LINGI1131- Project 2020
Captain Sonoz

Group 095

LADERRIERE Loïc (NOMA 4426-19-00)

PUCHE Fabian (NOMA 4442-19-00)

Each of the player strategies

All along our project we implemented two totally different artificial intelligence (RandomAI and XxD4rkPulv3r1sat0rxX).

RandomAI

It is a simple implementation that makes a lot of random actions. The first position is random. Displacements are random too but there are less possibilities to go surface contrary to north, east, south or west in order to don't waste too much turns because of going to surface means losing three turns. This is a lot. The loading of different items is also random. Each turn we increase the number of charges of an item in the list (drone, sonar, missile or mine). When one of them arrives to three charges, it is automatically fired at a random position and its charge returns to zero. It is the same case for mines which are placed and fired at the same time.

XxD4rkPulv3r1sat0rxX

This one is a more advanced artificial intelligence. Its strategy is to wait then collect information and finally pulverize enemies. The first position is random. This strategy is based on sonar and missiles. All along the party it tracks displacements of all enemies. A sonar is launched on the third turn. This time, the number of possible positions for each player is therefore reduced to the number of rows multiplied by the number of columns of the map. For instance, with a map 10 × 10. Nineteen possible positions exist for each player. The possible positions are compared to the path of the player since the beginning of the game. This allows reducing even more the number of possible positions by deleting impossible paths. Possible positions are approximately reduced from 25% to 50%. Each time the player is moving his possible positions is updated and impossible positions are also removed from the list. After launching the sonar, we only charge missiles for the rest of the game. A missile is fired only if a suspicious possible is in range. If a missile is fired on a position which does not make damage this position is removed for all the players. If the missile hurts. Two damages means we found the player and one damage means the enemy is just one tile next to the position of the missile. According to the displacement of this artificial intelligence submarine displacements are different depending on the situation. There are two totally different behaviors (random or rush). On the one hand, if the submarine has no missile loaded it moves randomly. On the other hand, if a missile is ready to be fired but no one suspicious position is in range, the artificial intelligence will find the shortest path to it using the BFS algorithm. If the submarine is in range of a possible position and a missile is charged, the missile is fired, and it tries to stay at most three tiles of this position which allow countering submarines which are playing with mines. The goal is to keep some distance with the enemy while shooting them from the maximum range.

The design of your implementation and the choices that you made.

For the random intelligence, a lot of things implemented into it were not used but it intended to be a base for the more advanced one.

The redundant code has been postponed to functions in order to make it reusable. We tried to create a function by operation. It allows to have a clear code and do not have too large functions with many operations. For instance, we have separated functions for charging, moving and so on.

Each submarine has a submarine record containing all the information about it and its adversaries. This record is used and updated with each message received and then returned to the submarine port pending the next message.

Any extension that you implemented

As extension we only made a map generator. This generation is based on grouping small maps with each other. We design a lot of patterns. For instance, a 10 × 10 map is constructed with four 5 × 5 patterns randomly picked in the list of all patterns.