Innovation Factories C/AV Challenge

Generated by Doxygen 1.8.15

1 Class Index 1

1 Class Index	1
1.1 Class List	1
2 File Index	1
2.1 File List	1
3 Class Documentation	1
3.1 car.Car Class Reference	1
3.1.1 Detailed Description	2
3.1.2 Member Function Documentation	2
3.2 car.Node Class Reference	3
3.2.1 Detailed Description	4
3.2.2 Constructor & Destructor Documentation	4
3.2.3 Member Function Documentation	5
4 File Documentation	8
4.1 car.py File Reference	8
4.1.1 Detailed Description	8
Index	9
1 Class Index	
1.1 Class List	
Here are the classes, structs, unions and interfaces with brief descriptions:	
car.Car An ADT that represents a Car	1
car.Node An ADT that represents a Node	3
2 File Index	
2.1 File List	
Here is a list of all documented files with brief descriptions:	
car.py Provides the Abstract Data Types for Car and Node objects	8
3 Class Documentation	
3.1 car.Car Class Reference	
An ADT that represents a Car.	

Public Member Functions

• def __init__ (self, ID, speed)

this is the initializer method

def get_ID (self)

this method is used to get the unique ID of the car

def get_speed (self)

this method is used to get the current speed of the car

• def update_speed (self, modifier)

this method is used to update the current speed of the car

def <u>__str__</u> (self)

this method is return the car object and it's information in the form of a string

Public Attributes

· speed

3.1.1 Detailed Description

An ADT that represents a Car.

Parameters

ID	is a unique identifier that is linked to each car
speed	is the speed of the car

3.1.2 Member Function Documentation

this method is return the car object and it's information in the form of a string

Returns

the car's information in the form of a string

3.1.2.2 get_ID()

```
def car.Car.get_ID (
     self )
```

this method is used to get the unique ID of the car

Returns

the ID of the car

3.1.2.3 get_speed()

this method is used to get the current speed of the car

Returns

the speed of the car

3.1.2.4 update_speed()

this method is used to update the current speed of the car

Parameters

modifier is the multiplier that we apply to the current speed to get to the new speed

Returns

the current of the car after the update

The documentation for this class was generated from the following file:

· car.py

3.2 car.Node Class Reference

An ADT that represents a Node.

Public Member Functions

- def __init__ (self, humidity, audio, temperature, pressure, vibration, video)
 Node constructor.
- def get_humidity (self)

Gets the humidity a Node records.

def get_audio (self)

Gets the audio a Node records.

def get_temperature (self)

Gets the temperature a Node records.

def get_pressure (self)

Gets the pressure a Node records.

• def get_vibration (self)

Gets the vibration a Node records.

• def determine_rain (self, audio)

Checks conditions to determine when rain will occur.

• def determine_snow (self, video)

Checks conditions to determine when snow will occur.

• def determine_fog (self)

Checks conditions to determine when fog will occur.

def determine_wind (self)

Checks conditions to determine when wind will occur.

def determine_day_and_night (self)

Checks conditions to determine when the time of day is in the morning or night.

• def dynamic_speed (self, car)

Checks conditions to determine by what magnitude to reduce the overall speed by @car Object of type car that will have its speed modified based on smallest magnitude.

Public Attributes

- · humidity
- · audio
- · temperature
- pressure
- vibration
- video

3.2.1 Detailed Description

An ADT that represents a Node.

3.2.2 Constructor & Destructor Documentation

Node constructor.

Initializes a Node object with an empty Node

Parameters

humidity	The humidity reported in Percent
audio	The audio is how strong the rain is by using the sound
temperature	The temperature reported in Kelvin
pressure	The pressure reported in Pascals
vibration	The vibration is the vibration of the node in meters per second

Generated by Doxygen

3.2.3 Member Function Documentation

3.2.3.1 determine_day_and_night()

```
\label{lem:car.Node.determine_day_and_night (} self \ )
```

Checks conditions to determine when the time of day is in the morning or night.

Returns

returns reduced speed by a fractional portion the time of day is night, and 1 if it is in the morning

3.2.3.2 determine_fog()

Checks conditions to determine when fog will occur.

Returns

returns reduced speed by a fractional portion if fog exists and 0 otherwise

3.2.3.3 determine_rain()

Checks conditions to determine when rain will occur.

Parameters

audio The audio is used to determine hard rain or light rain
--

Returns

returns reduced speed by a fractional portion if rain exists and 0 if otherwise

3.2.3.4 determine_snow()

```
\begin{tabular}{ll} $\operatorname{def car.Node.determine\_snow} & ( & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & &
```

Checks conditions to determine when snow will occur.

Parameters

	video	The video is used to determine heavy snow, light snow, or medium snow	
--	-------	---	--

Returns

returns reduced speed by a fractional portion if snow exists and 0 otherwise

3.2.3.5 determine_wind()

Checks conditions to determine when wind will occur.

Returns

returns reduced speed by a fractional portion if high wind exists, 1 if windkmh is less than or equal to 25 and 0 otherwise

3.2.3.6 dynamic_speed()

Checks conditions to determine by what magnitude to reduce the overall speed by @car Object of type car that will have its speed modified based on smallest magnitude.

Returns

returns reduced speed based on the smallest magnitude reduced

3.2.3.7 get_audio()

Gets the audio a Node records.

Returns

returns the audio

3.2.3.8 get_humidity()

Gets the humidity a Node records.

Returns

returns the humidity

3.2.3.9 get_pressure()

```
\begin{tabular}{ll} $\operatorname{def car.Node.get\_pressure} & ( \\ & self \end{tabular} ) \label{eq:car.Node.get_pressure}
```

Gets the pressure a Node records.

Returns

returns the pressure

3.2.3.10 get_temperature()

```
\begin{tabular}{ll} def & car.Node.get\_temperature ( \\ & self ) \end{tabular}
```

Gets the temperature a Node records.

Returns

returns the temperature

3.2.3.11 get_vibration()

```
\begin{tabular}{ll} \tt def car.Node.get\_vibration ( \\ & self ) \end{tabular}
```

Gets the vibration a Node records.

Returns

returns the vibration

The documentation for this class was generated from the following file:

car.py

4 File Documentation

4.1 car.py File Reference

Provides the Abstract Data Types for Car and Node objects.

Classes

· class car.Car

An ADT that represents a Car.

• class car.Node

An ADT that represents a Node.

Functions

• def car.main ()

Variables

• int car.dewpoint = 273

A constant that is set to Hamilton on January 26th 2020 for the dewpoint in kelvin.

4.1.1 Detailed Description

Provides the Abstract Data Types for Car and Node objects.

Author

Mostafa Mohsen, Chris Vishnu, Seif El Tobgy, Saif Fadhel

Date

26/01/2020

Index

```
___init_
    car.Node, 4
__str__
    car.Car, 2
car.Car, 1
    __str__, 2
    get_ID, 2
    get_speed, 2
    update_speed, 3
car.Node, 3
      _init__, 4
    determine_day_and_night, 5
    determine_fog, 5
    determine_rain, 5
    determine_snow, 5
    determine_wind, 6
    dynamic_speed, 6
    get_audio, 6
    get_humidity, 6
    get_pressure, 7
    get_temperature, 7
     get_vibration, 7
car.py, 8
determine_day_and_night
    car.Node, 5
determine_fog
    car.Node, 5
determine rain
    car.Node, 5
determine_snow
    car.Node, 5
determine_wind
    car.Node, 6
dynamic_speed
    car.Node, 6
get_audio
    car.Node, 6
get humidity
    car.Node, 6
get_ID
     car.Car, 2
get_pressure
    car.Node, 7
get_speed
    car.Car, 2
get_temperature
    car.Node, 7
get_vibration
    car.Node, 7
update_speed
    car.Car, 3
```