Event based control
Program description

Biernat Damian 226 309
Brzozowy Martin 226 339
Maliszewski Marcin 227 347
Mróz Sebastian 218 559
Sil Aleksander 218 576

October 2019

1 Program description

1.1 Classes

1.1.1 AgvRobot

Class that defines AGV robot.

Class fields

id	identification number of the robot [e.g. AGV001]
node	node (state) on the state graph
path	list of subsequent states that robot has to go throu

Methods

Name	Arguments	Return	Description
makeRequest		Request	Send move permission request to the main controller

1.1.2 AgvRobotsList

Class fields

agvRobots	list of AGV robots
agvCount	number of robots on the list

Methods

Name	Arguments	Return	Description
addAgvRobot	AgvRobot		Adds robot on the list

1.1.3 AgvRobotFactory

Class fields

agvCount	number of robots created so far

Methods

Name	Arguments	Return	Description
makeAgvRobot	initState	agvRobot	Creates an AgvRobot object
makeAgvRobotsList	robotsNumber, initState	agvRobotsList	Creates a list object with added few robots according to robotsNumber argument

1.1.4 MasterController

Class fields

mapGraph	directed graph of states and cost of transition between them
agvList	handler to list of AGV robots

Methods

Name	Arguments	Return	Description
loadMap			Loads map from .csv file with list of edges definition (pairs of 2 nodes and weight)
makeAgvRobotsList	robotsNumber, initState	agvRobotsList	opis
assignPath	agvRobot, task	path	Calculate path in order to let AGV robot finish the given task
plotGraph	figure	figure, plot	Plot map to given figure and return handler to the plot and figure
highlightPath		plot, path, rgbColor	Highlight path on the graph plot

1.1.5 Task

1.1.6 Request

1.1.7 Ship

1.1.8 Storage

1.2 Main loop

In the main loop there are 3 main parts:

Request making The robots ask for a permission to move along their path

Request evaluation The main controller gets the requests, evaluate them and make responses. If robot doesn't have a path then master controller determines a new task and send new path to the robot as well as a permission to move along it.

Robots action Robots act according to the response from master controller

1.3 Assumptions

- Each move between nodes takes 1 time unit - 1 main loop cycle
- When a robot moves to storage/ship it needs to spend 1 interval to unload/load cargo
- Only 1 robot can be at the crossroad at the same time

1.4 Details

Our main goal is to determine the tasks for the robots to perform and to make a controller which will let robots perform these tasks. Robots will receive a task with certain path to complete.

Master controller will determine tasks based on amount of cargo in the ship compartments. The task dispatching process will look for a robot without a task and determine a task for it. Closest (shortest path on a directed graph), non-empty storage will be determined for a robot to go to.

Edges of the map graph will have weights correlated to the number of robots that have to pass this edge. This means that if robot gets a path, program will increase weight of edges on this path. When the robot crosses certain edge then the weight of this edge is decreased.

On 2nd step of main loop the main controller will detect collisions and will determine which robot will pass the crossroad. The robot that was not allowed to enter the node will have a variable corresponding to time the robot had to wait. Master controller will prioritize robots which waited for a longer time.