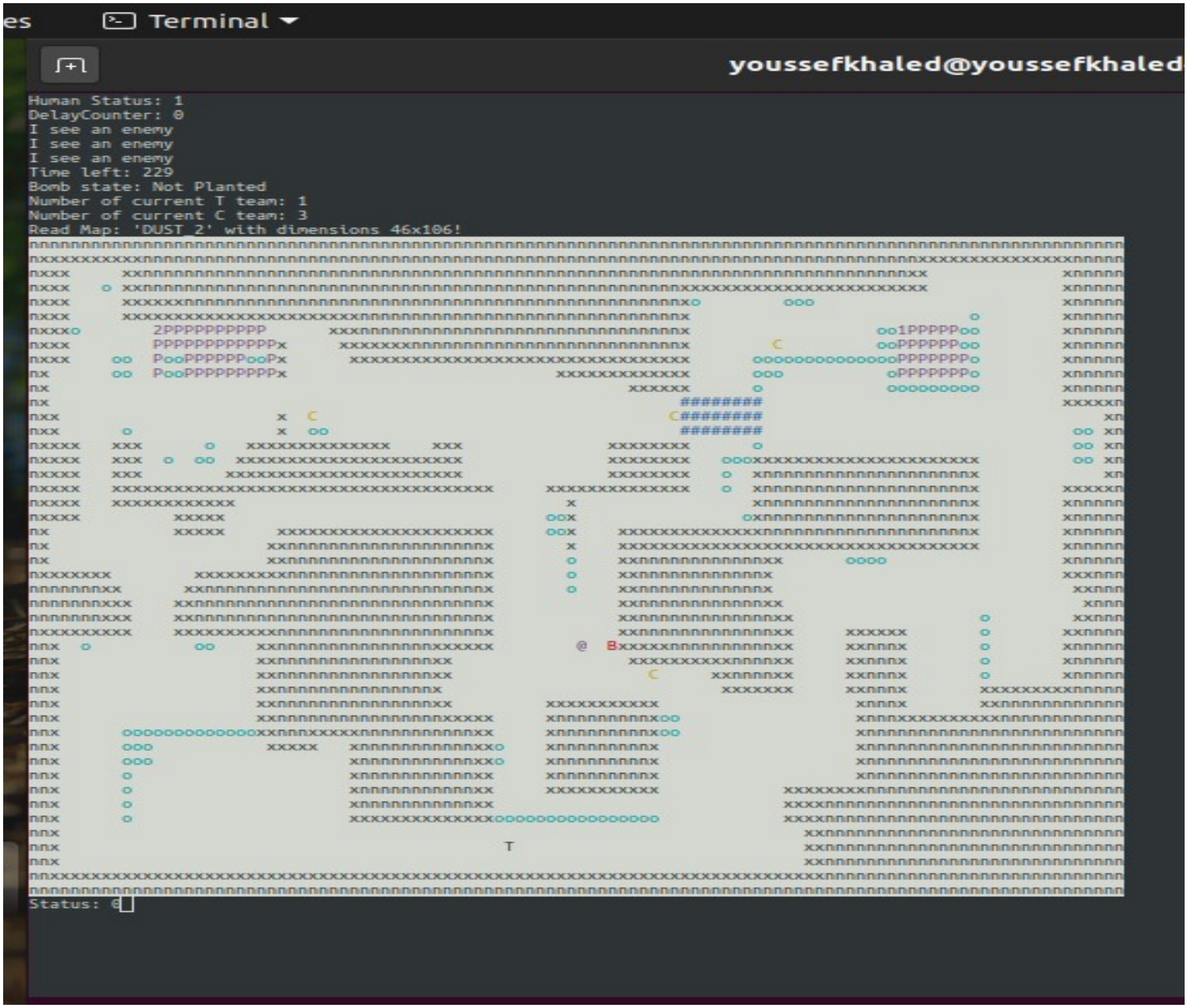
Part 2:


Terminal
youssefkhaled@youssefkhaled
Human Status: 1
DelayCounter: 0
I see an enemy
I see an enemy
I see an enemy
Time left: 229
Bomb state: Not Planted
Number of current T team: 1
Number of current C team: 3
Read Map: 'DUST_2' with dimensions 46x106!
Status:

Terminal

youssefkhaled@youssefkha

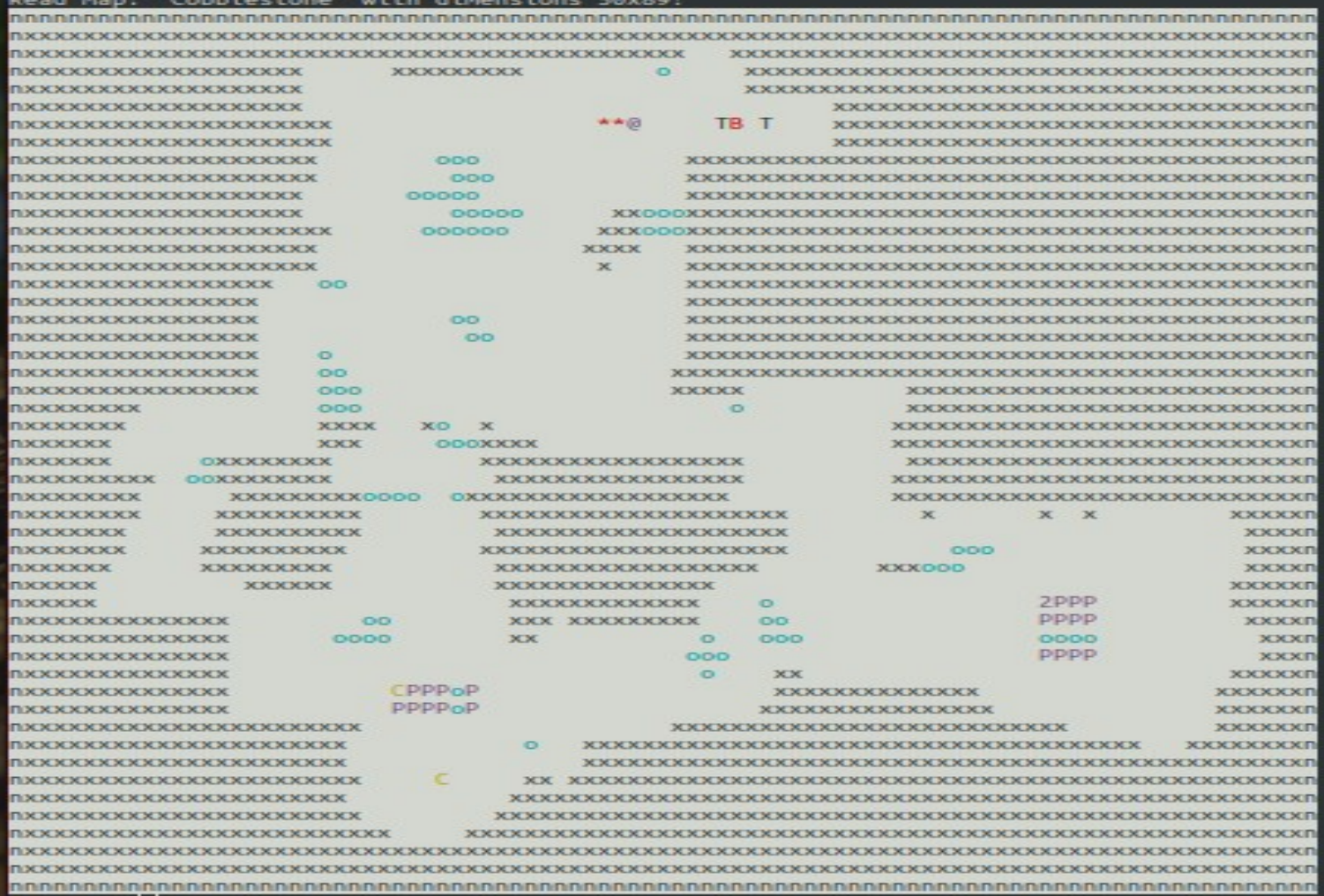
```
Human Status: 1
DelayCounter: 0
I see an enemy
I see an enemy
I see an enemy
going to the bomb site
Time left: 266
Bomb state: Not Planted
Number of current T team: 5
Number of current C team: 5
Read Map: 'DUST_2' with dimensions 46x106!
```

```
Status: 0
```

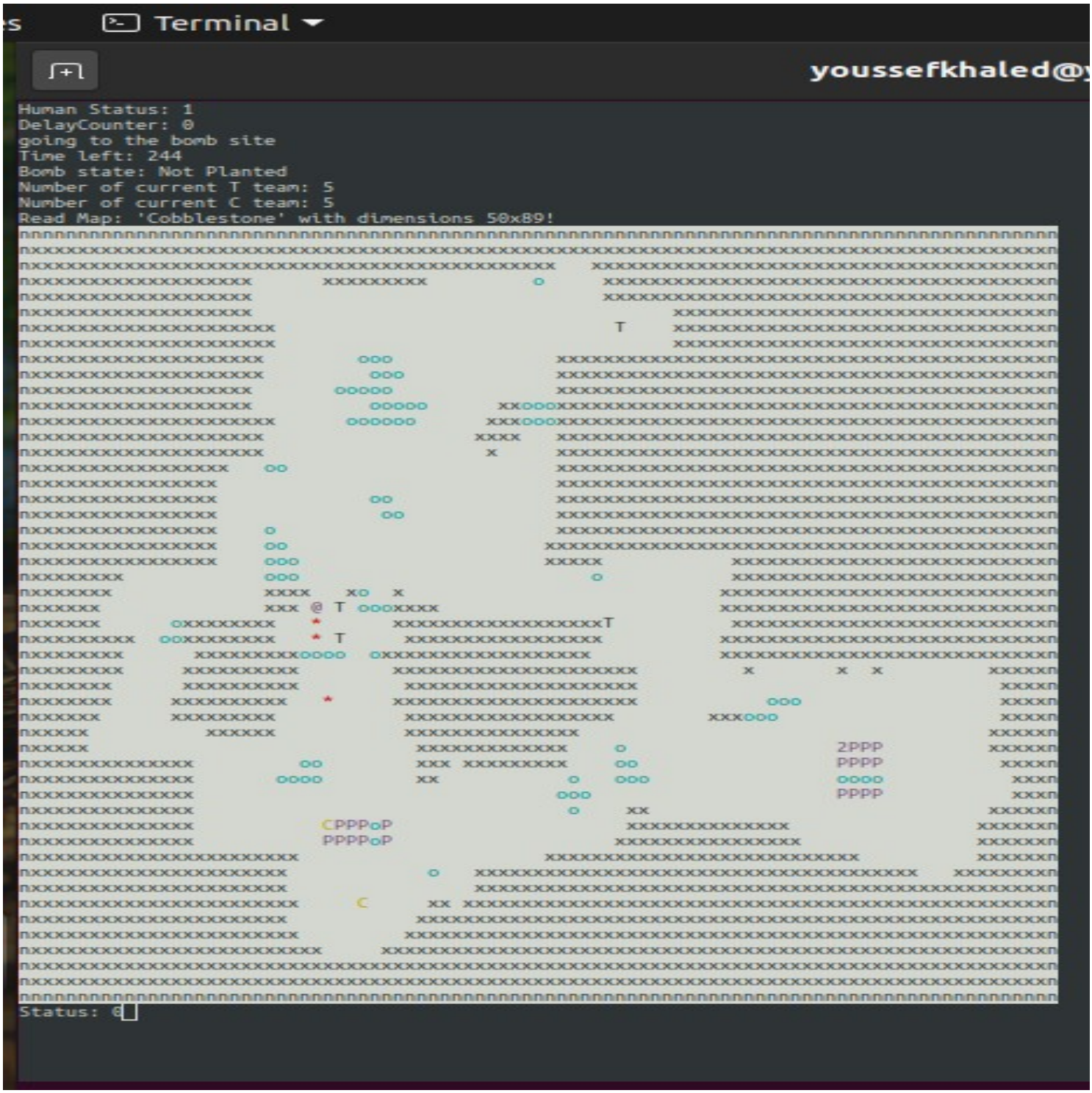
Terminal

youssefkhaled@

```
Shooting Proj .....
Human Status: 1
DelayCounter: 0
I should pick the bomb
I should pick the bomb
I should pick the bomb
I should pick the bomb
Time left: 285
Bomb state: Not Planted
Number of current T team: 5
Number of current C team: 5
Read Map: 'Cobblestone' with dimensions 50x89!
```



```
Status: 0
```


Terminal
youssefkhaled@
Human Status: 1
DelayCounter: 0
going to the bomb site
Time left: 244
Bomb state: Not Planted
Number of current T team: 5
Number of current C team: 5
Read Map: 'Cobblestone' with dimensions 50x89!
Status: 0

- Discussing the path-finding algorithm used and its efficiency:
  - I used BFS for finding the shortest path of the AI.
  - BFS(Breadth First Search) uses Queue data structure for finding the shortest path.
  - BFS can be used to find single source shortest path in an unweighted graph, because in BFS, we reach a vertex with minimum number of edges from a source vertex.
  - The Time complexity of BFS is O(V^2).
  - After I find the shortest path, I store it in a vector of pairs (parent, point).
  - Then I reverse that vector to start from the bot-play to the target point.
  - After that every tick of time the bot-play move using that vector to the next point until it arrive to the destination.
  - I didn’t use Dijkstra because in Dijkstra every edge have a certain weight but I don’t need that in bots AI because every edge have the same weight.
  - I could speed up BFS from the player to it’s target by doing bi-directional search.
  - A bi-directional search is basically doing a BFS from the source and from the target at the same time, on step from each – until the two fronts meet each other.
  - Complexity of BFS at worst case is O(B^d), (B is the branch factor, the degree of each node) – and (d is the depth of the solution).
  - On the other hand the complexity of the bi-directional BFS at worst case O(B^(d\2)*2) = O(B^(d\2))nodes, which is much smaller in large graphs.

- Discuss how your AI ran:
  - First I call BFS function and give it (level-Reference, player, target, RNG).
  - According to every target and the random number the BFS find the suitable path for it.
  - If the target equals 1 and the RNG = 0 then the BFS will set the first Bomb location as the destination.
  - If the target equals 1 and the RNG = 1 then the BFS will set the second Bomb location as the destination.
  - If the target equals 3 then the BFS will set the player who carries the bomb as the destination so the bot-play can follow him and defend him.
  - If the target equals 4 and RNG = 0 then the BFS will search where the bomb planted and set that location as the destination, so the CT bot-play can defuse the bomb.
  - If the target equals 4 and RNG = 1 then the BFS will generate a new RNG and if the new RNG = 0 then the BFS will set the first Bomb location as the destination, or RNG = 1 then the BFS will set the second Bomb location as the destination..
  - If the target equals 5 that means the bomb have been planted so the BFS will set the bomb location as the destination so the T bot-play can defend it.
  - If the target equals 6 then the BFS will set the last seen location of enemy as destination.
  - After setting the path for the bot-player we run View-Finding function to see if there is any enemy on my current path.
  - If there is an enemy in our Line of Sight Up, Down, Left, or Right. Then the bot-play will rotate towards the enemy
  - If the enemy still in bot-play last direction the we shoot.