

# Welcome to Dirt Racing

1.0

Generated by Doxygen 1.7.6.1

Mon Sep 16 2013 03:23:12



# Contents

<b>1</b>	<b>Class Index</b>	<b>1</b>
1.1	Class List . . . . .	1
<b>2</b>	<b>Class Documentation</b>	<b>3</b>
2.1	_GLMgroup Struct Reference . . . . .	3
2.2	_GLMmaterial Struct Reference . . . . .	3
2.3	_GLMmodel Struct Reference . . . . .	4
2.4	_GLMnode Struct Reference . . . . .	4
2.5	_GLMpolygon Struct Reference . . . . .	5
2.6	_GLMtriangle Struct Reference . . . . .	5
2.7	bonusObject Class Reference . . . . .	5
2.7.1	Detailed Description . . . . .	6
2.7.2	Constructor & Destructor Documentation . . . . .	6
2.7.2.1	bonusObject . . . . .	6
2.8	Image Class Reference . . . . .	6
2.8.1	Detailed Description . . . . .	6
2.8.2	Constructor & Destructor Documentation . . . . .	7
2.8.2.1	Image . . . . .	7
2.9	object Class Reference . . . . .	7
2.9.1	Constructor & Destructor Documentation . . . . .	8
2.9.1.1	object . . . . .	8
2.9.1.2	object . . . . .	8
2.10	statObject Class Reference . . . . .	8
2.10.1	Detailed Description . . . . .	8
2.10.2	Constructor & Destructor Documentation . . . . .	9
2.10.2.1	statObject . . . . .	9

2.11 Terrain Class Reference . . . . .	9
2.12 treeObject Class Reference . . . . .	9
2.12.1 Detailed Description . . . . .	10
2.12.2 Constructor & Destructor Documentation . . . . .	10
2.12.2.1 treeObject . . . . .	10
2.13 vector Class Reference . . . . .	10

# Chapter 1

## Class Index

### 1.1 Class List

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

<a href="#">_GLMgroup</a>	3
<a href="#">_GLMmaterial</a>	3
<a href="#">_GLMmodel</a>	4
<a href="#">_GLMnode</a>	4
<a href="#">_GLMpolygon</a>	5
<a href="#">_GLMtriangle</a>	5
<a href="#">bonusObject</a>	5
<a href="#">Image</a>	6
<a href="#">object</a>	7
<a href="#">statObject</a>	8
<a href="#">Terrain</a>	9
<a href="#">treeObject</a>	9
<a href="#">vector</a>	10



## Chapter 2

# Class Documentation

### 2.1 `_GLMgroup` Struct Reference

#### Public Attributes

- char \* **name**
- GLuint **numtriangles**
- GLuint \* **triangles**
- GLuint **material**
- struct [\\_GLMgroup](#) \* **next**

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

- glm.h

### 2.2 `_GLMmaterial` Struct Reference

#### Public Attributes

- char \* **name**
- GLfloat **diffuse** [4]
- GLfloat **ambient** [4]
- GLfloat **specular** [4]
- GLfloat **emmissive** [4]
- GLfloat **shininess**

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

- glm.h

## 2.3 `_GLMmodel` Struct Reference

### Public Attributes

- char \* **pathname**
- char \* **mtllibname**
- GLfloat **translacao** [3]
- GLfloat **escala** [3]
- GLfloat **rotacao** [4]
- GLuint **numvertices**
- GLfloat \* **vertices**
- GLuint **numnormals**
- GLfloat \* **normals**
- GLuint **numtexcoords**
- GLfloat \* **texcoords**
- GLuint **numfacetnorms**
- GLfloat \* **facetnorms**
- GLuint **numtriangles**
- [GLMtriangle](#) \* **triangles**
- GLint **numpolygons**
- [GLMpolygon](#) \* **polygons**
- char \* **texture\_file**
- GLuint **nummaterials**
- [GLMmaterial](#) \* **materials**
- GLuint **numgroups**
- [GLMgroup](#) \* **groups**
- GLfloat **position** [3]

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

- glm.h

## 2.4 `_GLMnode` Struct Reference

### Public Attributes

- GLuint **index**
- GLboolean **averaged**
- struct [\\_GLMnode](#) \* **next**

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

- glm.cpp



## 2.5 `_GLMpolygon` Struct Reference

### Public Attributes

- GLuint **numvertices**
- GLuint **n**
- GLuint **t**
- GLuint \* **vindices**
- GLuint \* **nindices**
- GLuint \* **tindices**
- GLuint **findex**

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

- glm.h

## 2.6 `_GLMtriangle` Struct Reference

### Public Attributes

- GLuint **vindices** [3]
- GLuint **nindices** [3]
- GLuint **tindices** [3]
- GLuint **findex**

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

- glm.h

## 2.7 `bonusObject` Class Reference

```
#include <object.h>
```

### Public Member Functions

- [bonusObject](#) (float pos[][3], float cent[])

### Public Attributes

- float [positions](#) [4][3]  
*all positions should be filled only in counterclockwise direction*
- float **centre** [3]
- bool **beenhit**

## Static Public Attributes

- static `bonusObject** bonusObjArray` = new `bonusObject*`[100]
- static int `totalbonusObjects` = 0  
*initialises no. of bonus to 0*

### 2.7.1 Detailed Description

Class containing the bonus Objects

### 2.7.2 Constructor & Destructor Documentation

#### 2.7.2.1 `bonusObject::bonusObject ( float pos[][3], float cent[] )`

<COnstructor for adding bonus (float\_array position\_array,flaot centre)

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

- `object.h`
- `object.cpp`

## 2.8 Image Class Reference

```
#include <LoadBMP.h>
```

### Public Member Functions

- `Image` (int w, int h, unsigned char \*data)  
*Constructur Image(width\_of\_file,Height\_of\_file,Unsigned\_pointer\_data )*

### Public Attributes

- int **width**
- int **height**
- unsigned char \* **imagedata**

### 2.8.1 Detailed Description

[LoadBMP.h](#)

Conatins IMAGE class, reads Bitmap file into memory

Created on: Aug 7, 2013 Author: cs1120239

## 2.8.2 Constructor & Destructor Documentation

### 2.8.2.1 Image::Image ( int *w*, int *h*, unsigned char \* *data* )

Constructur Image(width\_of\_file,Height\_of\_file,Unsigned\_pointer\_data )

LoadBMP.cpp

Created on: Aug 6, 2013 Author: cs1120239

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

- LoadBMP.h
- LoadBMP.cpp

## 2.9 object Class Reference

### Public Member Functions

- [object](#) (float, float, float, float, float, float)  
*Constructor [object\(float x,float y,float z,float max\\_speed,float teta,float fi\)](#)*
- [object](#) ()
- [vector](#) [getVelocity](#) ()  
*retuns velocity of the object*

### Public Attributes

- float [teta](#)  
*Angle of Object with the Y axis in RAD.*
- float [fi](#)  
*Angle of projectin of the Object on XY plane with the X axis in RAD.*
- float [speed](#)  
*Speed of the Object.*
- float [accel](#)  
*Acceleraton of Object.*
- [vector](#) [position](#)  
*position of Object in vector form*
- [vector](#) [normal](#)  
*UPNormal of object.*
- float **MAXSPEED**

## 2.9.1 Constructor & Destructor Documentation

### 2.9.1.1 `object::object ( float x, float y, float z, float max, float yangle, float xzangle )`

Constructor `object(float x,float y,float z,float max_speed,float teta,float fi)`

< Constructor `object(float x,float y,float z,float max_speed,float teta,float fi)`

### 2.9.1.2 `object::object ( )`

Default constructor sets the initial values to teta=0.0f; speed=0.0f; fi=0.0f; accel=0.0f; position=0.0f,0.0f,0.0f); normal=(0.0f,1.0f,0.0f); MAXSPEED = 100.0f;

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

- `object.h`
- `object.cpp`

## 2.10 statObject Class Reference

```
#include <object.h>
```

### Public Member Functions

- `statObject` (float pos[][3], float cent[])

### Public Attributes

- float `positions` [4][3]  
*all positions should be filled only in counterclockwise direction*
- float `centre` [3]

### Static Public Attributes

- static `statObject` \*\* `ObjArray` = new `statObject`\*[100]
- static int `totalObjects` = 0  
*Initialises no. of obstacles to 0.*

### 2.10.1 Detailed Description

Class containing the static obstacles

## 2.10.2 Constructor & Destructor Documentation

### 2.10.2.1 statObject::statObject ( float pos[][3], float cent[] )

<COnstructor for adding obstacles (float\_array position\_array, float centre)

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

- object.h
- object.cpp

## 2.11 Terrain Class Reference

### Public Member Functions

- **Terrain** (int w, int l)
- void **setHeight** (int, float, int)
- float **getHeight** (int, int)
- int **getWidth** ()
- int **getLength** ()
- void **computenormals** ()
- **vector** **getNormal** (int, int)

### Public Attributes

- float \*\* **twodpointer**
- int **width**
- int **length**
- **vector** \*\* **normals**

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

- Terrain.h
- Terrain.cpp

## 2.12 treeObject Class Reference

```
#include <object.h>
```

### Public Member Functions

- **treeObject** (float pos[][3], float cent[])  
*Cleans up memory after game is finished.*

## Public Attributes

- float `positions` [4][3]  
*all positions should be filled only in counterclockwise direction*
- float `centre` [3]

## Static Public Attributes

- static `treeObject` \*\* `treeObjArray` = new `treeObject`\*[100]
- static int `totalTreeObjects` = 0  
*initilises no. of trees to 0*

### 2.12.1 Detailed Description

Class containing the trees in track

### 2.12.2 Constructor & Destructor Documentation

#### 2.12.2.1 `treeObject::treeObject ( float pos[][3], float cent[] )`

Cleans up memory after game is finished.

<COnstructor for adding trees (float\_array position\_array, float centre)

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

- `object.h`
- `object.cpp`

## 2.13 vector Class Reference

### Public Member Functions

- **vector** (float, float, float)
- **vector** (const `vector` &v)
- void **setCoords** (float, float, float)
- **vector operator+** (const `vector` &v) const
- **vector operator-** (const `vector` &v) const
- **vector operator+=** (const `vector` &v) const
- **vector operator-=** (const `vector` &v) const
- **vector & operator=** (const `vector` &v)
- **vector scalarMult** (float f)
- float **dotproduct** (const `vector` &v)
- **vector crossproduct** (const `vector` &v)
- float **magnitude** ()

- float **magnitudesquare** ()
- [vector](#) **normalize** ()

### Public Attributes

- float \* **comp**

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

- vector.h
- vector.cpp