In our previous meeting we raised the question of how we will sequence moves. Instead of just providing a list of move sequences, we need to additionally assign which limb we want to move.

Additional heuristic value:

Stability: Check distance between hands and feet to ensure moves are done in a sequence where balance can be maintained. Its necessary to make sure no limb is "left behind" because this typically leads to big moves needing to be made for a limb to "catch up". Way of imposing an order of movement.

- Don't want to have to jump feet up the wall to get caught back up with the hands, instead walk them up incrementally.
- Find average distances between hands and feet on the x and the y axis.
- Ensures your limbs are at possible distances while transitioning between moves

Formula:

$$stability\_penalty = stability\_factor \times \sqrt{(hand\_x\_avg - foot\_x\_avg)^2 + (hand\_y\_avg - foot\_y\_avg)^2}$$

Calculation:

1. Average horizontal position of the hands ($hand\_x\_avg$):

$$hand\_x\_avg = \frac{left\_hand\_x + right\_hand\_x}{2}$$

2. Average vertical position of the hands ($hand\_y\_avg$):

$$hand\_y\_avg = \frac{left\_hand\_y + right\_hand\_y}{2}$$

3. Average horizontal position of the feet ($foot\_x\_avg$):

$$foot\_x\_avg = \frac{left\_foot\_x + right\_foot\_x}{2}$$

4. Average vertical position of the feet ($foot\_y\_avg$):

$$foot\_y\_avg = \frac{left\_foot\_y + right\_foot\_y}{2}$$

Problem: This could lead to a situation where the climber is scrunched up, which is also suboptimal.

Solution: Add a preferred distance on the y axis for hand and foot placement. This value will need to be tested with and will likely being in the 3-4 foot range, but also could be derived from climber's height.

$$stability\_penalty = stability\_factor \times \sqrt{(hand\_x\_avg - foot\_x\_avg)^2 + (|hand\_y\_avg - foot\_y\_avg| - d_{preferred})^2}$$

Potentially break up the x and y factors:

Formula:

$$stability\_penalty = stability\_penalty\_x + stability\_penalty\_y$$

Where:

1. **Horizontal Stability Penalty** ($stability\_penalty\_x$):

   $$stability\_penalty\_x = stability\_factor\_x \times |hand\_x\_avg - foot\_x\_avg|$$

2. **Vertical Stability Penalty** ($stability\_penalty\_y$):

   $$stability\_penalty\_y = stability\_factor\_y \times ||hand\_y\_avg - foot\_y\_avg| - d_{preferred}|$$

Basic Algorithm Setup: Store the current state as a list of tuples for each relevant limb.

state =

("LeftHand", HoldA), # Left hand is on HoldA

("RightHand", HoldB), # Right hand is on HoldB

("LeftFoot", HoldC), # Left foot is on HoldC

("RightFoot", HoldD) # Right foot is on HoldD

]

```
state = [
    ("LeftHand", (x1, y1)),   # Left hand is on hold at (x1, y1)
    ("RightHand", (x2, y2)),  # Right hand is on hold at (x2, y2)
    ("LeftFoot", (x3, y3)),   # Left foot is on hold at (x3, y3)
    ("RightFoot", (x4, y4))   # Right foot is on hold at (x4, y4)
]
```

Moving to a new state: Run a search for each limb and generate the optimal move value for each limb. Compare the limbs and select the one with the lowest cost as the limb to move along with the corresponding action.