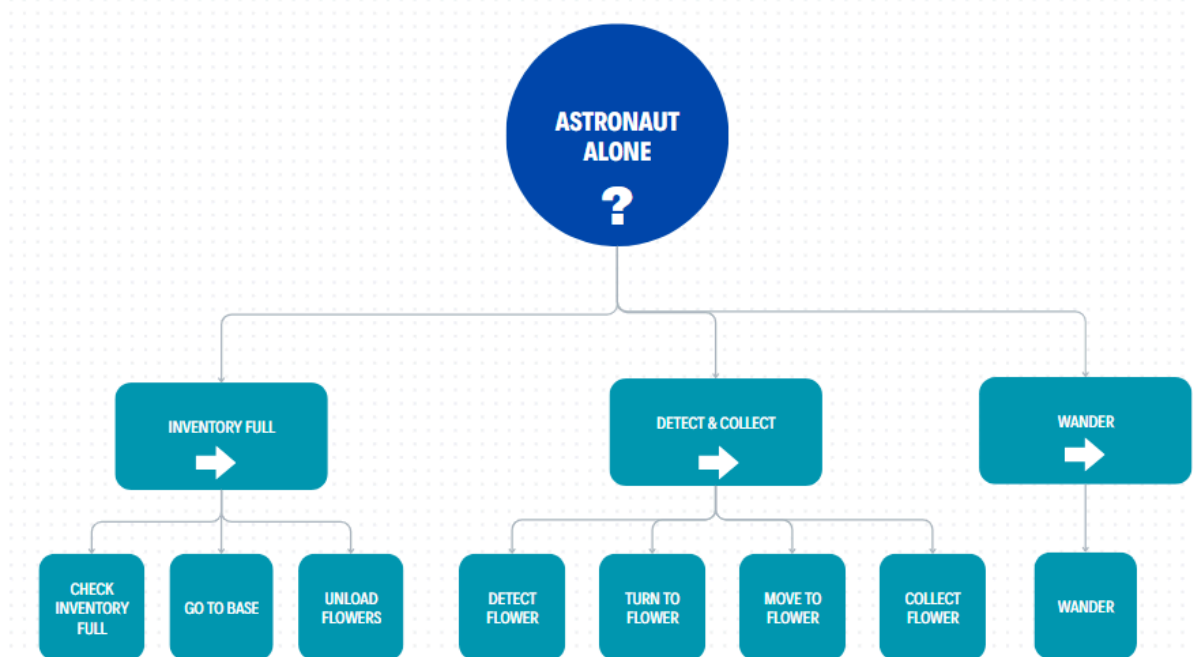


# AAPE REPORT BEHAVIOUR TREE

## ASTRONAUT ALONE

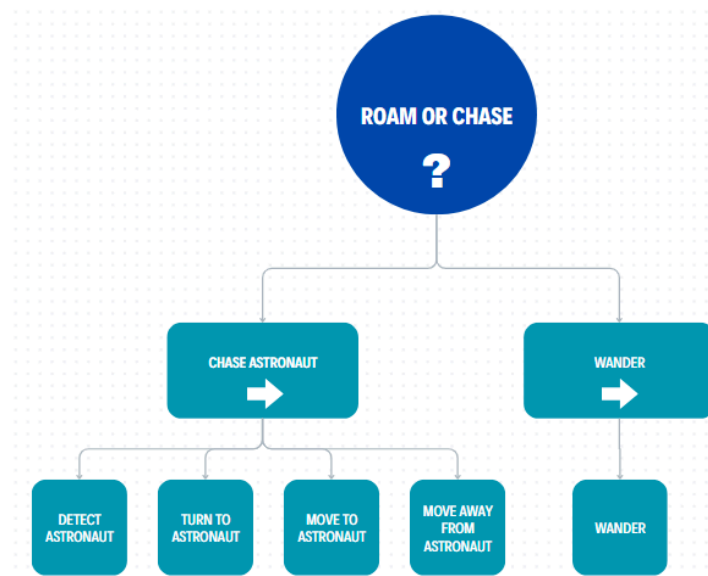


In this scenario, the astronaut is alone in the environment. Its goal is to wander around the environment and recollect flowers. It can only carry two flowers at a time, so it needs to go back to the base to unload the flowers and go back to looking for more flowers to collect.

1. **Inventory full:** aims to check if the astronaut has reached the inventory limit and needs to go to the base and unload the flowers.
  - a. **Check inventory full:** looks for `AlienFlowers` in the inventory, if the amount is equal or bigger than 2 it returns `SUCCESS`.
  - b. **Go to base:** sends the action `'walk_to,Base'` which is already programmed. Returns `SUCCESS` if the astronaut is in the base and it is not `OnRoute`; meaning that it is not moving to a target position
  - c. **Unload flowers:** executes the action `'leave,AlienFlower,2'`. Returns `SUCCESS` if there is no `AlienFlower` in the inventory.
2. **Detect & collect:** aims to look for flowers and collect them
  - a. **Detect flower:** uses the sensor information to look for flowers. Returns `SUCCESS` if some ray detects an `AlienFlower`.
  - b. **Turn to flower:** checks the sensor information again and chooses the direction where the flower is closest. After this, it looks for the angle to turn based on a dictionary and executes the turn using either `tr` or `tl`. Returns `SUCCESS` when it has turned the determined degrees.
  - c. **Move to flower:** checks again the sensor and chooses the closest flower and executes the `ForwardDist` goal with the determined distance.
  - d. **Collect flower:** executes the collect action and returns `SUCCESS` when the action task is finished.

3. **Wander:** executes the Avoid goal, which moves forward while slightly turning to a random direction while avoiding the obstacles in the environment. To allow the astronaut to collect flowers, while it is wandering, it checks for flowers in the sensors. If it detects a flower, it returns SUCCESS to allow the astronaut to execute the `DetectFlower` branch.

## CRITTERS



In the Critters scenario the astronaut is going to be chased by critters. The critters, initially, roam aimlessly avoiding the obstacles but, once a critter detects the astronaut, it attempts to approach her and bite her. A successful bite temporarily stunts the astronaut for 5 seconds and causes the loss of a flower in the inventory, if available. After biting, the critter retreats to give the astronaut space to recover and if it can no longer detect her, it returns to its default roaming behaviour.

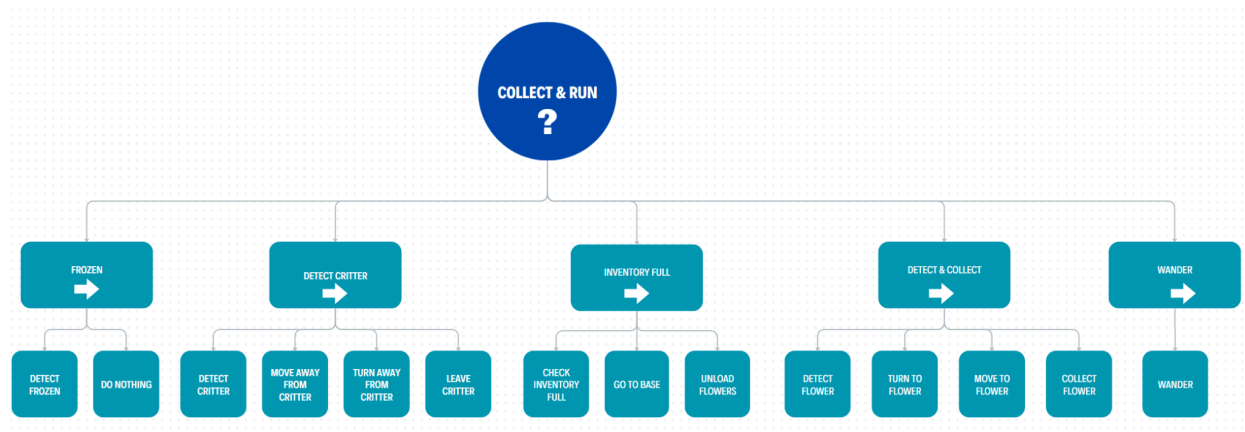
### 1. Chase Astronaut:

- a. **Detect Astronaut:** it uses the sensors to get information from the environment. It will return SUCCESS if the critter's rays detect the `Astronaut`.
- b. **Turn to Astronaut:** As the astronaut keeps moving, it has to be checked actively with the different rays that it is still there and it has not been lost. Therefore the process to detect the astronaut will be done once again.
- c. **Move to Astronaut:** Checks the sensor data and if the distance is less than 0.8 it returns SUCCESS, if not it considers that the astronaut has moved and therefore it has not been bitten (returns FAILURE)
- d. **Move away from Astronaut:** In case the previous node returns SUCCESS, the critter understands it has bitten the astronaut and thus moves away from the astronaut by moving backwards and turning left.

### 2. Wander:

- a. **Wander:** when none of the previous nodes return SUCCESS, the critters wander with the function `AvoidForCritters` which is the same as the avoid but with a bonus. When a critter has another critter detected in the sensor, it will turn left.

## COLLECT AND RUN



In this last scenario the astronaut is required to show the same behaviour as in the alone scenario but actively avoiding the Critters.

If a critter bites the astronaut, this gets frozen for 5 seconds. While the astronaut is frozen it cannot execute any action, that is why the first branch of the behaviour tree checks this condition.

Due to the critter stealing one flower when it bites the astronaut, the astronaut needs to double check the inventory. If the astronaut gets bitten while going to the base, it loses one flower, so the go to base task needs to return false if this happens. Otherwise, the astronaut would go to the base with only one flower.

During wandering, the astronaut needs to check if the sensors detect either flowers or critters, otherwise, the wandering would go on forever.

- **Frozen**
  - **Detect frozen:** checks the internal state of the astronaut and returns SUCCESS if `isFrozen` variable is set to True.
  - **Do nothing:** since the astronaut is not allowed to move while it's frozen, the `DoNothing` goal is executed.
- **Detect critter:**
  - **Detect critter:** checks the sensors and looks for `CritterMantaRay`. It returns SUCCESS if a `CritterMantaRay` is detected.
  - **Move away from critter:** the astronaut needs to escape from the critter. This node executes the `BackwardDist` goal and moves a fixed distance. When this action is finished, the node returns SUCCESS.

- **Turn away from critter:** the astronaut turns a fixed number of degrees. The direction of the turn depends on the obstacles around it; it turns to the less cluttered side.
- **Move away critter:** to better escape the critter, the astronaut moves forward a fixed distance. If the astronaut gets bitten while executing this, it returns FAILURE. If the action is executed successfully, the node returns SUCCESS.
- **Inventory Full:**
  - **Check Inventory Full:** This node has the same behavior as it did in the Alone Scene
  - **Go to Base:** This node has the same behavior as it did in the Alone Scene. If blocked, it will return FAILURE (since the critter will have bitten the astronaut and one flower will be lost)
  - **Unload Flowers:** This node has the same behaviour as it did in the Alone Scene
- **Detect & Collect**
  - **Detect Flower:** This node has the same behaviour as it did in the Alone Scene
  - **Turn to Flower:** This node has the same behaviour as it did in the Alone Scene
  - **Move to Flower:** This node has the same behaviour as it did in the Alone Scene
  - **Collect Flower:** This node has the same behaviour as it did in the Alone Scene
- **Wander:**
  - **Wander:** in order to keep wandering forever, both flowers and critters are going to be searched using the sensors.

## Code execution

For all the scenarios, open AAPE and choose the 3rd scenario.

- **BT Astronaut alone:**
  - In AAgent-1.json, set "initial\_task": "bt:BT Astronaut Alone"
  - `python AAgent_BT.py AAgent-1.json`
- **BT Critters:**
  - In AAgent-1.json, set "initial\_task": "bt:BT Astronaut Alone" or "initial\_task": "bt:BT Collect Run"
  - In AAgent-2.json, set "initial\_task": "bt:BT Roam Or Chase"
  - In APackAstroCritters.json modify the number of critters and astronauts
  - `python Spawners.py APackAstroCritters.json`
- **BT Collect and run:**
  - In AAgent-1.json, set "initial\_task": "bt:BT Collect Run"
  - In APackAstroCritters.json modify the number of critters and astronauts
  - `python Spawners.py APackAstroCritters.json`