**Team:**

**Alex Roemelt**

**Pingchuan Ma**

**Yu Xiang**

The work will be a collaborative effort of the three members. The most difficult part is the algorithm and how to be implemented, this part will be discussed among the team. Each member’s contribution will be clearly indicated at the end of the project (during the poster session).

**Problem definition:**

We are going to write a tank game. The tank game has two players: the user and the robot, each controlling a tank. The game happens on a squared map. At the (top/bottom) center locates the players’s flag. Within the map, walls and fuels are randomly distributed, the tank can not go over the walls. There is a time limit for the game. Within the time limit, whoever takes the other side’s flag first wins the game. If no flag is taken during the time limit, the game is won by the one who collects the most fuels when the time limit is reached.

So the player and robot prior goal is to protect their own flag, if it is possible to attack and occupy the other flag within the time limit and before their own flag being taken, they should attack the other’s flag, otherwise they should try to collect as much fuel as possible during the time limit (while they should not walk away from their flag too much so that their flag will not be taken during the time limit).

In each game, the game could be won by either trying to take other sides’ flag or collecting more fuels than the opponent before the time limit if no flag being taken.

If one side decides to attack the flag, the other side could tries to protect and significantly increasing the time that the flag being taken. When a tank is being attacked several times, it will disappear from the map for a few seconds, after that it will reappear on the map, exactly above their own flag. However, when a flag is without protection by the tank, a flag be taken by the other side in a short time.

As the map is randomly generated, given the time limit, it could happen that it is more meaningful to collect the fuels only rather than trying to take the other’s flag (it may happen the attacker may fail to take the flag within the time limit), or it could happen that it will be an easy win if one side tries to take the other’s flag directly.

The robot (AI) will evaluate the situation of the game in each step and decides which is the best move/strategy. The strategy can change during the game, e.g. if the player’s tank walks far away from the flag, the robot may change the strategy from collecting the fuels to attacking the player’s tank or taking the player’s flag.

The strategy is based on the fact that the AI knows all the information on the map. Later, we may increase the interest of the game by only allowing the AI to know partial information of the map.

**Environment**:

The game is going to be implemented with pygame.

**Approach:**

The main task of the algorithm which controls the AI’s moves during the game is to decide in every given situation which is the best strategy to choose and which is the best move depending on this strategy. The main parameters for the decisions will be the information the AI has about the current state. The amount of information could vary depending on several levels of difficulty which could be chosen by the human player. Given the maximum amount of available information about the game state, i.e. position of the opponent, position of every fuel resource and position of the walls. The AI should be able to make reasonable decisions based on computation of the parameters. Our ambition is to create an AI which can identify the best strategy during the game and thus make it hard for a human player to compete.

Choosing the right strategy, the AI should also take the randomly generated map into consideration. Aspects of the map to be created are fuel positions and walls. The walls and fuels are randomly generated, which means on average the fuel items are positioned in a way that none of the two players gets an advantage because of the positioning.

**Evaluation & Expected Results:**

The algorithm should strive to deliver the best performance for one player in every given state of the game. This should be achieved by computing every available information about the games state and identifying suitable strategies in order to win the game. In practice, the AI should strive to win the game, which means the strategy may be changed during the game with the changing of the situation. The initial strategy could be trying to collect as many fuel items as possible while defending its own flag or attacking the opponent's flag, and play this strategy successfully until the situation on the map is changed enough that it is time to change the strategy. I.e, it should react to situations during the game, i.e. detect the chance to attack the opponents flag and switch to the corresponding strategy if there is enough chance to succeed.

The performance of the algorithm can be measured by its success in choosing the most successful strategy.

At the highest difficulty level the AI should be able to beat human players by computing every available information. For further implementation, lower difficulty levels should leave certain information out and therefore make the AI’s decisions less successful.

**Hardware:**

No specific hardware is required.

**Excluded Presentation Date:**

Both date OK, but prefer 2nd of February.