# TEAM RECIPE AI DESIGN DOCUMENT

Team Members:

Kenji Johnson, Woodlin Smith, Jordan Baumeister,

Kendric Thompson, Mangesh Sakordekar

## REQUIREMENTS FOR PROGRAM

Navigation, pursuit, attack, and evasion.

Victory over target, mouse, turret, and sentry.

## DEVELOPMENT PROGRESSION

- Weekly feature sprints

### March 15

- Write user stories for each requirement
- Familiarize tank building

### March 22

- Review written user stories (review necessity, clarity, and detail)
- Discuss methods to meet first task of destroying a target

### March 29

- Create graph class for map exploration
- Implement attack function

### April 5

- Testing of spendAP
- Complete shortest path algorithm for move function
- Weigh the graph appropriately for attack, evade, and explore
- Analyze range and map data

### April 12

- MVP completion
- Implement evade and attack to pathfinding

**April 19**

- Completion of additional features
- Other improvements to AI

**FUNCTION OVERVIEW**

At a very high level, our tank's AI will prioritize information gathering. It will build multiple maps of the gamefield, which will store "safe zones"("safemap"), the 8 most recently visited map tiles("recent"), a persistent map of obstacle locations("fullmap"), a map that marks if a tile has line of sight on an enemy, and a map indicating if a tile has been visited("visited"). Safe zones are tiles that enemy tanks cannot fire upon due to their position.

Overall, our AI will have two distinct modes:

**1. Information Gathering:** This is the default behavior. In this mode, the AI will prioritize discovering more information about the map. At the start of every turn, the AI will add new obstacle locations to its persistent map. It will then mark its current location as "visited", and add it to its list of the 8 most recently visited tiles, removing the first one if the size is too large. Then, if no enemy tanks are detected within the AI's radar, the AI will prioritize finding a tile adjacent to it that fulfills 2 requirements, the first of which is mandatory:

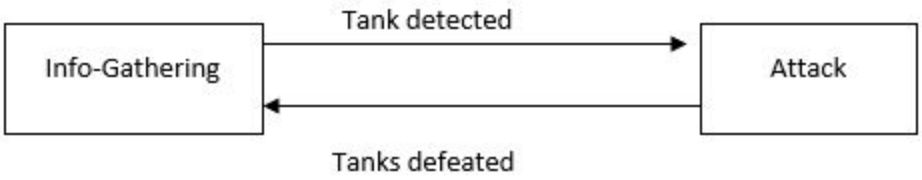
1. The adjacent tile must be a valid tile to move to. This means that it cannot be a tile with a rock or water on it. This requirement is fulfilled by checking the adjacent index with its corresponding value in the obstacleMap passed in each turn. It also cannot be one of the AI's 8 most recently visited tiles. This prevents the AI from backtracking and getting caught in a cycle in its exploration.
2. The adjacent tile should be one that has not been visited yet. The AI checks this by examining the "visited" map at the adjacent tile's index. If the "visited" map has a value of 0 at this location, this indicates that the tile has never been visited, and should be moved to.

If it cannot find a tile that fulfills both requirements, the AI will simply stay where it is. If it cannot find one that fulfills the second requirement, for each adjacent tile it will count the amount of unvisited tiles that are adjacent to them. It will then prioritize moving to the

one with the most amount of adjacent unvisited tiles. If it cannot find any unvisited tiles this way, which can happen in closed environments, it will choose a tile that it has visited with the most adjacent unknown tiles and move there. It continues this process until it discovers an enemy tank.

**2. Attack:** If our tank encounters an enemy, it will start moving in a different fashion. It will calculate “safe zones” that it will prioritize moving to by the end of its turn. These are areas that enemy tanks cannot fire upon. These “safe zones” are stored in the “safemap”, which is updated each turn. As it marks safe zones, it will also mark spots that have a line of attack on enemy tanks, which are the spots marked within the safe map. Our tank’s attack algorithm will be to move into a safe zone, move into an attack position, attack, and then move back into its safe zone. It will stay in this mode until all visible tanks are defeated. If our tank becomes “stuck” in this mode (i.e. moving back and forth due to the presence of an obstacle), it will target the obstacle and destroy it. This occurrence will mainly happen within closed environments.

A general diagram of the AI’s behavior is shown below




**PLACEMENT OF FUNCTION**

All maps and persistent AI info are updated at the start of the spendAP function. The move function will either use the attack moves (including safezones) or the explore moves depending on the presence of enemy tanks. However, the attack function determines whether or not the tank is stuck and it needs to target an obstacle. As the AI is heavily dependent on being able to move and attack multiple times per turn, all SPECIAL points are put into range, radar, and ap, with range and radar guaranteed to be equal.