VREP

Description	This is the class constructor and it is responsible for create all the Properties and Methods for the class.
synopsis	**Object Class:** V = VREP;
Parameters	**Library Connection**: RemApi (.dll file makes the bridge with Matlab and V-Rep) **parameterID**: ClientID (ID connection between Matlab and V-REP) **parameter**: Pioneer object handle (handle numbers). **parameter**: Objects Any other object you want to handle.
Functions	vConnect; vDisconnect; vGetLaserData; vGetSensorData; vHandle; vObject; vGetObjPosition; vSendControlSignals;

Example	`%% Create the constructor Class`
	
	`V = VREP;`

vConnect

Description	This is responsible to connect Matlab and V-Rep simulator.
synopsis	**Method:** vConnect(vrep);
Parameters	**parameterID**: ClientID (ID connection between Matlab and V-REP)
return	vrep.clientID (usually it is 1)

Example	`%% Create the constructor Class`
	`V = VREP;`
	
	`%% Connect Matlab to V-Rep`
	`V.vConnect;`

vDisconnect

Description	This is responsible to disconnect Matlab and V-Rep simulator.
synopsis	**Method:** vDisconnect(vrep);
Parameters	**parameterID**: ClientID (ID connection between Matlab and V-REP)
return	**Int**: vrep.clientID (usually it is 0)

Example	`%% Create the constructor Class`
	`V = VREP;`
	
	`%% Connect Matlab to V-Rep`
	`V.vConnect;`
	
	`%% Disconnect Matlab to V-Rep`
	`V.vDisconnect;`

vHandle

Description	This is responsible to get the Pioneer tags on the scene.
synopsis	**Method:** vHandle(String, Index);
Parameters	**String:** The name of the Pioneer you want to control on the V-Rep scene. **Index:** If there is more than one robot, it's necessary to specify the index robot number.
return	**Array:** vrep.Pioneer(1:19), 1-Pionner3DX, 2- Left Motor, 3- Right Motor, 4:19-Ultrasonic sensor.
| Example | ```matlab
%% Create the constructor Class
V = VREP;

%% Connect Matlab to V-Rep
V.vConnect;

%% Get handle tags from Pioneer_p3dx
V.vHandle('Pioneer_p3dx');

%% Get handle tags from Pioneer_p3dx#1
V.vHandle('Pioneer_p3dx','1');

%% Get handle tags from Pioneer_p3dx#2
V.vHandle('Pioneer_p3dx','2');
``` |

vGetSensorData

Description	This is responsible to get the position and velocity from Pioneer.
synopsis	**Method:** vGetSensorData(Index);
Parameters	**Index:** Usually it is 1, if there is more than one robot, it's necessary to specify the index robot number.
return	**Array:** Xc(1:3), it returns the cartesian coordinates x, y, and z from the robot center; **Array:** X(1:3), it returns the cartesian coordinates x, y, and z from the robot control point; **Array:** U(1:2), it returns the linear and angular velocity from the robot;
| Example | ```matlab
%% Create the constructor Class
V = VREP;

%% Connect Matlab to V-Rep
V.vConnect;

%% Get handle tags from Pioneer_p3dx
V.vHandle('Pioneer_p3dx');

%% Get handle tags from Pioneer_p3dx#1
V.vHandle('Pioneer_p3dx','1');

%% Request position and velocity data from Pioneer_p3dx
[Xc1,X1,U1] = V.vGetSensorData(1);
%% Request position and velocity data from Pioneer_p3dx#1
[Xc2,X2,U2] = V.vGetSensorData(2);
``` |

vObject

Description	This is responsible to get the position and velocity from any other object on the scene.
synopsis	**Method:** vObject(String);
Parameters	**String:** The object's name you want to control on the V-Rep scene.
return	**Int:** vrep.Object{ }, with the object tag.
| Example | ```
%% Create the constructor Class
V = VREP;

%% Connect Matlab to V-Rep
V.vConnect;

%% Get handle tags from object Disc
V.vObject('Disc');
``` |

vGetObjPosition

Description	This is responsible to get the position and orientation from any other object on the scene.
synopsis	**Method:** vGetObjPosition(String);
Parameters	**String:** The object's name you want to request data on the V-Rep scene.
return	**Array:** Xc(1:3), it returns the cartesian coordinates x, y, and z from the object center; **Array:** A(1:3), it returns the angles from axis x, y and z.
| Example | ```
%% Create the constructor Class
V = VREP;

%% Connect Matlab to V-Rep
V.vConnect;

%% Get handle tags from object Disc
V.vObject('Disc');

%% Get position and angle from object Disc
[Xc, A]= V.vGetObjPosition('Disc');
``` |

vGetLaserData

Description	This is responsible to get the laser data from Pioneer.
synopsis	**Method:** vGetLaserData(Index);
Parameters	**Index:** Usually it is 1, if there is more than one robot, it's necessary to specify the index robot number.
return	**Array:** Map(1:181,1:3), the first column indicates the coordinate x values, the second column indicates the y values, and the third column indicates the distance.
| Example | ```matlab
%% Create the constructor Class
V = VREP;

%% Connect Matlab to V-Rep
V.vConnect;

%% Get handle tags from Pioneer
V.vHandle('Pioneer_p3dx');

%% Get laser data
Map = V.vGetLaserData(1);
``` |

vSendControlSignals

Description	This is responsible to send control signals to Pioneer.
synopsis	**Method:** vSendControlSignals(Velocity,Index);
Parameters	**Velocity:** It contains on the first index the linear velocity, and on the second index the angular velocity. **Index:** Usually it is 1, if there is more than one robot, it's necessary to specify the index robot number.
return	**None**
| Example | ```matlab
%% Create the constructor Class
V = VREP;

%% Connect Matlab to V-Rep
V.vConnect;

%% Get handle tags from Pioneer
V.vHandle('Pioneer_p3dx');

%% Set Velocity
Ud = [0.3;0];
V.vSendControlSignals(Ud,1);
``` |