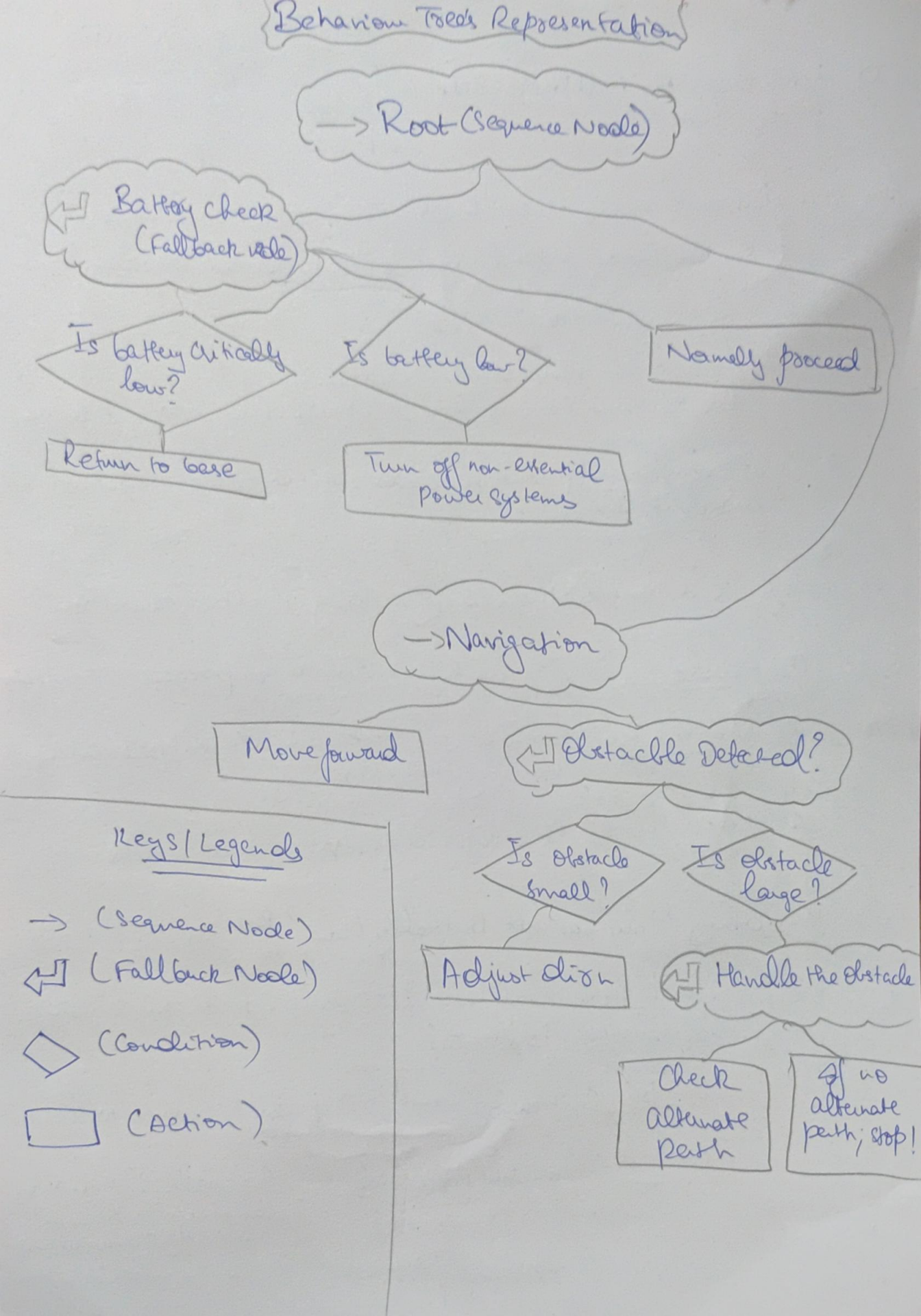
Behaviour Trees Representation



Srinath S

(a) How does the Fallback node help in making better decisions?

Ans: ⊙ A fallback node tries child nodes until one succeeds; i.e in the sense acts like a priority based decision.

⊙ Ex: Battery check;
↳ Here the rover checks critical case first
↳ If above case is false it checks low battery & so on.

⊙ This ensures efficient handling of edge cases without redundant checks.

(b) Why is it better than using a long if-else condition?

Ans: ⊙ Readability: Flowcharts are easy to comprehend than codes. Provides ease in debugging

⊙ Flexibility: Nodes can be rearranged/swapped without rewriting the entire logic.

⊙ Modularity: & breaks logic into reusable nodes

(c) Handling "Low but not Critically Low" Battery

Ans: • The fallback node evaluates its childs until one succeeds.
so Process is as follows;

↳ Critically low? → Fails & moves to next child
↳ Low but not critial → Succeeds & turns off Cameras & runs in reduced power usage.