Basic Robot Behavior Tree Simulation

Nishanth Yeniebera

07/31/2025

Zeid Kootbally

Python 3.10.12

**Modules Used:** py\_trees[[1]](#footnote-19147), time[[2]](#footnote-17331), logging[[3]](#footnote-24020)

**Objective:**

Create a behavior tree that simulates (text-only in terminal) 1 robot moving a single part from a “part tray” to a “kit tray.” The goal is for the robot to move 3 parts from the “part tray” to the “kit tray.” Permitted actions and conditions are as listed:

1. Move robot towards part tray
2. Move robot towards kit tray
3. Pick up a part
4. Put down a part

**Project File Structure:**

*root/*

create\_tree.py: Tree is created here

main.py: Execution file for created behavior tree

**behaviors/**

Actions.py: Creates custom action node objects (*Pickup, PutDown)*

base.py: Initializes a basic condition *(Condition)* and action *(Action)* node objects. Logging functionality for debug is also set up.

Conditions.py: Creates custom condition node objects (*KitTrayFull, PartTraySufficient)*

Constants.py: Creates shared constants throughout the project

custom\_blackboard.py: Creates a blackboard for project variables

**Behavior Tree Structure:**

(Italicized names are a reference to the node’s “name” attribute)

*rootSequence* (Sequence)

- *sequenceA* (Sequence)

- *PartTraySufficient* (Condition): Checks if “part tray” has enough parts initially.

- *kitTrayFull* (Inverted Condition): Checks if “kit tray” has sufficient space before placing new parts

- *sequenceB* (Sequence)

- *MoveToPartTray* (Action): Robot begins to travel until it reaches the “part tray”

- *PickUpPart* (Action): Robot picks up a part from the tray (assuming it is at a tray)

- *MoveToKitTray* (Action): Robot begins to travel until it reaches the “kit tray”

- *PutPartDown* (Action): Robot sets down a part on the tray (assuming it is actively carrying one)

**Variables:**

Blackboard variables:

* *partTrayQty* (integer): Quantity of parts in “part tray”
* *kitTrayQty* (integer): Quantity of parts in “kit tray”
* *partsMoved* (integer): Quantity of parts moved
* *isSufficient* (boolean): Outcome of a check for “part tray” having the sufficient initial quantity of parts to complete goal

Constants:

* *DEBUG\_PAUSE* (float): Time between debugging logs (and by multiple of 10 is time between ticks)
* *CARRYING\_AMOUNT* (integer): Quantity robot is capable of carrying
* *GOAL\_AMOUNT\_MOVED* (integer): Goal quantity of parts to move
* *TRAY\_MAX\_PARTS* (integer): Maximum quantity of parts a tray may contain

1. <https://py-trees.readthedocs.io/en/devel/> [↑](#footnote-ref-19147)
2. https://docs.python.org/3/library/time.html [↑](#footnote-ref-17331)
3. https://docs.python.org/3/library/logging.html [↑](#footnote-ref-24020)