CS4341: Intro to AI
C-Term 2020
Final Group Project

Kaitlin McDermott, Parker Simpson, and Alex Tavares

For variant1 in Scenario 1, our group implemented the A* algorithm. We wrote the code based on the pseudo-code in lecture 2. To keep track of the path we used a dictionary, came_from, that took a coordinate and returned the coordinate that preceded it . To keep track of g values we used a dictionary, cost_so_far, which took a coordinate and returned the lowest cost to it. came_from and cost_so_far are updated if a lower cost for an existing coordinate is found. To calculate the heuristic values we utilized the diagonal distance function (shown below).

```
function heuristic(node) =
    dx = abs(node.x - goal.x)
    dy = abs(node.y - goal.y)
    return D * (dx + dy) + (D2 - 2 * D) * min(dx, dy)
```

We implemented a priority queue and used f(x, y) as the priority where f(x,y) = cost_so_far[x,y] + diagonal_distance(x,y). After the A* search has been completed, we reconstruct the path by following the came_from dictionary, starting with the end coordinate, until we reach the coordinate that came from the start. Using this A* algorithm we were able to find the exit for scenario1 variant1 100% of the time.

When A* was no longer enough to cut it, we decided to implement approximate Q-Learning to complete the more challenging variants. To train for scenario1, variant1 we used three features: distance to the closest exit, distance to the closest bomb, time to explosion. While using a learning rate of .2 for training we were able to find weights that allowed to the q-learning character to find the exit more than 90% of the time