

**Link:** [**https://editor.plantuml.com**](https://editor.plantuml.com)

Link aufrufen, Code rauskopieren, png image anklicken, speichern/download

@startuml

' Titel für das Diagramm

title Architektur der Multi-Agenten-Simulation zur autonomen Exploration

' Layout-Richtung von links nach rechts festlegen

left to right direction

' --- Hauptmodell und Umgebungs-Agenten ---

package Model {

class Exploration {

+grid : OrthogonalMooreGrid

+pubSubBroker : PubSubBroker

+datacollector: DataCollector

+step()

}

class Ground {

+explored: boolean

}

class Obstacle {

+cell\_blocking: boolean

+view\_blocking: boolean

}

class PubSubBroker {

+subscribe()

+publish()

}

}

Exploration "1" \*-- "1" PubSubBroker : hat

Exploration "1" \*-- "\*" Ground : erzeugt

Exploration "1" \*-- "\*" Obstacle : erzeugt

Exploration "1" \*-- "\*" Agents.ExplorerRobot : erzeugt

' --- Roboter-Agenten und Speicher ---

package Agents {

abstract class ExplorerRobot {

#local\_memory : LocalMemory

+view\_radius: int

+orientation: int

+step()

+scan\_environment()

}

class FBERobot {

-goal: tuple

-path: list

+step()

}

class RandomWalkRobot {

+step()

}

class LocalMemory {

-grid\_info: dict

-frontier\_info: dict

+get\_known\_neighbor\_positions()

}

class CellInfo {

+agents: list<AgentInfo>

}

class AgentInfo{

+unique\_id: int

}

class FrontierInfo {

-status: FrontierStatus

}

class FrontierStatus <<enumeration>> {

IN\_WORK

OPEN

}

}

Agents.ExplorerRobot <|-- Agents.FBERobot

Agents.ExplorerRobot <|-- Agents.RandomWalkRobot

Agents.ExplorerRobot "1" \*-- "1" Agents.LocalMemory : hat

Agents.LocalMemory "1" \*-- "\*" Agents.CellInfo : enthält

Agents.CellInfo "1" \*-- "\*" Agents.AgentInfo : enthält

Agents.LocalMemory "1" \*-- "\*" Agents.FrontierInfo : enthält

Agents.FrontierInfo ..> Agents.FrontierStatus

Agents.FBERobot ..> Model.PubSubBroker : verwendet

' --- Algorithmus-Strategiemuster ---

package Algorithms {

interface Pathfinder {

+find\_path()

}

class AStar {

+find\_path()

}

class Node {

+pos: tuple

+cost: float

}

interface MovementGoalFinder {

+find\_goals()

}

class OriginalFBE <<Inferred from Factory>> {

+find\_goals()

}

interface MovementGoalSelector {

+select\_goal()

}

class NearestBiggestFrontier <<Inferred from Factory>> {

+select\_goal()

}

}

Algorithms.Pathfinder <|.. Algorithms.AStar

Algorithms.MovementGoalFinder <|.. Algorithms.OriginalFBE

Algorithms.MovementGoalSelector <|.. Algorithms.NearestBiggestFrontier

Algorithms.AStar ..> Algorithms.Node : verwendet

' --- Factories zur Erzeugung von Algorithmen ---

package Factories {

class PathfinderFactory {

+give\_pathfinder()

}

class MovementGoalFinderFactory {

+give\_movement\_goal\_finder()

}

class MovementGoalSelectorFactory {

+give\_movement\_goal\_selector()

}

class PathfinderEnum <<enumeration>> {

ASTAR

}

class MovementGoalFinderEnum <<enumeration>> {

ORIGINAL\_FBE

}

class MovementGoalSelectorEnum <<enumeration>> {

NEAREST\_BIGGEST\_FRONTIER

}

}

Factories.PathfinderFactory ..> Algorithms.AStar : erzeugt

Factories.PathfinderFactory ..> Factories.PathfinderEnum : verwendet

Factories.MovementGoalFinderFactory ..> Algorithms.OriginalFBE : erzeugt

Factories.MovementGoalFinderFactory ..> Factories.MovementGoalFinderEnum : verwendet

Factories.MovementGoalSelectorFactory ..> Algorithms.NearestBiggestFrontier : erzeugt

Factories.MovementGoalSelectorFactory ..> Factories.MovementGoalSelectorEnum : verwendet

' --- Verbindungen zwischen Agenten und Algorithmen ---

Agents.FBERobot "1" \*-- "1" Algorithms.Pathfinder : hat <<Strategie>>

Agents.FBERobot "1" \*-- "1" Algorithms.MovementGoalFinder : hat <<Strategie>>

Agents.FBERobot "1" \*-- "1" Algorithms.MovementGoalSelector : hat <<Strategie>>

Agents.FBERobot ..> Factories.PathfinderFactory : verwendet

Agents.FBERobot ..> Factories.MovementGoalFinderFactory : verwendet

Agents.FBERobot ..> Factories.MovementGoalSelectorFactory : verwendet

@enduml