Foundations Of Artificial Intelligence
LAB1 Report
Name: Pratik Bhagwat

Relevant Decisions Made in the code:

- The speed is being reduced during uphill as well as downhill as it makes no sense to have a greater speed downhill for safety purposes.
- In case of the starting point at an Island and the finish point outside the island swimming is the only option. Thus using swimming as the last option with minimum speed.
- The constant speed of marathon on easy terrains such as paved roads , open land , foot path is 10 kmph which is 2.77 m/s.
- In the worst case scenario where there is no option other that climbing a steep slope, I am considering that I am well equipped to climb the slope with a particular speed (explained later)
- The speed on ICE and MUD is kept same as both terrains have similar difficulty for walking on them.
- The Checkpoints are marked as purple color and the neighbouring pixels are also marked so that the checkpoints are properly visible.

Implementation of the cost function:

- Here the cost is considered as time taken to travel from one pixel to another.
- The time is calculated as Distance between two **points** and the speed taken to reach from one point to another. (The point is considered as pixel and the elevation of that pixel)
- The distance is calculated by simple geometry and the speed is calculated as the cos component of the speed where the angle is the angle of elevation between two points.
- To further reduce the speed due to gravity or very high steepness another cos component is added to the speed. Thus the speed is not SPEED cos^2(angle of elevation) The 2nd cos component takes care of the steepness and gravity

Implementation of the Heuristic.

- The Heuristic is calculated as the Manhattan Distance between two points / max speed possible in the entire terrain.
- This will make sure that we are calculating the lowest time possible to reach the destination.

Speeds in different seasons:

Summer

```
speedMap = {
    (248, 148, 18): 2.7777,
    (255, 192, 0): 1.38,
    (255, 255, 255): 2.7777,
    (2, 208, 60): 1.9444,
    (2, 136, 40): 1.38,
    (5, 73, 24): 0.0000000000000001,
    (0, 0, 255): 0.0000000000000001,
    (71, 51, 3): 2.7777,
    (0, 0, 0): 2.7777,
    (205, 0, 101): 0.0000000000000001,
}
```

Winter

```
speedMap = {
    (248, 148, 18): 2.7777,
    (255, 192, 0): 1.38,
    (255, 255, 255): 2.7777,
    (2, 208, 60): 1.9444,
    (2, 136, 40): 1.38,
    (5, 73, 24): 0.0000000000000001,
    (0, 0, 255): 0.0000000000000001,
    (71, 51, 3): 3,
    (0, 0, 0): 2.7777,
    (205, 0, 101): 0.0000000000000001,
    (110, 255, 255): 1.38 // ICE speed
}
```

Fall

```
speedMap = {
    (248, 148, 18): 2.7777,
    (255, 192, 0): 1.38,
    (255, 255, 255): 2.0,
    (2, 208, 60): 1.1,
    (2, 136, 40): 1.1,
    (5, 73, 24): 0.0000000000000001,
    (0, 0, 255): 0.0000000000000001,
    (71, 51, 3): 2.7777,
    (0, 0, 0): 2.7777,
    (205, 0, 101): 0.0000000000000001,
}
```

Spring

```
speedMap = {
    (248, 148, 18): 2.7777,
    (255, 192, 0): 1.38,
    (255, 255, 255): 2.7777,
    (2, 208, 60): 1.9444,
    (2, 136, 40): 1.38,
    (5, 73, 24): 0.0000000000000001,
    (0, 0, 255): 0.0000000000000001,
    (71, 51, 3): 2.7777,
    (0, 0, 0): 2.777,
    (205, 0, 101): 0.0000000000000001,
    (133, 87, 35):1.38 // MUD Speed

}
```

The above speeds are Mentioned in Meter/Sec

Algorithms for different seasons

Algorithm for Summer:
It is a basic A* search is applied from source to destination. Considering the heuristics as mentioned above.

Algorithm for Fall:
As mentioned in the LAB it gets difficult to pass from the forests during Fall. Thus the terrains where the forests are involved have been assigned with a reduced speed.

Algorithm for Winter:
Initially the edges of the lakes are calculated , then those are frozen. After that we go deeper into the lake layer by layer to freeze the layers. This is done 7 times. The layers are calculated from the prior knowledge of the edges.
Then the speed for ICE is introduced and the A* algorithm is applied.

Algorithm for Spring:
Initially the shores of all the lakes are calculated.
The pixels in these shores are checked for the height. If the difference between the water height and the pixel height is less than 1m then that pixel is converted into mud. This pixel will now also have its own characteristic that will contain the value of the pixel due to which it is converted into mud.
Now using the information obtained in this layer we will go to the next layer and check whether the water can reach this new layer or not.
If the difference between the level of water stored in the characteristics of the earlier and the new pixel is less than 1 then this new pixel is also converted into mud. This is done until we reach 15 pixels from the shore.

After the correct pixels are converted to mud we run the A* algorithm.

Description of the Output:
The path generated from the A* algorithm is colored red and the checkpoints are colored purple. The neighbors of the checkpoints are also colored so that the checkpoints are properly visible.