State Space Description:

The state space will consist of the following:

- Discretized positions on the x-z plane (e.g. 50x50 grid).
  Each position contains the following:
  - Whether or not the position is in the line of sight of a turret (bool)
  - Contains Health Item (bool)
  - Whether or not position is the goal state (bool)
- The agent's mind map (explored positions so far).
- Whether or not the agent has a smoke grenade available (bool)
- Current Health (0, 1, 2, 3)

Mathematical Descriptions:

State Space is determined by the size of our grid X * Z

State = { (X, Z, LOS, MED, GOAL), MAP, SMK, HP }

X { 0 - (X - 1) }: X Positions.
Z { 0 – (Z – 1) }: Z Positions.

LOS: { true / false }: Boolean describing if the position is in a turret LOS.

MED: { true / false }: Boolean describing if the position contains a health item.

GOAL: { true / false }: Boolean describing if the position is the goal state.

MAP: {true / false}[]: X*Z array that represents if the corresponding position was explored by the agent.

SMK: { true / false }: Boolean describing the agent has a smoke grenade.

HP: {0, 1, 2, 3}: The current health level of the agent.

Actions = {MoveUp, MoveDown, MoveLeft, MoveRight, ThrowSmoke, Idle}

State Transitions:

T (s` | s, a):

Overall, partially stochastic environment. Transitions depend on current state (s) and action taken (a).

Movement Actions / Idle

- Remove health item from state if agent steps on a health item and increase HP.
- If a new state position in turret LOS, agent takes damage based on probability p(x).
- If HP drops to zero, end game.
- Update current agent map if current agent sight includes a new position.
- If a new position is a goal, end game.

Throw Smoke:

- Update LOS of affected positions.
- Set SMK to false.

Observation Model:

At every time step the agent will observe the following:
Obs = (V, HP, SMK)

V: Set of visible positions merged with positions already in MAP.

HP: Current HP as described in state space.

SMK: Is smoke available as described in state space.

At each time step, the agent updates its existing MAP with any new/updated position information that it observes.