# CollectorRos

- frameld: string
- moveBaseClient: SimpleActionClient<MoveBaseAction>
- collector: Collector
- + Collector(collectionRange: double)
- + MoveBaseGoal createMoveBaseGoal(marker: AlvarMarker, frameId: string)
- + pair<double, double> translatePointToOrigin(x: double, y: double)



## Collector

- collectionRange: double
- + Collector(collectionRange: double)
- + MoveBaseGoal createMoveBaseGoal(marker: AlvarMarker, frameId: string)
- + pair<double, double> translatePointToOrigin(x: double, y: double)

## walkRobotRos

- n: NodeHandle
- sensorLaser: Subscriber
- velociy: Publisher
- + walk: walkRobot
- + walkRobotRos()
- + void laserSensorCallback(scanMsg: LaserScan)
- + void robotWalkRos()



## walkRobot

- + obstacle: bool
- + msg: Twist
- + action: int
- + walkRobot()
- + void laserSensorCallback(scanMsg: LaserScan)
- + Twist robotWalk()

# randCoord

- + width\_: float
- + length\_: float
- + centerPoint: vector<float>
- + float randX()
- + float randY()

#### **DetectorRos**

- nh: NodeHandle
- arTrackSub: Subscriber
- detector: Detector
- + DetectorRos(validMarkerIds: vector<string>)
- + void detect(msg: AlvarMarker)
- + bool isMarkerDetected()
- + AlvarMarker getClosestMarker()



#### **Detector**

- validMarkerIds: vector<string>
- markers: vector<AlvarMarker>
- + Detector(validMarkerIds: vector<string>)
- + void detect(msg: AlvarMarker)
- + bool isMarkerDetected()
- + AlvarMarker getClosestMarker()
- + bool isMarkerIdValid(markerId: string)

## modelHandlingRos

- + rosPackage\_: string
- + path: string
- +model: SpawnModel
- + deleteMsg: pair<DeleteModelRequest, DeleteModelResponse>
- handling: modelHandling
- n: NodeHandle
- spawnClient: ServiceClient
- deleteClient: ServiceClient
- world: Subscriber
- models: vector<Pose>
- + modelHandlingRos(rosPackage: string)
- + void addModels(id: string, xPos: float, yPos: float, zPos: float)
- + void removeModels(id: string)
- + void randModels(modelNames: vector<string>, width: float, length
- + ServiceClient getSpawnClient()
- + ServiceClient getDeleteClient()



## modelHandling

- + ModelHandling()
- + SpawnModel getModel(path: string, id: string, xPos: float, yPos:
- + pair<DeleteModelRequest, DeleteModelResponse> getDeleteMs