- **Class Main.Node**
  Parameters (State s , Node parent ,list children, integer priority, integer cost, integer depth)

  Function expand calls different actions on the current state s and returns a list of children

- **Class Main.Agent**
  Parameters(State initial State, State initial Node, integer maximum capacity, Strategy search strategy, queue of nodes, hashset of states for visited stated, array of states for goal path states)

  Agent Class constructor accept (String grid, String strategy name , Boolean visualize)

  Function generateInitState it set the value of the parameter of initState and initNode using String grid

  Function search.
  While(true)
  If queue nodes empty
  return fail
  front = the node in the front of the queue
  If front has a goal state return the node
  Children = expand(front)
  Strategy.setPriorityToNodes(Children)
  QING(children)

Function QING takes a list of nodes as a parameters. It adds the nodes to the queue if the node has no repeated state.

- **Class Main.State**
  Parameters(Element[][] grid , integer capacity, Action performed action, integer xAgent, Integer yAgent, Integer number of alive passengers , integer number of retrieved boxes,)

  Function getcost. Returns the cost of the state
  The cost = number of damaged boxes + (number of dead passengers * m * n)

  Function isGoal, returns if the state goal state or not
  The state goal state when there is no ships nor wrecks

Class actions.Action

Abstract class with two abstract functions.

Abstract function ActionOnState(State s) return the successor state of state s after this action

IsValid checks if the action valid one or not

Class Down subclass of Action

Function ActionOnState(State s) return the successor of state s after moving the agent location on step down if it is Valid (isValid methed).

Class Up subclass of Action

Function ActionOnState(State s) return the successor of state s after moving the agent location on step up if it is Valid (isValid methed).

Class Left subclass of Action

Function ActionOnState(State s) return the successor of state s after moving the agent location on step Left if it is Valid (isValid methed).

Class Right subclass of Action

Function ActionOnState(State s) return the successor of state s after moving the agent location on step down if it is Valid (isValid methed).

Class Drop subclass of Action

Function ActionOnState(State s) return the successor of state s after droping passengers with the agent in the station if it is Valid (isValid methed).

Class Pickup subclass of Action

Function ActionOnState(State s) return the successor of state s after picking up passengers from the ship , agent stands on it. if it is Valid (isValid methed).

Class retrieve subclass of Action

Function ActionOnState(State s) return the successor of state s after retrieveing the black box from the wreck , agent stands on it. if it is Valid (isValid methed).

Class elements.Element

Abstarct class with one main function

PerformAction(action a) return the new element after performing the action a on this element

Class Ship subclass of Element

Function PerformAction(action a) return a the successor ship after performing the action a if a not pickup action then the new one same as current one but with one dead passenger

Class Wreck subclass of Element

Function PerformAction(action a) return a the successor wreck after performing the action a if a not retrieve action then the new one same as current one but with one additional damage.

Class Station subclass of Element

Function PerformAction(action a) return a copy of this station.

Class strategies.Strategy

Abstract class with one main Function

Function QPriority(List Node) return new list of nodes after adding priorities to given nodes according to the strategy

Class BreadthFirstSearch subclass of Strategy

Class DepthFirstSearch subclass of Strategy

Class GreedySearch subclass of Startegy

This class takes an additional parameters (1 or 2) to specify with heuristic to use

Class AStarSearch subclass of Strategy

This class takes an additional parameters (1 or 2) to specify with heuristic to use

heuristic functions

heuristic 1 : assume all ships in the same cell and agent capacity has no limit. The optimal way is to sort ships according to number of passengers on it in array alive . And pickup all passengers from the largest one then go to next one pickup all passengers -1 (if possible) and the next pickup all passengers -2 and so one.

So the h1 (0 + min(1,alive[len(alive)-2]) + min(2,alive[len(alive)-3). ……)*m*n

H1 = sum(min(I,alive[len(alive)-i+1))*m*n

heuristic 2 : assume all passengers in the one ship

H2 = (numberOfAllAlivePassengers/(agentCapacity + 3)) * 3

3 represent 3 actions have to be done, the agent pickup x passengers and a step to the station and drop. So at least 3 actions

Run with grid "5,5;69;3,3;0,0,0,1,1,0;0,3,78,1,2,2,1,3,14,4,4,9;"

BF RAM = 50M CPU = 7% expanded nodes = 37822

DF RAM = 20M CPU = 5% expanded nodes = 97

ID RAM = 50M CPU = 6% expended nodes = 151417

GR1 RAM = 15M CPU = 5% expanded nodes = 2551

GR2 RAM = 17M CPU = 5% expanded nodes = 3306

AS1 RAM = 20M CPU = 5% expanded nodes = 1674

AS2 RAM = 22M CPU = 5% expanded nodes = 1642