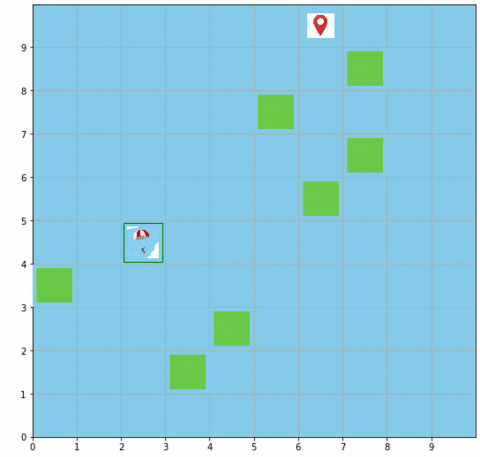
Project Title: Parachute Navigation

Purpose: Manual control of parachute has high chances of error impacting life. Hence, if the control could be automated with better decision-making, it would help with safe landing which is the most critical part and immature parachuting enthusiasts can fulfill their dreams of flying without much risk involved.

Environment : We have defined a 2D grid environment where the y-axis is in the direction of wind and x-axis is the axis perpendicular to it. The parachute(agent) is at y-coord=0 and the destination is at y-coord= shortest distance of the destination point from the x-axis passing through the starting point. And the total width of the grid environment = sum of the max distance that the parachute can travel down on either direction. Below is a basic Parachute Navigation environment defined:



Assumptions made in the current implementation:
Since the wind plays a major role in determining the next state of the agent given an action and the agent moves in the direction of the wind, for now we assume that the wind direction stays constant throughout the training period and is in the direction of y-axis in our grid.

Agent: Parachutist

The number of **time steps** for our environment will be a **constant** since the parachute cannot land until it covers the vertical distance. We assume that navigation from one grid to the next grid will cause the parachute to move down by a unit distance. Hence, total Height of the environment = total units of vertical distance of the parachute from the land.

State space and reward - Landing on Destination point (+100), Landing on Safe point (+70), Landing on unsafe point/Crash landing(-100)

A step in the direction of destination point will be given a reward of +10 and away from the destination point will be given a reward of -10.

Action space - 1. Pulling of left string which will cause the parachute to move diagonally left by 45 degrees under the force of wind in the forward direction. 2. Pulling of right string which will cause the parachute to move diagonally right by 45 degrees. 3. No action causing the parachute to move forward.

i. If the action is diagonal, the vertical distance covered will not be equal to the distance covered by doing no action and moving forward. Calculations have to be made accordingly.
ii. This has to be extended to real life application by incorporating changing wind speed and direction.
iii. Angle of movement are not only 45 degrees diagonal, but can be made continuous by varying the strength of pulling the strings
iv. By reducing the each grid units' dimension, we will get continuous set of states starting with getting one real environment structure for use. It can be obtained in real time by integrating with google maps.