



Elena: elevation-based navigation

Yu-Chuan Chen

Huan Wang

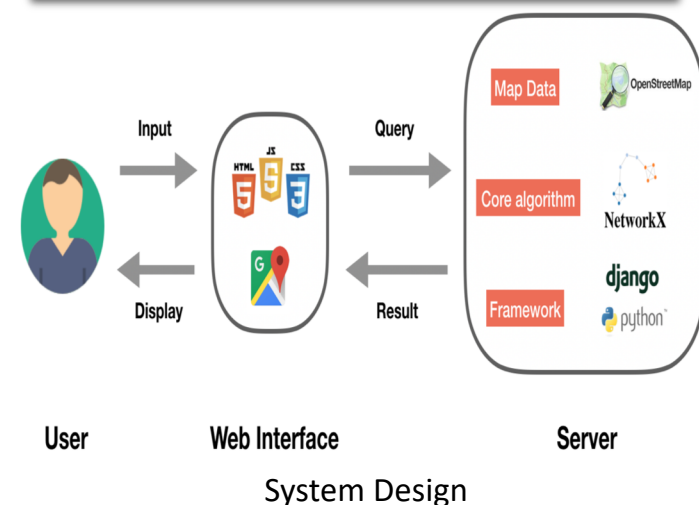
Chenmei Li

Ruei-Yao Sun

Introduction

Build a software that takes 2 places as input, output the maximum/minimum elevation gain within an user-defined x% of the shortest path.

Methodology



More Details

- Google API: Places API, Maps JavaScript API, Geocoding API, Maps Elevation API, Directions API
- Web: HTML/CSS, Bootstrap 4, JQuery, JavaScript
- Use GitHub to do collaborative development

Algorithm

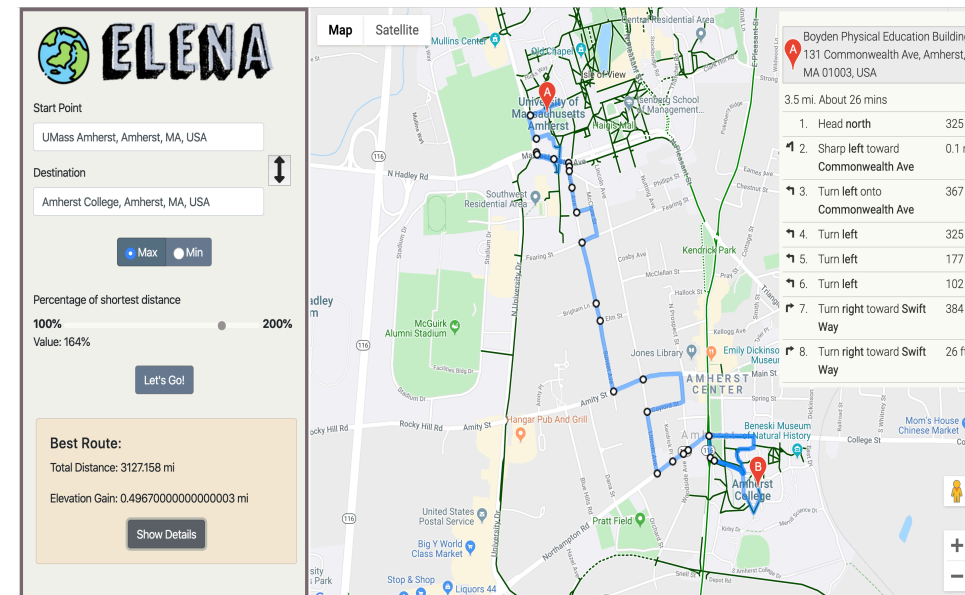
Algorithm 1 Get Minimum Elevation Gain

```

1: function GET_MIN_ELEVATION_GAIN(start, end, r)
2:   path ← shortest_path_algorithm(G)
3:   Calculate min_gain and min_dist in path
4:   max_dist = min_dist * r
5:   Initialize elevation gain (v.g), dist (v.d), parent(v.p)
6:   pq = Priority_Queue(G.V)
7:   while pq is not empty do
8:     u ← pq.pop()
9:     for v in u.neighbor do
10:      if e(u, v) is seen then
11:        continue
12:      end if
13:      Record edge (u, v) is seen
14:      if v.d + d(u, v) > max_dist
15:        or v.g + g(u, v) > min_gain then
16:        continue
17:      end if
18:      Try to minimize v.g and update v.g, v.d, v.p
19:      min_gain = min(min_gain, end.g)
20:      pq.push(v)
21:    end for
22:  end while
23:  Get new path, elevation gain, and distance from G
24:  return new path, elevation gain, and distance
  
```

* For maximum, it can be conducted in a similar way.

Web Interface



Evaluation

Route Comparison

- Umass → Amherst College
- x%=200%



Max elevation gain



Google Map



Min elevation gain