A* Heuristic Calculation Example

This heuristic estimates the difficulty of transitioning between two holds, taking into account the following factors:

1. Distance Between Holds (d):

The distance between two holds plays a major role in determining the difficulty of a move. The Euclidean distance is used:

d = scalar * sqrt((x2 - x1)^2 + (y2 - y1)^2)

A greater distance makes the move harder, so the heuristic scales accordingly.

2. Hold Type (h_t):

Different holds require different amounts of strength and skill to use. Assign difficulty values to each hold type. May be able to tie this in to the climber's account and have them rate their comfortability on each hold type:

- Jug: 0 (easiest)

- Crimp: 0.75 (harder)

- Sloper: 1 (hardest)

- Pinch: 0.75

- Pocket: 0.75

- Undercling: 0.5

h_t_penalty = scalar * hold type

3. Hold Size (h_s):

Blob detection can be used to estimate hold size, with larger holds generally being easier:

h_s _penalty = scalar * (1 / blob_size)

4. Movement Complexity (m_c):

Movements that involve more complex techniques or greater body control require more effort.

Complexities here are on a scale of 0->1 that we can multiply against a scalar for weighting this component:

- Basic Reach: 0 (no penalty)
- Dynamic Movement (Jump): 1

- Crossing Hands/Feet: 0.5

- Heel Hook (Use heel to establish on a hold): 0.6

- Flagging/smearing (Having a foot touching the wall or nothing at all for stability purposes or if no better move exists. Typically only occurs if 3 other solid points of contact exist): 0.8

Penalize moves with greater complexity using:

m_c_penalty = scalar * movement_difficulty

5. Elbow Angle (phi):

Hanging from straight arms is more energy efficient than using bent arms. Penalize based on the degree to which arms are bent: e_penalty = scalar * (180 - phi) where phi is the elbow angle, modeling straight (unbent) arms as 180.

Heuristic Formula:

h(n) = euc_distance

+ h_t_penalty

+ h_t_size_penalty

+ m_c_penalty

+ e_penalty

h(n) =

A * sqrt((x2 - x1)^2 + (y2 - y1)^2)

+ B * (hold_difficulty)

+ C * (1 / blob_size)

+ D * (movement_complexity)

+ E * (180 - phi)

Where A,B,C,D,E are scalars to weight the given components