

Contents

1	Class Architecture	1
1.1	Class Descriptions	1
1.1.1	Introduction	1
1.1.2	CommandCenter	1

Chapter 1

Class Architecture

1.1 Class Descriptions

1.1.1 Introduction

In this section we will provide a detailed introduction of the classes of our project. We describe the attributes and the methods that each class has. We are not describing the setter or getter methods because it is straightforward - they can be figured out by their name and variables. Also here we describe the test- cases for testing setter and getter methods and for all other methods in particular for each class.

1.1.2 CommandCenter

This class manages all the separate components that are required to communicate with the server and to decide a course of action for the Rover. Taking into consideration that there will be only Command Center, we recommend that this class respects the Singleton Pattern. The CommandCenter also acts as a mediator between the different components and in essence runs the entire application.

Attributes

Name	Type	Description
brain	Brain	This is the brain of the application. It is responsible for deciding the course of action using the provided brain file.
communication	CommunicationCenter	This component handles the bidirectional communication between the server and the application.
map	Map	The map that we are currently using to navigate.
logger	Logger	The logger is used for debugging and logging important events that occur during the lifetime of the application.

Operations

CommandCenter()	
Input	None
Output	None
Description	Basic constructor for the CommandCenter class. If we implement the Singleton pattern this should be made private.

CommandCenter* getInstance(int port, string address, string brainLoc)	
Input	The port and address of the server and the location of the brainFile.
Output	None
Description	This method is used to get the instance of the CommandCenter.

CommandCenter()	
Input	None
Output	None
Description	Destructor used for freeing up resources used by the CommandCenter.

Brain* getBrain()	
Input	void
Output	Pointer to the Brain object.
Description	Pointer to the Brain object.

CommunicationCenter* getCommunication()	
Input	void
Output	Pointer to the CommunicationCenter object.
Description	Pointer to the CommunicationCenter object.

Map* getMap()	
Input	void
Output	Pointer to the Map object.
Description	Pointer to the Map object.

Logger* getLogger()	
Input	void
Output	Pointer to the Logger object.
Description	Pointer to the Logger object.

void setLogger(Logger log)	
Input	Logger object.
Output	void
Description	Set the logger to be used by the CommandCenter.

void setBrain(Brain brain)	
Input	Brain object.
Output	void
Description	Set the Brain to be used by the CommandCenter.

void setCommunicationCenter(CommunicationCenter comm)	
Input	CommunicationCenter object.
Output	void
Description	Set the CommunicationCenter to be used by the CommandCenter.