the farthest column that a allied ranged soldier will occupy of the battle formation. It is in this column where these soldiers will wait and attack the coming enemy.

In the same way, the formation rows are the rows that these same ranged soldiers can occupy. Both concepts are visible on Fig. 1. For difficult 0, the battle front was at column twenty while the formation rows where all except the middle one.

On the easier difficulty, the enemies spawn in random cells, i.e. they have no formation. This allows the melee soldiers to find ”holes” in the attack of the enemy so that it can reach the last column of the board. To do so, the melee soldiers are always recruited on the center row (the row where the base is located) and move forward until the battle front is reached.

When the battle front is reached, three actions can happen:

1) if there is no enemy in the next two cells then move forward;
2) if going forward is not a feasible solution, check if going up or down will put the melee soldiers in danger of dying in the next turn. If not, then choose one at random and move there;
3) when all other options failed, simply move backwards in the hope than a better escape opportunity appears. If it does not, and the soldiers can not go backwards any further, sacrifice themselves for a greater good.

Notice that all this happens for both melee soldiers that are and are not hidden. The two cell check for moving forward is needed because if less, then in a turn where both the allied and enemy move forward, they would end up in the same cell and, as a result, in an unwanted clash.

In parallel to this, the ranged soldiers are working in building the defence block. The defense block consists in moving towards the battle front and occupying all the formation rows with the maximum amount of soldiers able to attack (fifty ranged soldiers). When the battle front is fully occupied, the immediately previous column is filled in the same way, and so on until the game ends or no more columns are free. When the enemy can reach and eliminate some or all the allied soldiers, then the formation is rebuilt by the soldiers on the back column moving forward.

In this difficulty, the formation rows are all except the middle one. This is intend so that the melee soldier have where to pass towards their goal.

Given that a ranged soldier can only eliminate enemy soldiers if it is stationary, before each movement a check is performed. If an enemy is present at a distance equal or shorter than three cells (Manhattan distance) then the soldiers will not move. This happens whatever the current goal and position are. In code, this is a simple breadth-first search tree that starts with the closer cells and expands one unit distance at each time until if finds an enemy or the distance is greater than three.

Finally, the missing piece is the behaviour of the base. In the beginning, the base is upgraded whenever possible (the amount of resources is greater than the upgrade cost). When level seven is reached, the base stops the first phase of upgrading and switches to a buy strategy in order to start building some defences. The strategy consist in buying all the ranged soldiers possible with the current resources, but considering that we will want to buy, at most, twenty melee soldiers. This option was taken so that a good equilibrium between attacking and defending is gained.

Notice that, in this difficulty, it is desired to have the center row free for the melee soldiers to pass and attack the enemy. This is the reason why the melee soldiers are recruited of the cell in front of the base while the ranged are split in half and recruited in both the top and bottom cells adjacent to the base. Fig. 1 shows the general idea of the formations.

A second upgrading step for the base when the ranged soldier fully occupy the first two columns of the battle front, taking the base up to the level fourteen. The third and final step, the one that takes the base up to the final level of eighteen, happens when the first four rows are fully occupied.

As an emergency action, if during any of the upgrading steps, the battle front soldiers happen to be attacked en eliminated, the base immediately starts recruiting more soldiers (using the same policy) until that column formation is rebuilt.

B. Difficulty 1

On the harder difficulty, the macro strategy is very similar, while the micro strategies differ. The first difference is that the battle front is closer to the base to give time for the ranged soldiers to gather in their positions (see Fig. 2).

Also, since now the enemies appear in a diagonal formation, there are no holes that the melee soldiers can take advantage of. As a counterpart, it takes a known (and longer) number of turns for an enemy soldier to spawn in the same row. This makes it more beneficial for the melee soldiers to attack through the top and bottom rows. The formation behaviour of the enemies makes it unfeasible to try to avoid them. So, to save resources, the melee soldiers are only recruited while they (in the hidden amount) can successfully reach the last column. When the enemy spawns are so large that they can not, they stop. This is controlled by only recruiting while the base does not reach the second step of the upgrading process.

One of the biggest threats happens when enemy soldiers from different cells come together and add up to amounts that the defence block can not stop. To try to keep them spread, when the recruitment of the melee soldiers ends, the ranged soldiers also start to occupy the rows that were previously free for their passage.

Given the increased costs on this difficulty, the author decided to maintain three steps on the upgrade strategy. Still, they happen as follows: in the start, the base is only upgraded up to level five. Then, when the first two columns on the battle front are completed with squads composed of fifty ranged soldiers, the second step starts and goes to level ten. This is also when the agent stops recruiting melee soldiers. The final step takes the base to level fourteen and occurs when the first three columns of the battle front are filled. After this point, at each turn the maximum possible amount of ranged soldiers are recruited and the agent keeps trying to hold the army positions until the enemies spawns are so large that they can punch holes on the army.

The most important parameters of this implementation have already been mentioned: battle front (column), formation rows and when the base upgrade happens and to what level. Given the define strategy, namely for difficulty 1, level fourteen is the most feasible that could be set. When trying to upgrade to any further, it would not be reached because it gives time for the enemy to start creating holes on the defense. Contrarily, on difficulty 0, going further than level eighteen provided no measurable benefit since it would take too long and too many resources to give very little reward.

IV. Implementation Alternatives

The initial idea to solve this assignment was to use a neural network that would learn the best strategy for the game.

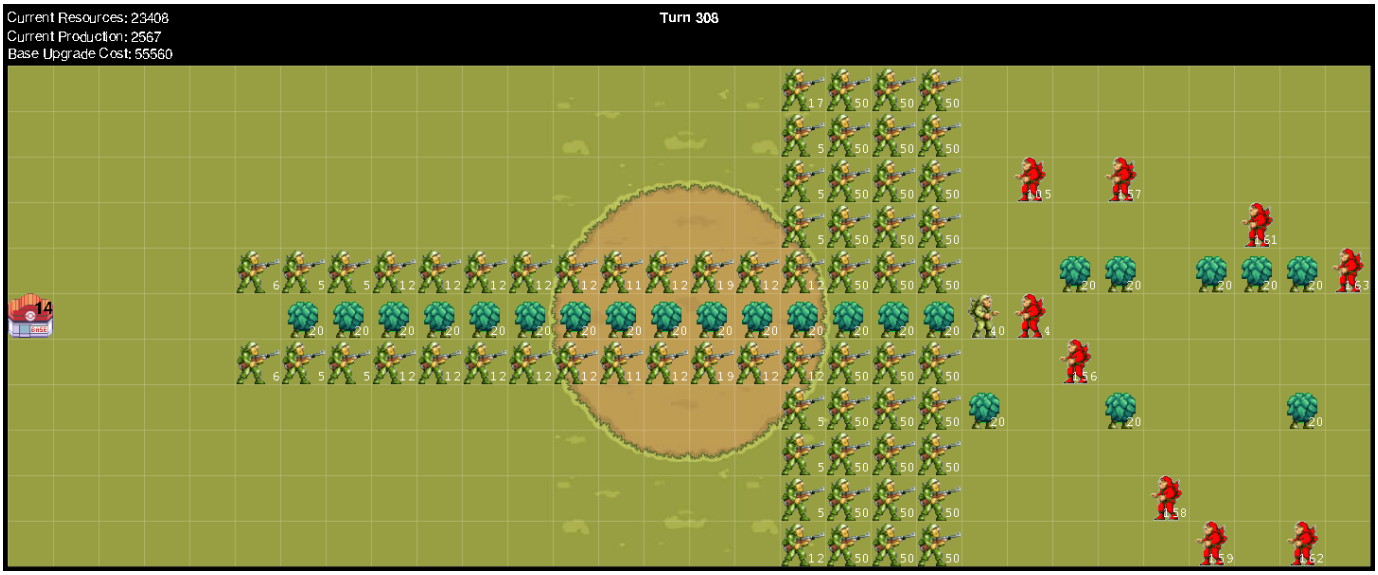

Fig. 1. Example frame of the adopted strategy for difficulty 0.

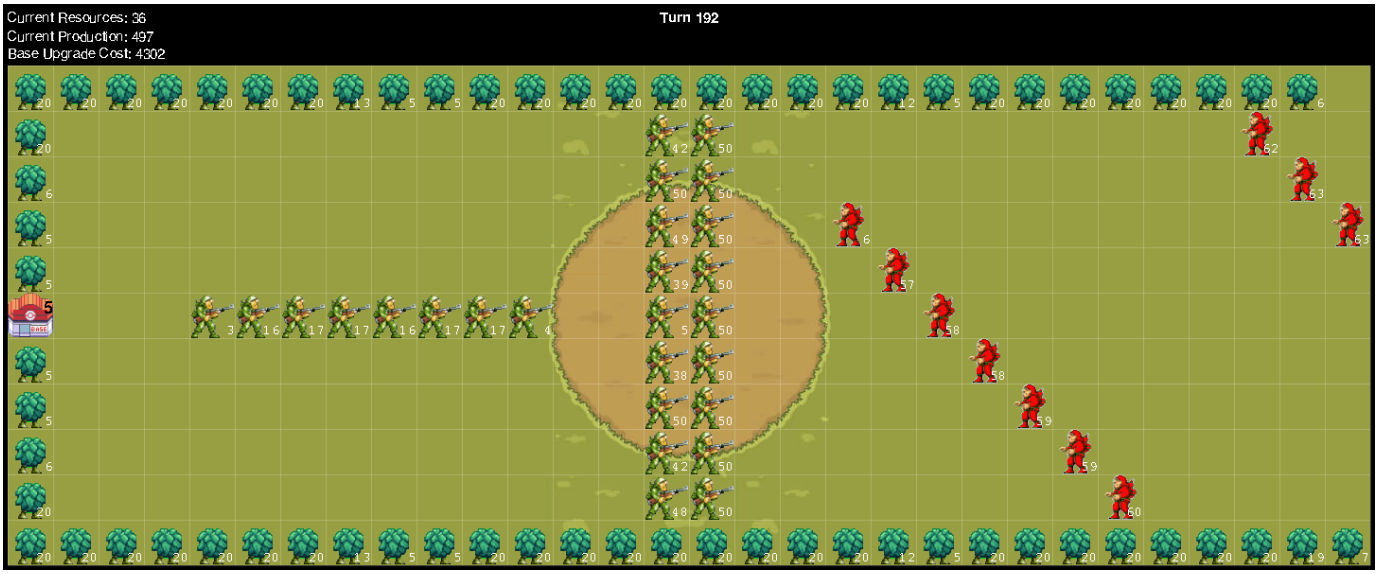

Fig. 2. Example frame of the adopted strategy for difficulty 1.

Unfortunately, the author did not successfully implement it. The major issue was that this should be a multi-agent neural network. The author tried to train a general neural network that would then decide the best action for each cell with a allied soldier. It was implemented by iterating over all cells with allied soldiers and then using the network to predict the best action. The network would be able to output the following set of actions:

- do nothing;
- move all soldiers up, down, left or right;
- split, horizontal or vertically, the soldiers in half;
- upgrade the base;
- recruit the maximum possible amount of melee or ranged soldiers.

As outputs, it would require information about the current resources, difficulty, upgrade cost, current production, position, amount and the type and quantity of the cells within a distance of three cells.

As there was no easy or feasible way to measure how good an action is in the long term, all the network learned was to make no moves, which was not very helpful.

This lead the author to implement a more classical approach, using a a blend between a decision tree and a state machine.

V. Performance Analysis

As mentioned earlier, the time limit for the agent to take a set of actions was two hundred milliseconds. The implemented solution takes, in average, between ten and twenty millisec-

onds. This is clearly bellow the imposed limit and may suggest that a more complex agent should, maybe, be implemented to take advantage of all the time available, but only if that granted a higher score.

It is worth mentioned that this limit was not an issue during the agent development.

VI. Results

In a five game mean, the developed agent is capable of reaching a score of 655 at difficulty 0 and 585.8 at difficulty 1. Additionally, the agent is able to inflict damage on the enemy, measured by the retard. The final (mean) scores for difficulty 0 and 1 were, respectively, 1037.12 and 761.96.

A note should be given to the strategies of the allied melee soldiers. The author had idealized that these soldiers, in addition to their attack goal, could be used to allure the enemies in such way that would be beneficial for the defense. However, all the tested methods caused the same problem. While being allured, the enemies would gather in amount that would, in the medium term, severely harm the overall defense. This is the reason why the only goal of the melee soldiers is to stay hidden and reach the last column.

VII. Conclusion

On the whole, the author believes that the major goals of the assignment were achieved. Not only the developed agent is able to control and take deliberate actions to protect its base, but also can intentionally attack the enemy while that is beneficial.

Unfortunately, learning algorithms were not successfully implemented. In counterpart, this assignment actually required other concepts that were more inline with the classes.