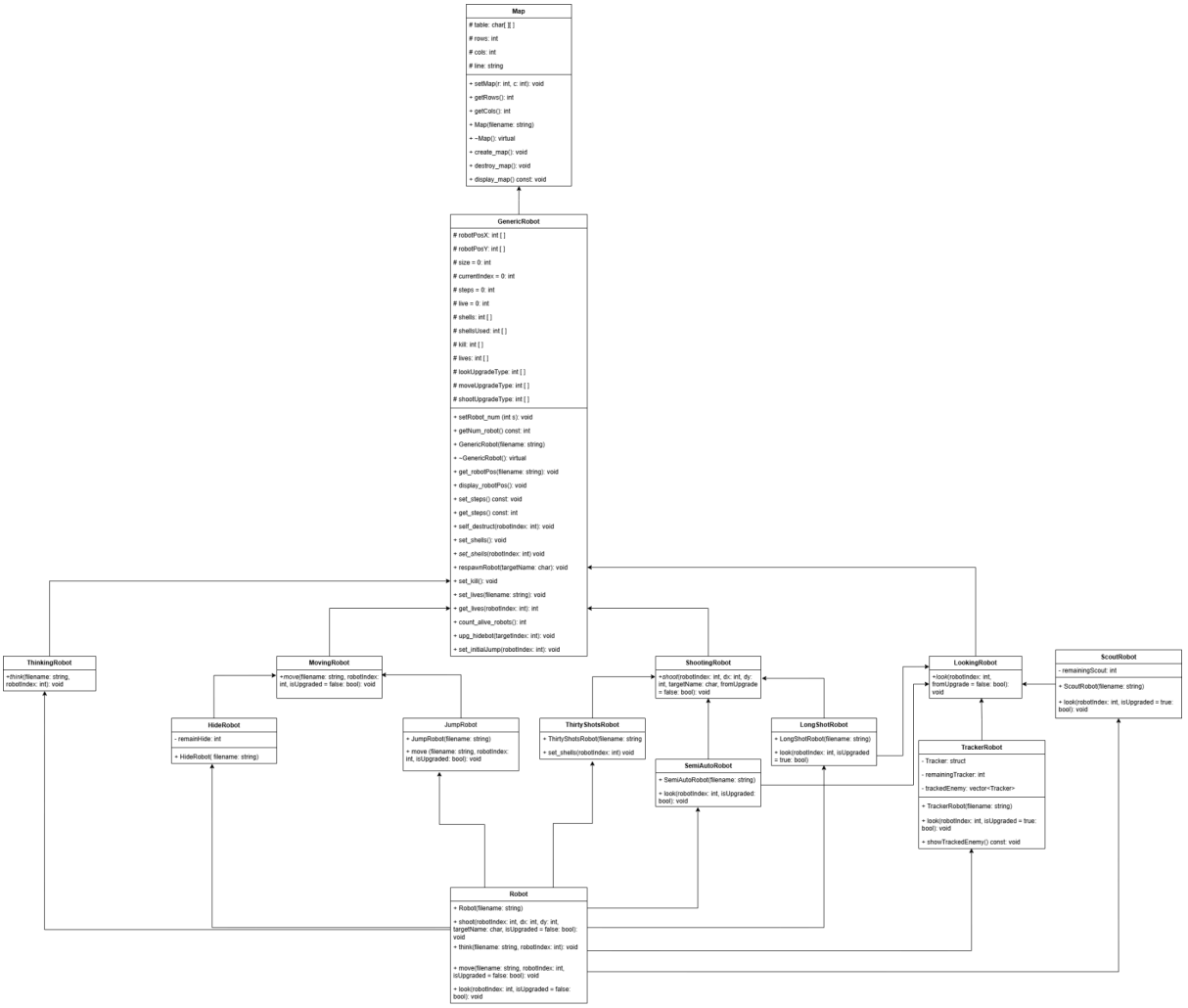
b) Design documents such as class diagrams, flowcharts and/or pseudo codes in PDF format to
explain your work.
c) Screen-shots and explanation of your program running compiled into a document in PDF format.

Class Diagram



Pseudo-codes

Start
Class Map:
    Attributes:
        table: 2D array of characters
        rows: integer
        cols: integer
        line: string for reading from file

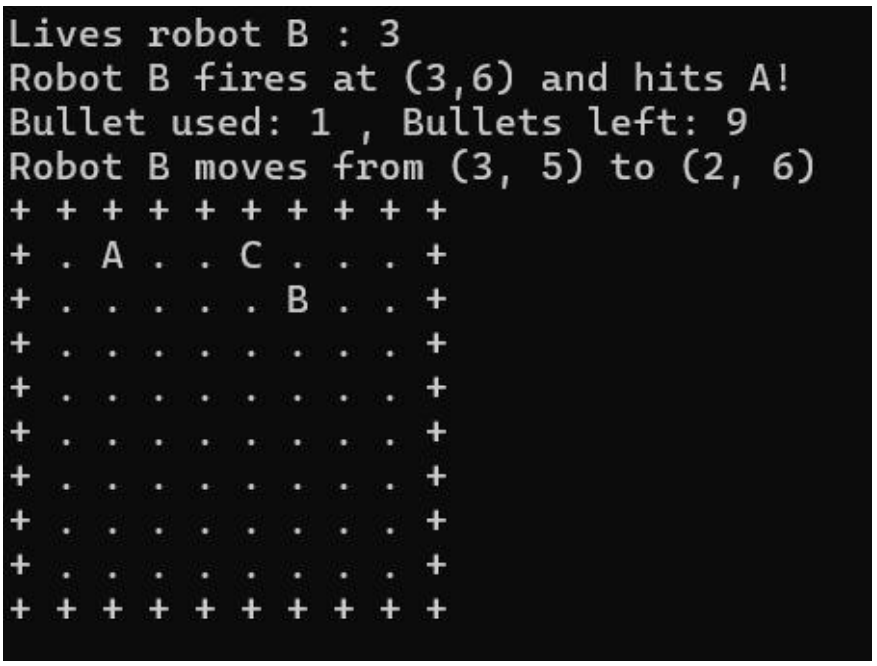
```
    Methods:
        Constructor(filename):
            Open file
            For each line in file:
                If line contains "Rows":
                    Extract row and column values
                    Set map size
            Close file

        Destructor():
            Print "Game Over"
            Call destroy_map()

        setMap(r, c):
            Set rows = r, cols = c

        getRows():
            Return rows

        getCols():
            Return cols

        create_map():
            Allocate table with dimensions rows x cols
            Fill borders with '+', rest with '.'

        destroy_map():
            Deallocate memory used for table

        display_map():
            Print the table to console
Class GenericRobot extends Map:
    Attributes:
        robotPosX, robotPosY: arrays for robot coordinates
        size, currentIndex, steps, live: integers
        shells, shellsUsed, kill, lives: arrays
        lookUpgraded: boolean array
        lookUpgradeType: integer array

    Methods:
        Constructor(filename):
            Call Map constructor
            Open file
            Find number of robots and set size
            Allocate all arrays
            Initialize lookUpgradeType to 0 for all robots
```

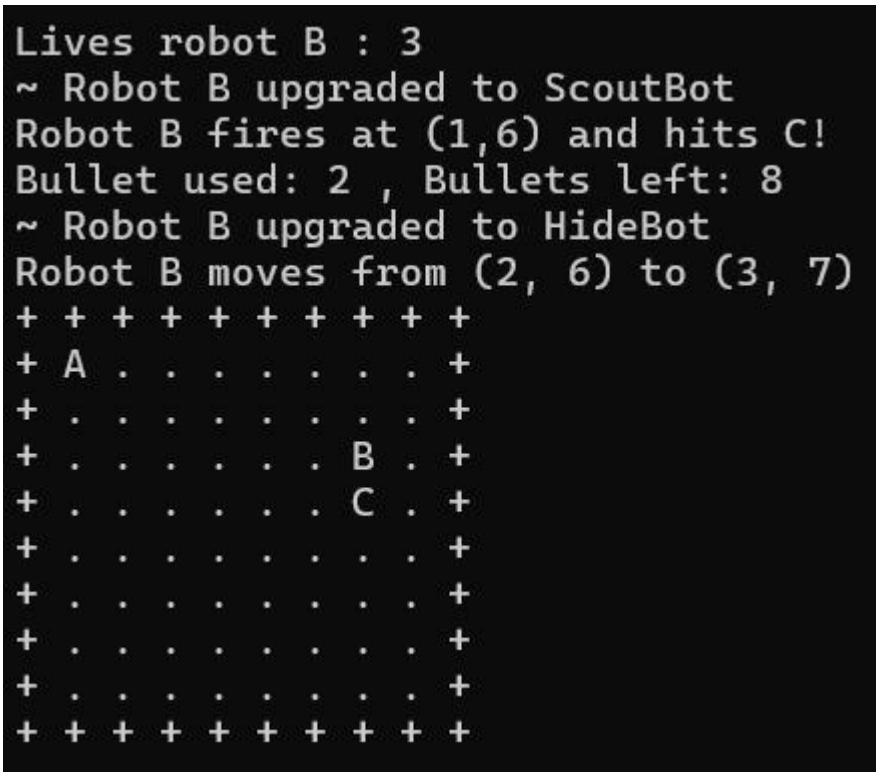
```
Destructor():
    Deallocate all arrays

setRobot_num(s):
    Set size = s

getNum_robot():
    Return size

get_robotPos(filename):
    Open file
    For each line:
        If line contains robot position:
            Parse coordinates
            Store in robotPosX/Y at currentIndex
            Increment currentIndex
    Close file

display_robotPos():
    For each robot:
        Convert index to letter
        Place letter in table at robot position

set_steps(filename):
    Open file
    For each line:
        If line contains "steps":
            Extract number and set to steps
    Close file

get_steps():
    Return steps

self_destruct(robotIndex):
    Remove robot from map (set position to '.')

respawnRobot(targetName):
    Convert targetName to index
    Remove from old position
    Find random empty cell
    Update robot position

set_shells():
    Set all shells = 10
    Set all shellsUsed = 0

set_kill():
    Set all kill = 0
```

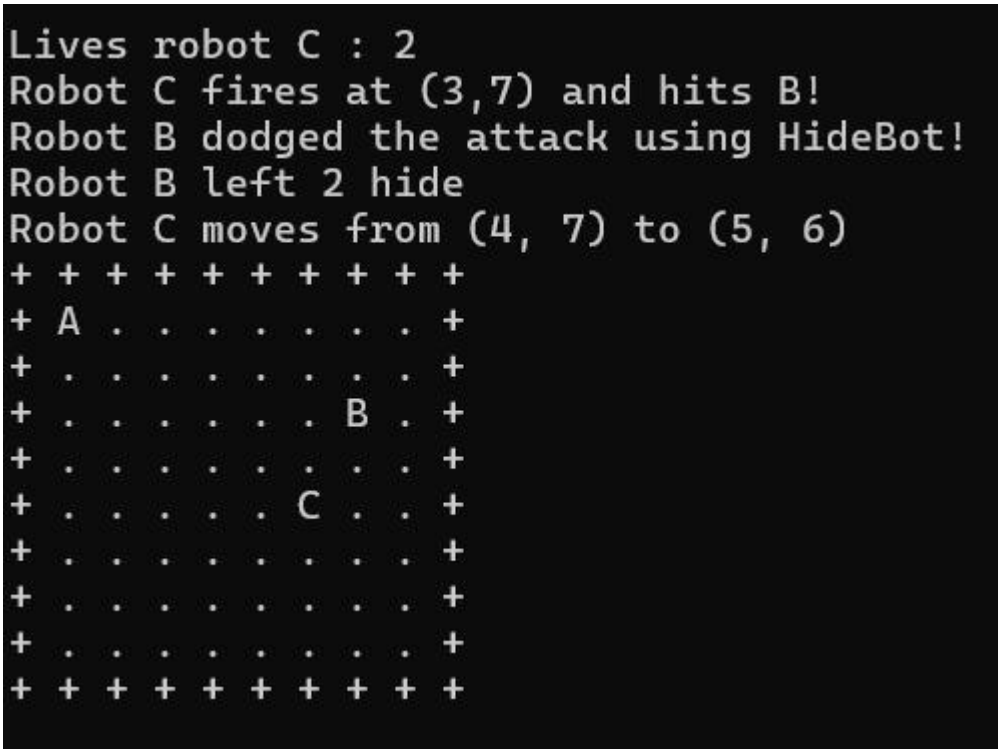
```
        set_lives(filename):
            Open file
            For each line:
                If line contains "lives":
                    Extract value and set to live
            Close file
            Set all lives[i] = live

        get_live(robotIndex):
            Return lives[robotIndex]

        check_map():
            Count all non-dot, non-wall cells
            Return count (alive robots)

Class ShootingRobot extends GenericRobot:
    Method:
        shoot(robotIndex, dx, dy, targetName): Abstract

Class MovingRobot extends GenericRobot:
    Method:
        move(filename, robotIndex): Abstract

Class ThinkingRobot extends GenericRobot:
    Method:
        think(filename, robotIndex): Abstract

Class LookingRobot extends GenericRobot:
    Method:
        look(robotIndex): Abstract

Class ScoutRobot extends LookingRobot:
    Attribute:
        remainingScout = 3

    Constructor(filename):
        Call GenericRobot constructor

    look(robotIndex):
        If no scouts left, return
        Print scanning message
        For each cell:
            If contains robot (A-Z):
                Print coordinates
        Decrease remainingScout

Class Robot extends ShootingRobot, MovingRobot, ThinkingRobot, ScoutRobot:
```

```
    Constructor(filename):
        Call ScoutRobot constructor

    shoot(robotIndex, dx, dy, targetName):
        Compute target position
        If shooting itself, print error and return
        If target out of bounds, print and return
        Else:
            70% chance to hit
                If hit:
                    Print hit message
                    Respawn target
                    Increase kills
                    Decrease target lives
                Else:
                    Print miss message
            Decrease shells, increase shellsUsed
        For each robot:
            If shells <= 0, self-destruct
            If lives < 0, self-destruct

    think(filename, robotIndex):
        Call look()
        Call move()

    move(filename, robotIndex):
        Remove robot from current position
        Randomly pick a direction (8 possible)
        Compute new position
        If new position valid and empty:
            Update position in table
main():
    Initialize Robot with input filename
    Call create_map()
    Call get_robotPos()
    Call set_steps()
    Call set_lives()
    Call set_shells()
    Call set_kill()
    While steps not exhausted and more than 1 robot alive:
        For each robot:
            Call think()
            Call shoot()
        Display map
    End simulation
    Display final stats (kills, lives, shells used, etc.)
End program
```

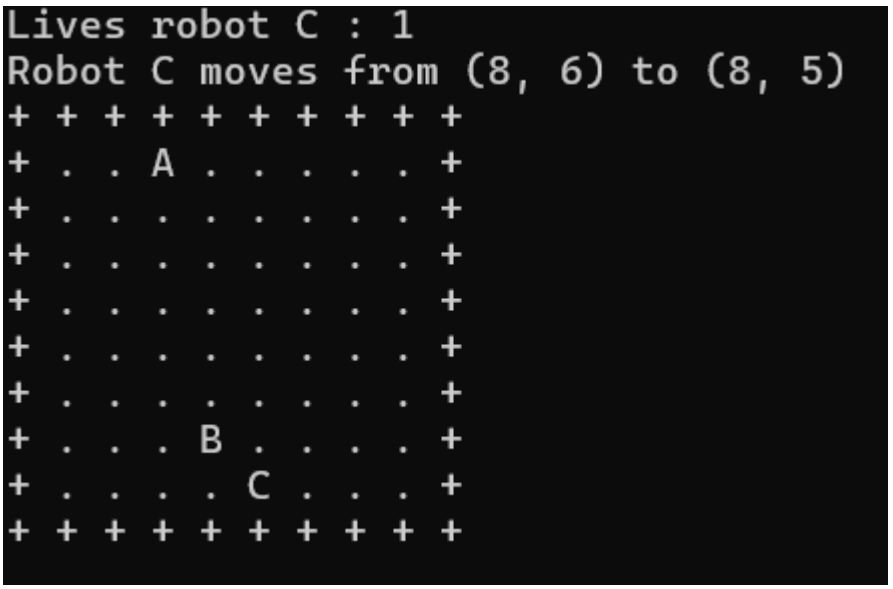
Screenshot to show the function is working

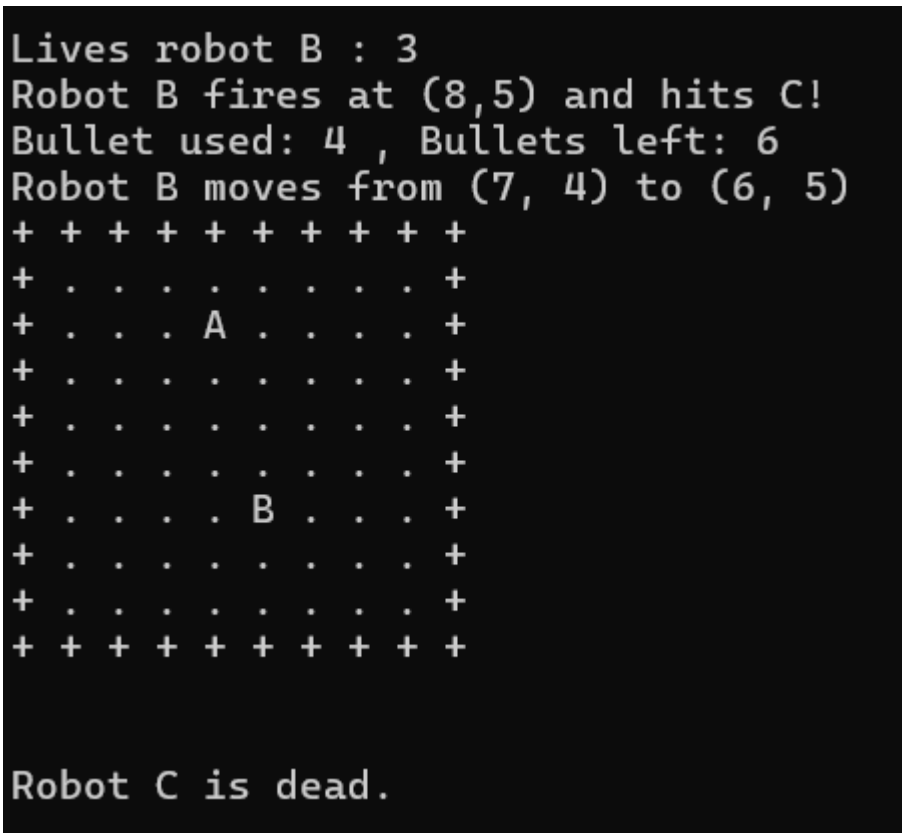

Lives robot B : 3
Robot B fires at (3,6) and hits A!
Bullet used: 1 , Bullets left: 9
Robot B moves from (3, 5) to (2, 6)
+ + + + + + + + + + +
+ . A . . C . . . +
+ . . . . . B . . +
+ . . . . . +
+ . . . . . +
+ . . . . . +
+ . . . . . +
+ . . . . . +
+ . . . . . +
+ + + + + + + + + + +


Lives robot B : 3
~ Robot B upgraded to ScoutBot
Robot B fires at (1,6) and hits C!
Bullet used: 2 , Bullets left: 8
~ Robot B upgraded to HideBot
Robot B moves from (2, 6) to (3, 7)
+ + + + + + + + + + +
+ A . . . . . +
+ . . . . . +
+ . . . . . B . +
+ . . . . . C . +
+ . . . . . +
+ . . . . . +
+ . . . . . +
+ . . . . . +
+ + + + + + + + + + +



Robot B first shoot kill robot A and got upgraded to ScoutBot
Robot B second shoot kill robot C and got upgraded to HideBot
Robot C take revenge but fail because of hide upgrade





Robot C got last HP and kill by robot B, removed from the map and wont display robot C again

```
Lives robot B : 2
TrackerRobot is using tracker, but surrounding have no enemy
Robot B fires at (7,4) and hits C!
Bullet used: 4 , Bullets left: 6
~ Robot B upgraded to JumpBot
Remaining Jump upgrade 2
Robot 'B' jump from (8,3) to (3,4)
+ + + + + + + + + +
+ . . . . . . . . +
+ . . . A . . . . +
+ . . C B . . . . +
+ . . . . . . . . +
+ . . . . . . . . +
+ . . . . . . . . +
+ . . . . . . . . +
+ . . . . . . . . +
+ + + + + + + + + +
```

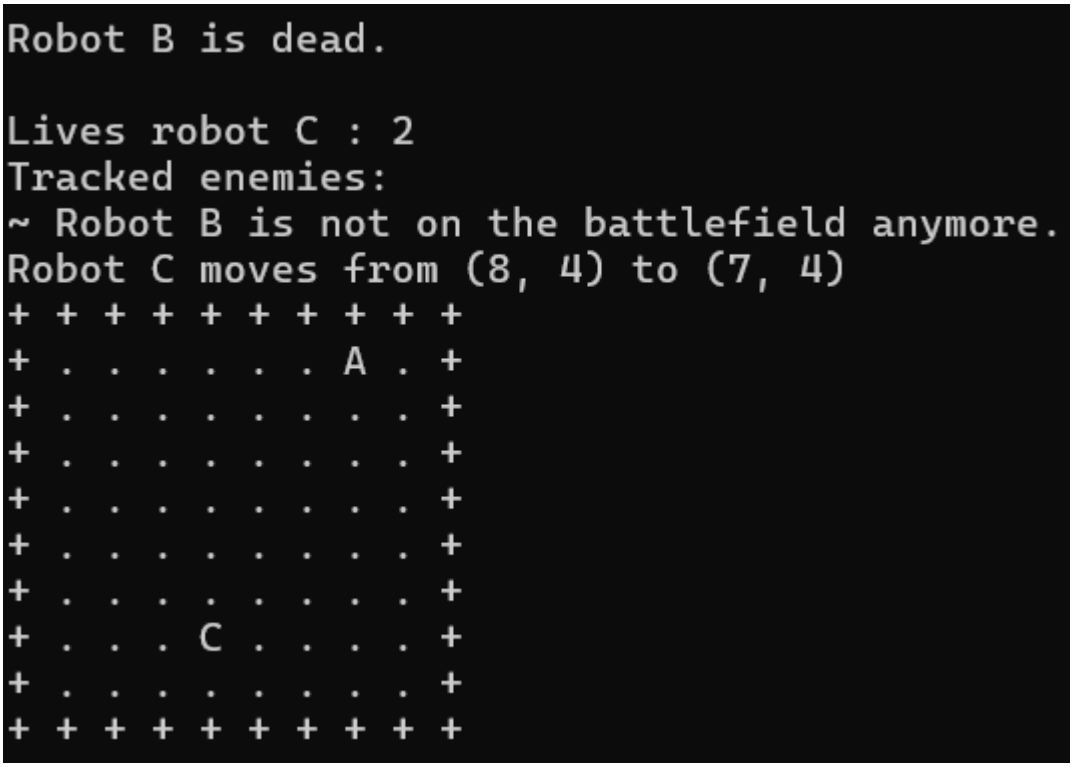
Second kill of robot B got upgraded to JumpBot, 50% chance to use the ability.

```
Lives robot A : 1
Look upgrades already taken by other robots. No upgrade applied.
Robot A moves from (1, 6) to (2, 7)
+ + + + + + + + + +
+ . . . . . . . . +
+ . . . . . . A . +
+ . . . . . . . . +
+ C . . . . . . . +
+ . . . . . . . . +
+ . . . . . . . . +
+ . . . . . . . . +
+ . . . . . B . . +
+ + + + + + + + + +
```

Since we only have 2 upgrades for look, robot B and C have already taken ScoutBot and TrackBot so robot A have no upgrades here.

```
Lives robot C : 3
Tracker planted on robot B.
Tracked enemies:
~ Robot B is now at (6, 7)
Robot C fires at (6,7) and misses the shot to B
Bullet used: 4 , Bullets left: 6
Robot C moves from (7, 7) to (6, 8)
+ + + + + + + + + + +
+ . . . . . . . . . +
+ . . . . . . A . . +
+ . . . . . . . . . +
+ . . . . . . . . . +
+ . . . . . . . . . +
+ . . . . . . B C . +
+ . . . . . . . . . +
+ . . . . . . . . . +
+ + + + + + + + + + +
```

```
Lives robot C : 2
Tracked enemies:
~ Robot B is now at (7, 6)
Robot C fires at (7,6) and hits B!
Bullet used: 5 , Bullets left: 5
~ Robot C upgraded to JumpBot
Robot C moves from (8, 5) to (8, 4)
+ + + + + + + + + + +
+ . . . . . . A . . +
+ . . . . . . . . . +
+ . . . . . . . . . +
+ . . . . . . . . . +
+ . . . . . . . . . +
+ . . . . . . . . . +
+ . . . . . . . . . +
+ . . . C . . . . . +
+ + + + + + + + + + +
```



Tracker will plan to others, although it is dead but still will update all