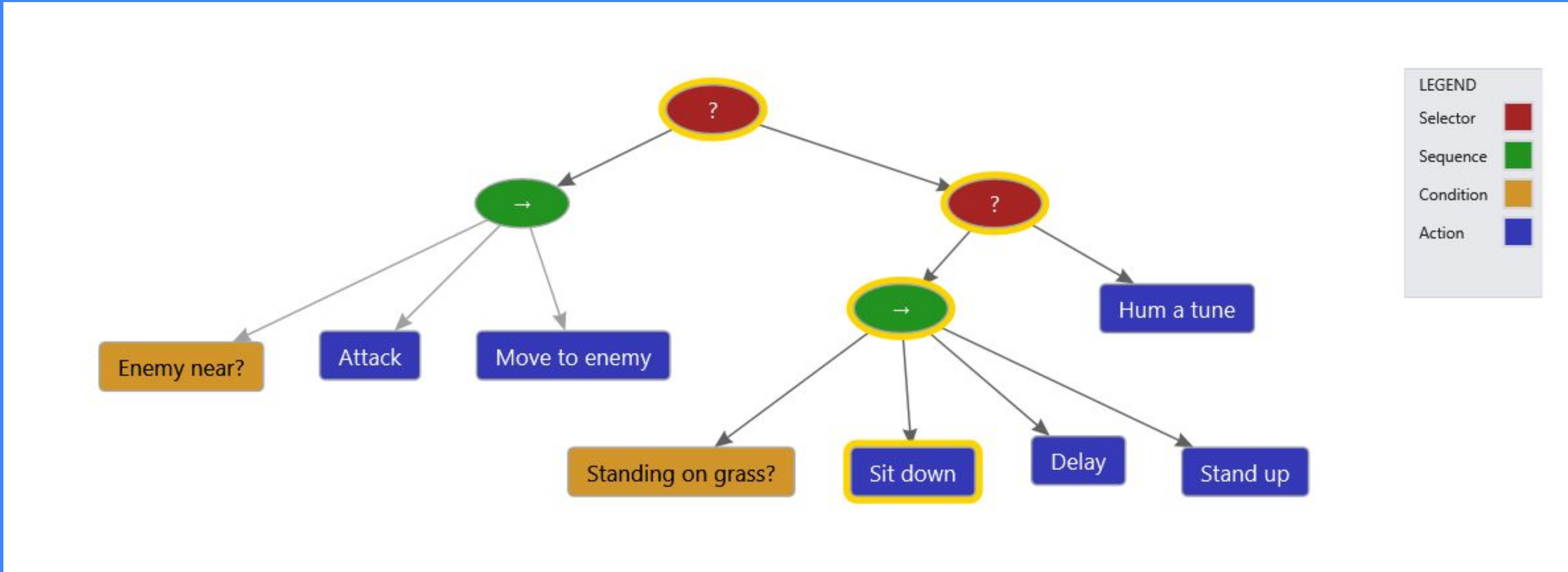
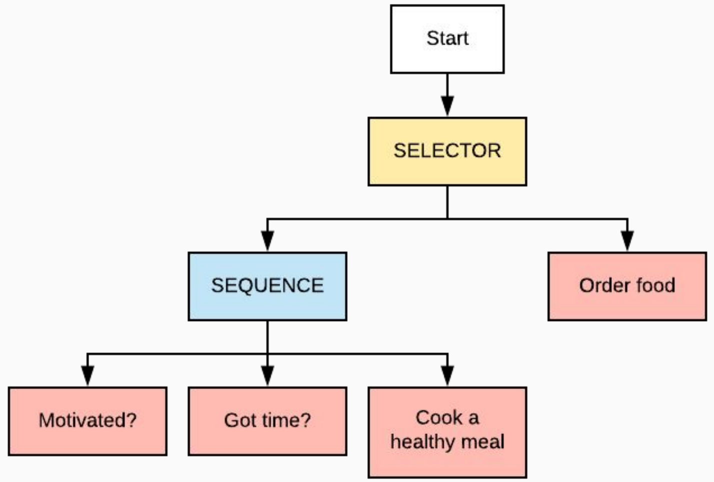
Behaviour Trees And Putting All Together



Behaviour Trees

- Used to compactly define sequential processes.
- Leads to complex behaviour in a easy and understandable way
- Have mostly replaced Finite State Machines(FSM)
- Each node returns either Success, Failure or Running.

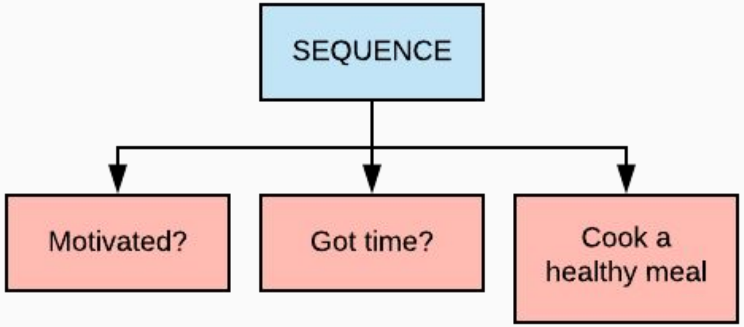


Behaviour Trees, Sequences

- Sequence nodes contain one or more children. Upon execution, it executes every child and fails when one of the children fails.

- Pseudo Code

```
for child in children:
    status = child.run()
    if status == RUNNING or status == FAILURE:
        return status
return SUCCESS
```

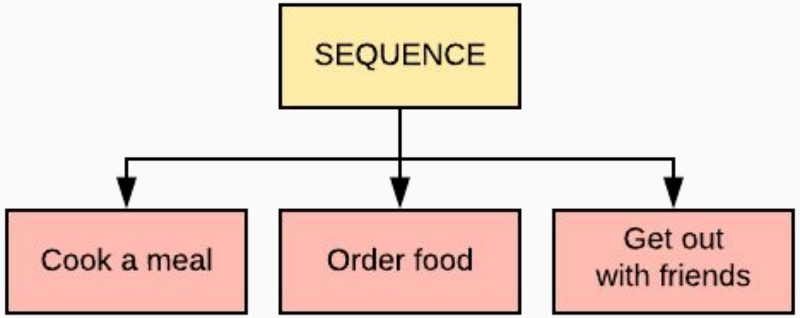


Behaviour Trees, Selector

- Selector nodes contain one or more children. Upon execution, it executes every child until one of them succeeds, otherwise it fails.

- Pseudo Code

```
for child in children:
    status = child.run()
    if status == RUNNING or status == SUCCESS:
        return status
return FAILURE
```



Behaviour Trees, Decorator

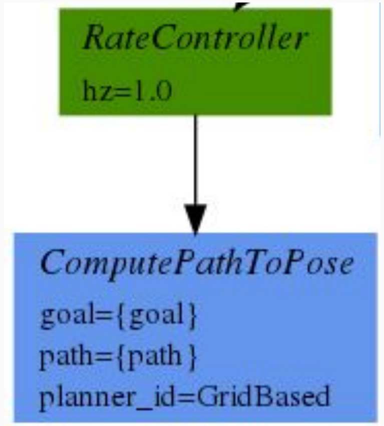
- Decorator nodes can only have a single child. They are mostly used as utility nodes, for example:

  Repeater: Runs the child node indefinitely or a number of times.

  Inverter: Inverts the result of the child node.

  AlwaysSucceed: Failure becomes Success.

  UntilFail: Runs the child node until it fails.

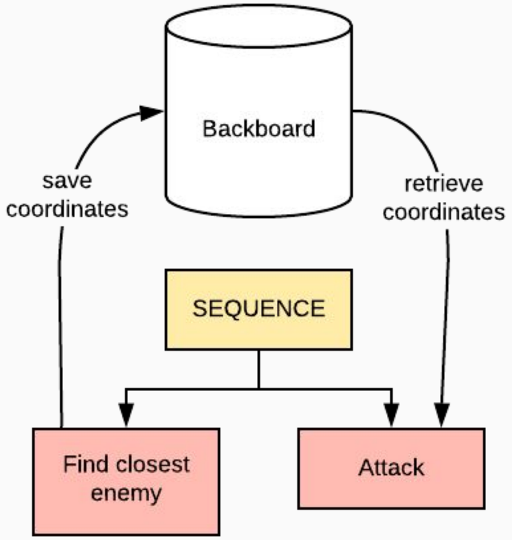


Behaviour Trees, BlackBoards

- some nodes will want to “talk” to other nodes. Behavior Trees can have some kind of data store that is global and accessible from all nodes to do so.

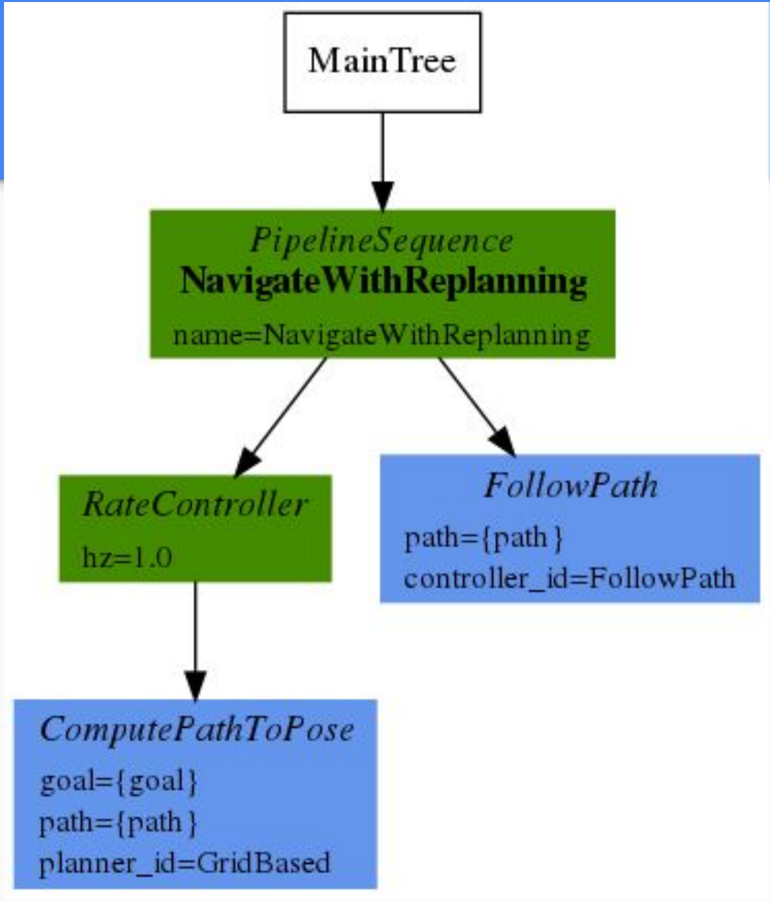
Usage

- **Video games**: To model the behaviors of enemies and non-player characters.
  This is where behavior trees shine.
- **Conversational AI**: I recently read a blog post about building a conversational AI using behavior trees and the result is quite impressive. How to Build an End-to-End Conversational AI System using Behavior Trees
- **Robotics Systems**: These last years behavior trees have also received attention in the robotics domain. If you want to learn more, here’s an interesting paper: Behavior Trees in Robotics and AI: An Introduction.



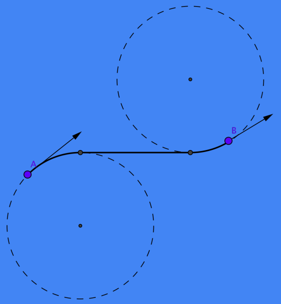
Behaviour Trees, Navigation

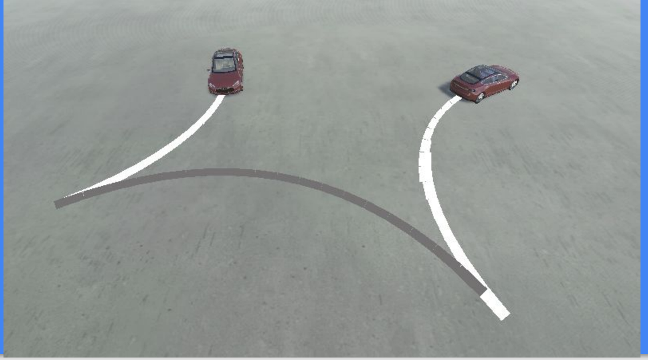
- Used to compactly define sequential processes that Robot needs to do
- Leads to complex behaviour in a easy and understandable way
- Have mostly replaced Finite State Machines(FSM)
- Each node returns either Success, Failure or Running.



Video, Navigation without *ros::navigation*

Issues





- Generation of probabilistic plans sometimes leads to unfeasible, plans SE2, could be replaced with DUBINS or REEDSHEEP spaces

- MPC Controller struggles with following paths when the given goal is at back side

- MPC speed is constant during path following, but in the sharp turns this leads to huge drifts, constant speed might not be feasible for real application

- There are bugs on the planner plugins, will need to crack some of them. There are certain limitation in MPC controller but when planner generates a decent plan, controller actually does good job.