**List of types and functions:**

* States: NeedleState
* Actions: ????
* MDP: Needle
* State space:

StateSpace

POMDPs.states,

POMDPs.iterator(space::StateSpace),

POMDPs.rand(rng::AbstractRNG, space::StateSpace, s::NeedleState)

* Action space:

POMDPs.actions,

ActionSpace

POMDPs.iterator(space::ActionSpace),

POMDPs.rand (rng::AbstractRNG, space::ActionSpace, a::Symbol)

POMDPs.rand(rng::AbstractRNG, space::ActionSpace)

POMDPs.create\_state(mdp::Needle)

POMDPs.create\_action(mdp::Needle)

* Transition distribution:

NeedleDistribution

POMDPs.create\_transition\_distribution(mdp::Needle)

POMDPs.iterator(d::NeedleDistribution)

POMDPs.pdf(d::NeedleDistribution, s::NeedleState)

POMDPs.rand(rng::AbstractRNG, d::NeedleDistribution, s::NeedleState)

* Transition model:

inbounds(mdp:: Needle,x::Int64,y::Int64)

inbounds(mdp:: Needle,state:: NeedleState)

fill\_probability!(p::Vector{Float64}, val::Float64, index::Int64)

POMDPs.transition(mdp::Needle,state::NeedleState,action::Symbol,d::NeedleDistribution=create\_transition\_distribution(mdp))

* Reward model:

POMDPs.reward(mdp::Needle, state::NeedleState, action::Symbol, statep::NeedleState)

* Other functions:

Base.copy!

Base.hash: ?

Base.isequal

POMDPs.n\_states(mdp:: Needle)

POMDPs.n\_actions(mdp:: Needle)

POMDPs.discount(mdp::Needle)

POMDPs.state\_index(mdp::Needle, state::NeedleState)

POMDPs.isterminal(mdp::Needle, s::NeedleState)