

OOP Project

Generated by Doxygen 1.9.2

1 Hierarchical Index	1
1.1 Class Hierarchy	1
2 Class Index	3
2.1 Class List	3
3 File Index	5
3.1 File List	5
4 Class Documentation	7
4.1 Bike Class Reference	7
4.1.1 Detailed Description	7
4.1.2 Constructor & Destructor Documentation	7
4.1.2.1 Bike()	8
4.1.3 Member Function Documentation	9
4.1.3.1 draw()	9
4.2 Car Class Reference	9
4.2.1 Detailed Description	10
4.2.2 Constructor & Destructor Documentation	10
4.2.2.1 Car()	10
4.2.3 Member Function Documentation	10
4.2.3.1 draw()	10
4.3 Debris Class Reference	11
4.3.1 Detailed Description	11
4.3.2 Constructor & Destructor Documentation	11
4.3.2.1 Debris()	11
4.4 Digit Class Reference	12
4.4.1 Detailed Description	12
4.4.2 Constructor & Destructor Documentation	12
4.4.2.1 Digit()	12
4.4.3 Member Function Documentation	13
4.4.3.1 animate()	13
4.4.3.2 draw()	13
4.5 Game Class Reference	13
4.5.1 Detailed Description	14
4.6 Jamalo Class Reference	14
4.6.1 Detailed Description	15
4.6.2 Constructor & Destructor Documentation	15
4.6.2.1 Jamalo()	15
4.6.3 Member Function Documentation	15
4.6.3.1 createObjects()	15
4.6.3.2 drawObjects()	15
4.6.3.3 hitRegistered() [1/2]	16

4.6.3.4 hitRegistered() [2/2]	16
4.6.3.5 isOver()	16
4.6.3.6 move()	17
4.6.3.7 radar()	17
4.7 Lives Class Reference	17
4.8 RaceCar Class Reference	18
4.8.1 Detailed Description	18
4.8.2 Constructor & Destructor Documentation	18
4.8.2.1 RaceCar()	18
4.8.3 Member Function Documentation	19
4.8.3.1 draw()	19
4.9 Sheep Class Reference	19
4.9.1 Detailed Description	20
4.9.2 Constructor & Destructor Documentation	20
4.9.2.1 Sheep()	20
4.9.3 Member Function Documentation	20
4.9.3.1 draw()	20
4.9.3.2 getter()	21
4.9.3.3 move()	21
4.10 Timer Class Reference	21
4.11 Truck Class Reference	22
4.11.1 Detailed Description	22
4.11.2 Constructor & Destructor Documentation	22
4.11.2.1 Truck()	22
4.11.3 Member Function Documentation	23
4.11.3.1 draw()	23
4.12 Truck2 Class Reference	23
4.12.1 Detailed Description	24
4.12.2 Constructor & Destructor Documentation	24
4.12.2.1 Truck2()	24
4.12.3 Member Function Documentation	24
4.12.3.1 draw()	24
4.13 Unit Class Reference	25
4.13.1 Detailed Description	25
4.13.2 Constructor & Destructor Documentation	25
4.13.2.1 Unit()	25
4.13.3 Member Function Documentation	26
4.13.3.1 draw()	26
4.14 Vehicle Class Reference	26
4.14.1 Detailed Description	27
4.14.2 Constructor & Destructor Documentation	27
4.14.2.1 Vehicle()	27

4.14.3 Member Function Documentation	27
4.14.3.1 changeLaneDown()	27
4.14.3.2 changeLaneUp()	27
4.14.3.3 draw()	28
4.14.3.4 getter()	28
5 File Documentation	29
5.1 Bike.hpp	29
5.2 Car.hpp	29
5.3 Debris.hpp	29
5.4 Digit.hpp	29
5.5 game.hpp	30
5.6 Jamalo.hpp	30
5.7 Lives.hpp	31
5.8 Racecar.hpp	31
5.9 Sheep.hpp	31
5.10 Timer.hpp	31
5.11 Truck.hpp	32
5.12 Truck2.hpp	32
5.13 Unit.hpp	32
5.14 Vehicle.hpp	32
Index	35

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Game	13
Jamalo	14
Timer	21
Unit	25
Debris	11
Digit	12
Lives	17
Sheep	19
Vehicle	26
Bike	7
Car	9
RaceCar	18
Truck	22
Truck2	23

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Bike	This is the class for the Bike type objects	7
Car	This is the class for the Car type objects	9
Debris	This class creates the debris when a Vehicle is destroyed	11
Digit	Digit used in timer	12
Game	Game class is the container for Jamalo class and handles input and output interfacing	13
Jamalo	This is the Level Design class, and contains all other game related classes	14
Lives	17
RaceCar	This is the class for the Racecar type objects	18
Sheep	This class generates the Sheep	19
Timer	21
Truck	This is the class for the Truck type objects	22
Truck2	This is the class for the Truck2 type objects	23
Unit	The unit class from which all game related classes are inherited	25
Vehicle	Vehicle class, the parent for all vehicles	26

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

Bike.hpp	??
Car.hpp	??
Debris.hpp	??
Digit.hpp	??
game.hpp	??
Jamalo.hpp	??
Lives.hpp	??
Racecar.hpp	??
Sheep.hpp	??
Timer.hpp	??
Truck.hpp	??
Truck2.hpp	??
Unit.hpp	??
Vehicle.hpp	??

Chapter 4

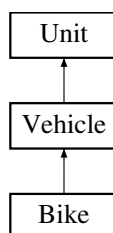
Class Documentation

4.1 Bike Class Reference

This is the class for the [Bike](#) type objects.

```
#include <Bike.hpp>
```

Inheritance diagram for Bike:



Public Member Functions

- [Bike](#) (SDL_Renderer *rend, SDL_Texture *ast, SDL_Rect mov)
Construct a new [Bike](#)::[Bike](#) object.
- void [draw](#) (bool flag)
Draws the [Bike](#) type Object.

4.1.1 Detailed Description

This is the class for the [Bike](#) type objects.

This class inherits from [Vehicle](#) class and provides the sprite and mover coordinates to it.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 Bike()

```
Bike::Bike (
    SDL_Renderer * rend,
    SDL_Texture * ast,
    SDL_Rect mov )
```

Construct a new [Bike:: Bike](#) object.

Parameters

<i>rend</i>	
<i>ast</i>	
<i>mov</i>	

4.1.3 Member Function Documentation

4.1.3.1 draw()

```
void Bike::draw (  
    bool flag )
```

Draws the [Bike](#) type Object.

Calls the draw function of [Vehicle](#) and passes the flag to the draw function of [Vehicle](#) class which then determines whether the [Bike](#) will move forward or not.

Parameters

<i>flag</i>	
-------------	--

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

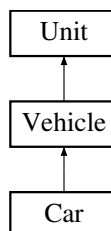
- [Bike.hpp](#)
- [Bike.cpp](#)

4.2 Car Class Reference

This is the class for the [Car](#) type objects.

```
#include <Car.hpp>
```

Inheritance diagram for Car:



Public Member Functions

- [Car](#) (SDL_Renderer *rend, SDL_Texture *ast, SDL_Rect mov)
Construct a new [Car](#):: [Car](#) object.
- void [draw](#) (bool flag)
Draws the [Car](#) type Object.

4.2.1 Detailed Description

This is the class for the [Car](#) type objects.

This class inherits from [Vehicle](#) class and provides the sprite and mover coordinates to it.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 Car()

```
Car::Car (
    SDL_Renderer * rend,
    SDL_Texture * ast,
    SDL_Rect mov )
```

Construct a new [Car](#):: [Car](#) object.

Parameters

<i>rend</i>	
<i>ast</i>	
<i>mov</i>	

4.2.3 Member Function Documentation

4.2.3.1 draw()

```
void Car::draw (
    bool flag )
```

Draws the [Car](#) type Object.

Calls the draw function of [Vehicle](#) and passes the flag to the draw function of [Vehicle](#) class which then determines whether the [Car](#) will move forward or not.

Parameters

<i>flag</i>	
-------------	--

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

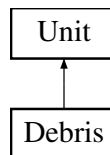
- Car.hpp
- Car.cpp

4.3 Debris Class Reference

This class creates the debris when a [Vehicle](#) is destroyed.

```
#include <Debris.hpp>
```

Inheritance diagram for Debris:



Public Member Functions

- void **animate** ()
Animates and draws the debris.
- [Debris](#) (SDL_Renderer *rend, SDL_Texture *ast, SDL_Rect mov)
Construct a new [Debris::Debris](#) object.

4.3.1 Detailed Description

This class creates the debris when a [Vehicle](#) is destroyed.

Whenever a [Car](#) collides with another [Car](#) or with [Sheep](#) an explosion animation is displayed, this class handles the animation and drawing of the explosion.

4.3.2 Constructor & Destructor Documentation

4.3.2.1 Debris()

```
Debris::Debris (
    SDL_Renderer * rend,
    SDL_Texture * ast,
    SDL_Rect mov )
```

Construct a new [Debris::Debris](#) object.

Parameters

<i>rend</i>	
<i>ast</i>	
<i>mov</i>	

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

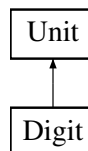
- Debris.hpp
- Debris.cpp

4.4 Digit Class Reference

[Digit](#) used in timer.

```
#include <Digit.hpp>
```

Inheritance diagram for Digit:



Public Member Functions

- [Digit](#) (SDL_Renderer *rend, SDL_Texture *ast, SDL_Rect mov, SDL_Rect src)
Construct a new [Digit::Digit](#) object.
- void [draw](#) ()
Draws the digit.
- void [animate](#) (unsigned int res)
Animates the digit.

4.4.1 Detailed Description

[Digit](#) used in timer.

Creates a digit type object to be used in the timer.

4.4.2 Constructor & Destructor Documentation

4.4.2.1 Digit()

```
Digit::Digit (
    SDL_Renderer * rend,
    SDL_Texture * ast,
    SDL_Rect mov,
    SDL_Rect src )
```

Construct a new [Digit::Digit](#) object.

Parameters

<i>rend</i>	
<i>ast</i>	
<i>mov</i>	
<i>src</i>	

4.4.3 Member Function Documentation

4.4.3.1 `animate()`

```
void Digit::animate (
    unsigned int res )
```

Animates the digit.

Changes the src of the digits based on the counter

Parameters

<i>res</i>	
------------	--

4.4.3.2 `draw()`

```
void Digit::draw ( )
```

Draws the digit.

Calls the draw function of the [Unit](#) class to draw the digit.

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

- Digit.hpp
- Digit.cpp

4.5 Game Class Reference

[Game](#) class is the container for [Jamalo](#) class and handles input and output interfacing.

```
#include <game.hpp>
```

Public Member Functions

- bool **init** ()
- bool **loadMedia** ()
- void **close** ()
- SDL_Texture * **loadTexture** (std::string path)
- void **run** ()

4.5.1 Detailed Description

[Game](#) class is the container for [Jamalo](#) class and handles input and output interfacing.

game class takes the inputs from the users and provides it to the jamalo class to be processed. It also takes care of the start and the end screens.

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

- game.hpp
- game.cpp

4.6 Jamalo Class Reference

This is the Level Design class, and contains all other game related classes.

```
#include <Jamalo.hpp>
```

Public Member Functions

- [Jamalo](#) (SDL_Renderer *, SDL_Texture *)
Construct a new [Jamalo](#)::[Jamalo](#) object.
- void [drawObjects](#) ()
Draws all objects.
- void [createObjects](#) ()
Generates [Vehicle](#) type objects.
- void [move](#) (char x)
Calls the move function of the [Sheep](#) and passes it the user input.
- bool [hitRegistered](#) ([Vehicle](#) v)
Checks if vehicle is colliding with sheep.
- bool [hitRegistered](#) ([Vehicle](#) v, list< [Vehicle](#) > &vehicles)
Checks if any vehicles are currently colliding.
- void [radar](#) ([Vehicle](#) ¤tV)
Implementation of radar system in the Vehicles.
- bool [isOver](#) ()
Checks game state.

4.6.1 Detailed Description

This is the Level Design class, and contains all other game related classes.

The [Jamalo](#) class aggregates all other game related classes, namely [Sheep](#), [Vehicle](#) and its childs, [Timer](#) and [Lives](#). All top-level functionality is implemented in this class, the purpose of this class is to connect all objects in a proper manner, and to carry out class operations over them. Majority of the game mechanics are implemented here. This class: -Draws all objects -Creates all objects -Contains the radar system through which vehicles slow down if there is a vehicle in front of them -Checks for collisions between objects

4.6.2 Constructor & Destructor Documentation

4.6.2.1 Jamalo()

```
Jamalo::Jamalo (
    SDL_Renderer * renderer,
    SDL_Texture * asst )
```

Construct a new [Jamalo](#):: [Jamalo](#) object.

Creates a [Jamalo](#) type object and initializes values for all other objects in the game.

Parameters

<i>renderer</i>	
<i>asst</i>	

4.6.3 Member Function Documentation

4.6.3.1 createObjects()

```
void Jamalo::createObjects ( )
```

Generates [Vehicle](#) type objects.

Randomly generates [Vehicle](#) type objects and randomly selects a lane for them to spawn in.

4.6.3.2 drawObjects()

```
void Jamalo::drawObjects ( )
```

Draws all objects.

Iterates over the list of all vehicles and calls their draw function. It also calls the draw function of all other objects.

4.6.3.3 hitRegistered() [1/2]

```
bool Jamalo::hitRegistered (
    Vehicle vehicle_hit )
```

Checks if vehicle is colliding with sheep.

The function is an overloaded instance which checks if the given vehicle is colliding with the sheep or not.

Parameters

<i>vehicle_hit</i>	
--------------------	--

Returns

true

false

4.6.3.4 hitRegistered() [2/2]

```
bool Jamalo::hitRegistered (
    Vehicle v,
    list< Vehicle > & vehicles )
```

Checks if any vehicles are currently colliding.

The function takes a [Vehicle](#) type object and compares it with all other Vehicles to see if their mover rectangles are overlapping or not, this is to check if the vehicles are colliding with each other or not.

Parameters

<i>v</i>	
<i>vehicles</i>	

Returns

true

false

4.6.3.5 isOver()

```
bool Jamalo::isOver ( )
```

Checks game state.

Returns true if game is over i.e all 3 lives have been lost.

Returns

true
false

4.6.3.6 move()

```
void Jamalo::move (
    char x )
```

Calls the move function of the [Sheep](#) and passes it the user input.

Parameters

<i>x</i>	
----------	--

4.6.3.7 radar()

```
void Jamalo::radar (
    Vehicle & current )
```

Implementation of radar system in the Vehicles.

The function a vehicle and determines the distance between that vehicle and the vehicle in front of it, if there is a vehicle in front of the current vehicle then it gives a signal for the current vehicle to stop. It is also responsity for the drawing the vehicles.

Parameters

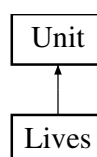
<i>current</i>	
----------------	--

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

- Jamalo.hpp
- Jamalo.cpp

4.7 Lives Class Reference

Inheritance diagram for Lives:



Public Member Functions

- **Lives** (SDL_Renderer *rend, SDL_Texture *ast, SDL_Rect mov)
- void **draw** ()

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

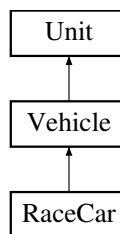
- Lives.hpp
- Lives.cpp

4.8 RaceCar Class Reference

This is the class for the Racecar type objects.

```
#include <Racecar.hpp>
```

Inheritance diagram for RaceCar:



Public Member Functions

- **RaceCar** (SDL_Renderer *rend, SDL_Texture *ast, SDL_Rect mov)
*Construct a new Race **Car**:: Race **Car** object.*
- void **draw** (bool flag)
Draws the Racecar type Object.

4.8.1 Detailed Description

This is the class for the Racecar type objects.

This class inherits from **Vehicle** class and provides the sprite and mover coordinates to it.

4.8.2 Constructor & Destructor Documentation

4.8.2.1 RaceCar()

```

RaceCar::RaceCar (
    SDL_Renderer * rend,
    SDL_Texture * ast,
    SDL_Rect mov )
  
```

Construct a new Race **Car**:: Race **Car** object.

Parameters

<i>rend</i>	
<i>ast</i>	
<i>mov</i>	

4.8.3 Member Function Documentation

4.8.3.1 draw()

```
void RaceCar::draw (
    bool flag )
```

Draws the Racecar type Object.

Calls the draw function of [Vehicle](#) and passes the flag to the draw function of [Vehicle](#) class which then determines whether the Racecar will move forward or not.

Parameters

<i>flag</i>	
-------------	--

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

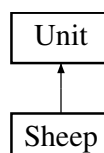
- Racecar.hpp
- Racecar.cpp

4.9 Sheep Class Reference

This class generates the [Sheep](#).

```
#include <Sheep.hpp>
```

Inheritance diagram for Sheep:



Public Member Functions

- [Sheep](#) (SDL_Renderer *rend, SDL_Texture *ast)
Construct a new [Sheep:: Sheep](#) object.
- SDL_Rect [getter](#) () const
The getter function for the mover rectangle of the [Sheep](#).
- void [draw](#) ()
Draws the sheep.
- void [move](#) (char x)
Moves the sheep.
- void [getHit](#) ()
A function to display effects when sheep gets hit.

4.9.1 Detailed Description

This class generates the [Sheep](#).

The protagonist sheep and functionality related to its movement, drawing and the hit reactions are implemented here.

4.9.2 Constructor & Destructor Documentation

4.9.2.1 Sheep()

```
Sheep::Sheep (
    SDL_Renderer * rend,
    SDL_Texture * ast )
```

Construct a new [Sheep:: Sheep](#) object.

Parameters

<i>rend</i>	
<i>ast</i>	

4.9.3 Member Function Documentation

4.9.3.1 draw()

```
void Sheep::draw ( )
```

Draws the sheep.

Calls the draw function from the [Unit](#) class and passes the relevant functions to it.

4.9.3.2 getter()

```
SDL_Rect Sheep::getter ( ) const
```

The getter function for the mover rectangle of the [Sheep](#).

Returns

SDL_Rect

4.9.3.3 move()

```
void Sheep::move (
    char x )
```

Moves the sheep.

The function takes input from the [Jamalo](#) class and moves the sheep based on the key pressed. If sheep is at the border then it prevents it from moving any further.

Parameters

x	
---	--

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

- Sheep.hpp
- Sheep.cpp

4.10 Timer Class Reference

Public Member Functions

- **Timer** (SDL_Renderer *rend, SDL_Texture *ast)
- void **draw** ()
- void **count** ()

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

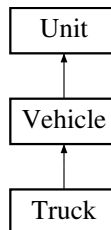
- Timer.hpp
- Timer.cpp

4.11 Truck Class Reference

This is the class for the [Truck](#) type objects.

```
#include <Truck.hpp>
```

Inheritance diagram for Truck:



Public Member Functions

- [Truck](#) (SDL_Renderer *rend, SDL_Texture *ast, SDL_Rect mov)
Construct a new [Truck](#):: [Truck](#) object.
- void [draw](#) (bool flag)
Draws the [Truck](#) type Object.

4.11.1 Detailed Description

This is the class for the [Truck](#) type objects.

This class inherits from [Vehicle](#) class and provides the sprite and mover coordinates to it.

4.11.2 Constructor & Destructor Documentation

4.11.2.1 Truck()

```
Truck::Truck (
    SDL_Renderer * rend,
    SDL_Texture * ast,
    SDL_Rect mov )
```

Construct a new [Truck](#):: [Truck](#) object.

Parameters

<i>rend</i>	
<i>ast</i>	
<i>mov</i>	

4.11.3 Member Function Documentation

4.11.3.1 draw()

```
void Truck::draw (
    bool flag )
```

Draws the [Truck](#) type Object.

Calls the draw function of [Vehicle](#) and passes the flag to the draw function of [Vehicle](#) class which then determines whether the [Truck](#) will move forward or not.

Parameters

<i>flag</i>	
-------------	--

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

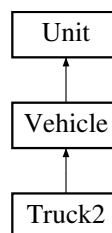
- Truck.hpp
- Truck.cpp

4.12 Truck2 Class Reference

This is the class for the [Truck2](#) type objects.

```
#include <Truck2.hpp>
```

Inheritance diagram for Truck2:



Public Member Functions

- [Truck2](#) (SDL_Renderer *rend, SDL_Texture *ast, SDL_Rect mov)
Construct a new [Truck 2](#):: [Truck 2](#) object.
- void [draw](#) (bool flag)
Draws the [Truck2](#) type Object.

4.12.1 Detailed Description

This is the class for the [Truck2](#) type objects.

This class inherits from [Vehicle](#) class and provides the sprite and mover coordinates to it.

4.12.2 Constructor & Destructor Documentation

4.12.2.1 Truck2()

```
Truck2::Truck2 (
    SDL_Renderer * rend,
    SDL_Texture * ast,
    SDL_Rect mov )
```

Construct a new [Truck 2](#):: [Truck 2](#) object.

Parameters

<i>rend</i>	
<i>ast</i>	
<i>mov</i>	

4.12.3 Member Function Documentation

4.12.3.1 draw()

```
void Truck2::draw (
    bool flag )
```

Draws the [Truck2](#) type Object.

Calls the draw function of [Vehicle](#) and passes the flag to the draw function of [Vehicle](#) class which then determines whether the [Truck](#) will move forward or not.

Parameters

<i>flag</i>	
-------------	--

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

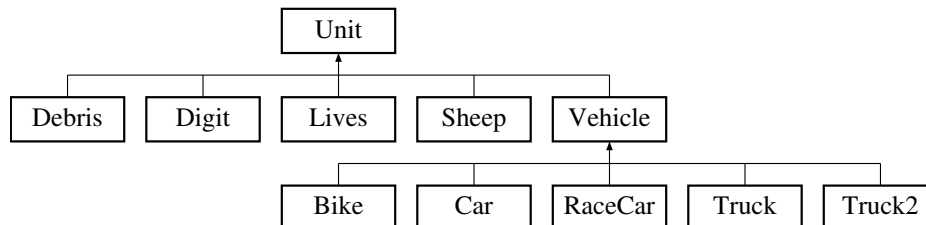
- [Truck2.hpp](#)
- [Truck2.cpp](#)

4.13 Unit Class Reference

The unit class from which all game related classes are inherited.

```
#include <Unit.hpp>
```

Inheritance diagram for Unit:



Public Member Functions

- **Unit** (SDL_Renderer *, SDL_Texture *)
*Construct a new **Unit**:: **Unit** object.*
- void **draw** (SDL_Rect srcRect, SDL_Rect moverRect)
Draws an object.

4.13.1 Detailed Description

The unit class from which all game related classes are inherited.

This class provides the drawing functionality to all other classes that need to draw a sprite on the screen.

4.13.2 Constructor & Destructor Documentation

4.13.2.1 Unit()

```
Unit::Unit (
    SDL_Renderer * rend,
    SDL_Texture * ast )
```

Construct a new **Unit**:: **Unit** object.

Parameters

<i>rend</i>	
<i>ast</i>	

4.13.3 Member Function Documentation

4.13.3.1 draw()

```
void Unit::draw (
    SDL_Rect srcRect,
    SDL_Rect moverRect )
```

Draws an object.

This function draws an object based on the src and mover rectangles passes to it.

Parameters

<i>srcRect</i>	
<i>moverRect</i>	

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

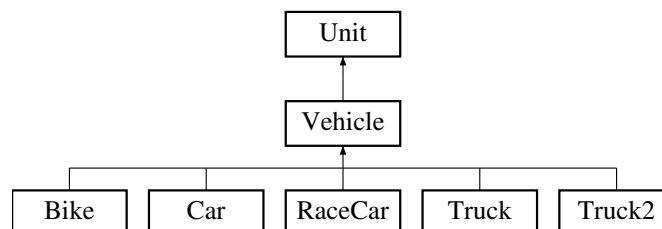
- Unit.hpp
- Unit.cpp

4.14 Vehicle Class Reference

[Vehicle](#) class, the parent for all vehicles.

```
#include <Vehicle.hpp>
```

Inheritance diagram for Vehicle:



Public Member Functions

- [Vehicle](#) (SDL_Renderer *rend, SDL_Texture *ast, SDL_Rect mov, SDL_Rect src, int s)
Construct a new [Vehicle](#):: [Vehicle](#) object.
- void [draw](#) (bool flag)
Draws the Vehicles.
- SDL_Rect [getter](#) () const
Returns mover rect of the [Vehicle](#) object.
- void [changeLaneUp](#) ()
Moves the car to the upper lane if any.
- void [changeLaneDown](#) ()
Moves the car to the lower lane if any.

4.14.1 Detailed Description

[Vehicle](#) class, the parent for all vehicles.

This class is inherited by [Bike](#), [Car](#), [Racecar](#), [Truck](#) and [Truck2](#). It provides all the functionality relating to the movement of the cars and their speed.

4.14.2 Constructor & Destructor Documentation

4.14.2.1 Vehicle()

```
Vehicle::Vehicle (
    SDL_Renderer * rend,
    SDL_Texture * ast,
    SDL_Rect mov,
    SDL_Rect src,
    int s )
```

Construct a new [Vehicle::Vehicle](#) object.

Parameters

<i>rend</i>	
<i>ast</i>	
<i>mov</i>	
<i>src</i>	
<i>s</i>	

4.14.3 Member Function Documentation

4.14.3.1 changeLaneDown()

```
void Vehicle::changeLaneDown ( )
```

Moves the car to the lower lane if any.

The function carries out the process of making the car move to the lower lane, it is not being called as it contains bugs at the moment.

4.14.3.2 changeLaneUp()

```
void Vehicle::changeLaneUp ( )
```

Moves the car to the upper lane if any.

The function carries out the process of making the car move to the upper lane, it is not being called as it contains bugs at the moment.

4.14.3.3 draw()

```
void Vehicle::draw (
    bool flag )
```

Draws the Vehicles.

The function calls the draw function of the [Unit](#) class, it also changes the mover rectangle based on the flag passed to it.

Parameters

<i>flag</i>	
-------------	--

4.14.3.4 getter()

```
SDL_Rect Vehicle::getter ( ) const
```

Returns mover rect of the [Vehicle](#) object.

Returns

SDL_Rect

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

- Vehicle.hpp
- Vehicle.cpp

Chapter 5

File Documentation

5.1 Bike.hpp

```
1 #include "Vehicle.hpp"
2
10 class Bike : public Vehicle
11 {
12
13 public:
14     Bike(SDL_Renderer *rend, SDL_Texture *ast, SDL_Rect mov);
15     void draw(bool flag);
16 };
```

5.2 Car.hpp

```
1 #include "Vehicle.hpp"
2
10 class Car : public Vehicle
11 {
12
13 public:
14     Car(SDL_Renderer *rend, SDL_Texture *ast, SDL_Rect mov);
15     void draw(bool flag);
16 };
```

5.3 Debris.hpp

```
1 #include "Unit.hpp"
2
10 class Debris : public Unit
11 {
12
13     SDL_Rect mover;
14     SDL_Rect src = {551, 9, 24, 25};
15
16 public:
17     void
18     animate();
19
20     Debris(SDL_Renderer *rend, SDL_Texture *ast, SDL_Rect mov);
21 };
```

5.4 Digit.hpp

```
1 #include "Unit.hpp"
2 #include <SDL.h>
3
10 class Digit : public Unit
```

```

11 {
12 private:
13     SDL_Rect src, mover;
14
15 public:
16     Digit(SDL_Renderer *rend, SDL_Texture *ast, SDL_Rect mov, SDL_Rect src);
17     void draw();
18     void animate(unsigned int res);
19 };

```

5.5 game.hpp

```

1 #include <SDL.h>
2 #include <SDL_image.h>
3 #include <stdio.h>
4 #include <iostream>
5 #include <string>
6 #include <stdlib.h>
7 #include <time.h>
8 #include "Jamalo.hpp"
9 #include <SDL_mixer.h>
10
11 class Game
12 {
13     // Game state
14     bool start = false;
15     bool end = false;
16
17     //Screen dimension constants
18     const int SCREEN_WIDTH = 1000;
19     const int SCREEN_HEIGHT = 600;
20
21     //The window we'll be rendering to
22     SDL_Window *gWindow = NULL;
23
24     //The window renderer
25     SDL_Renderer *gRenderer = NULL;
26
27     //Current displayed texture
28     SDL_Texture *gTexture = NULL;
29     //global reference to png image sheets
30     SDL_Texture *assets = NULL;
31     Mix_Music *bgMusic = NULL;
32
33 public:
34     bool init();
35     bool loadMedia();
36     void close();
37     SDL_Texture *loadTexture(std::string path);
38     void run();
39 };

```

5.6 Jamalo.hpp

```

1 #pragma once
2 #include <SDL.h>
3 #include "Vehicle.hpp"
4 #include "Sheep.hpp"
5 #include "Timer.hpp"
6 #include "Lives.hpp"
7 #include <list>
8 using namespace std;
9
10 class Jamalo
11 {
12     Sheep j;
13     list<Vehicle> vehicles;
14     SDL_Renderer *gRenderer;
15     SDL_Texture *assets;
16     int nLives;
17     Timer t;
18     Lives l1;
19     Lives l2;
20     Lives l3;
21
22 public:
23     Jamalo(SDL_Renderer *, SDL_Texture *);
24     void drawObjects();

```

```

38     void createObjects();
39     void move(char x);
40     bool hitRegistered(Vehicle v);
41     bool hitRegistered(Vehicle v, list<Vehicle> &vehicles);
42     // bool hitRegistered()
43     void radar(Vehicle &currentV);
44     bool isOver();
45 }
46
47 ;

```

5.7 Lives.hpp

```

1 #include "Unit.hpp"
2 #include <SDL.h>
3
4 class Lives : public Unit
5 {
6     SDL_Rect src = {12, 124, 55, 54};
7     SDL_Rect mover;
8
9 public:
10     Lives(SDL_Renderer *rend, SDL_Texture *ast, SDL_Rect mov);
11
12     void draw();
13 };

```

5.8 Racecar.hpp

```

1 #include "Vehicle.hpp"
2
10 class RaceCar : public Vehicle
11 {
12
13 public:
14     RaceCar(SDL_Renderer *rend, SDL_Texture *ast, SDL_Rect mov);
15     void draw(bool flag);
16 };

```

5.9 Sheep.hpp

```

1 #pragma once
2 #include "Unit.hpp"
3 using namespace std;
4
12 class Sheep : public Unit
13 {
14
15 private:
16     SDL_Rect src = {16, 23, 43, 42};
17     SDL_Rect mover = {10, 430, 50, 50};
18     int nLives;
19
20 public:
21     // SDL_Rect mover;
22     Sheep(SDL_Renderer *rend, SDL_Texture *ast);
23
24     SDL_Rect getter() const;
25     void draw();
26     void move(char x);
27     void getHit();
28     // friend bool Jamalo::hitRegistered(Vehicle v);
29 };

```

5.10 Timer.hpp

```

1 #include "Digit.hpp"
2
3 class Timer
4 {
5     private:

```

```
6     Digit minutes;
7     Digit second1;
8     Digit second2;
9     int starttime;
10
11 public:
12     Timer(SDL_Renderer *rend, SDL_Texture *ast);
13     // int start_timer();
14     void draw();
15
16     void count();
17 };
18
```

5.11 Truck.hpp

```
1 #include "Vehicle.hpp"
2
10 class Truck : public Vehicle
11 {
12
13 public:
14     Truck(SDL_Renderer *rend, SDL_Texture *ast, SDL_Rect mov);
15     void draw(bool flag);
16 };

```

5.12 Truck2.hpp

```
1 #include "Vehicle.hpp"
2
10 class Truck2 : public Vehicle
11 {
12
13 public:
14     Truck2(SDL_Renderer *rend, SDL_Texture *ast, SDL_Rect mov);
15     void draw(bool flag);
16 };

```

5.13 Unit.hpp

```
1 #pragma once
2 #include <SDL.h>
3
11 class Unit
12 {
13     SDL_Renderer *gRenderer;
14     SDL_Texture *assets;
15
16 public:
17     Unit(SDL_Renderer *, SDL_Texture *);
18     void draw(SDL_Rect srcRect, SDL_Rect moverRect);
19 };

```

5.14 Vehicle.hpp

```
1 #pragma once
2 #include "Unit.hpp"
3 using namespace std;
4
13 class Vehicle : public Unit
14 {
15
16 private:
17     SDL_Rect src, mover;
18     int speed;
19     bool changing_lane_down;
20     bool changing_lane_up;
21
22 public:
23     Vehicle(SDL_Renderer *rend, SDL_Texture *ast, SDL_Rect mov, SDL_Rect src, int s);
24     void draw(bool flag);

```

```
25     // friend bool Jamalo::hitRegistered(Vehicle v);
26     SDL_Rect getter() const;
27     void changeLaneUp();
28     void changeLaneDown();
29 };
```


Index

- animate
 - Digit, [13](#)
- Bike, [7](#)
 - Bike, [7](#)
 - draw, [9](#)
- Car, [9](#)
 - Car, [10](#)
 - draw, [10](#)
- changeLaneDown
 - Vehicle, [27](#)
- changeLaneUp
 - Vehicle, [27](#)
- createObjects
 - Jamalo, [15](#)
- Debris, [11](#)
 - Debris, [11](#)
- Digit, [12](#)
 - animate, [13](#)
 - Digit, [12](#)
 - draw, [13](#)
- draw
 - Bike, [9](#)
 - Car, [10](#)
 - Digit, [13](#)
 - RaceCar, [19](#)
 - Sheep, [20](#)
 - Truck, [23](#)
 - Truck2, [24](#)
 - Unit, [26](#)
 - Vehicle, [27](#)
- drawObjects
 - Jamalo, [15](#)
- Game, [13](#)
- getter
 - Sheep, [20](#)
 - Vehicle, [28](#)
- hitRegistered
 - Jamalo, [15](#), [16](#)
- isOver
 - Jamalo, [16](#)
- Jamalo, [14](#)
 - createObjects, [15](#)
 - drawObjects, [15](#)
 - hitRegistered, [15](#), [16](#)
 - isOver, [16](#)
 - Jamalo, [15](#)
 - move, [17](#)
 - radar, [17](#)
- Lives, [17](#)
- move
 - Jamalo, [17](#)
 - Sheep, [21](#)
- RaceCar, [18](#)
 - draw, [19](#)
 - RaceCar, [18](#)
- radar
 - Jamalo, [17](#)
- Sheep, [19](#)
 - draw, [20](#)
 - getter, [20](#)
 - move, [21](#)
 - Sheep, [20](#)
- Timer, [21](#)
- Truck, [22](#)
 - draw, [23](#)
 - Truck, [22](#)
- Truck2, [23](#)
 - draw, [24](#)
 - Truck2, [24](#)
- Unit, [25](#)
 - draw, [26](#)
 - Unit, [25](#)
- Vehicle, [26](#)
 - changeLaneDown, [27](#)
 - changeLaneUp, [27](#)
 - draw, [27](#)
 - getter, [28](#)
 - Vehicle, [27](#)