

Reference Manual

Generated by Doxygen 1.8.13

Contents

1	Class Index	2
1.1	Class List	2
2	File Index	3
2.1	File List	3
3	Class Documentation	4
3.1	CannonBall Class Reference	4
3.1.1	Detailed Description	5
3.1.2	Constructor & Destructor Documentation	5
3.1.3	Member Function Documentation	7
3.1.4	Member Data Documentation	10
4	File Documentation	13
4.1	cannonball.cpp File Reference	13
4.1.1	Function Documentation	14
4.1.2	Variable Documentation	16
4.2	CannonBall.h File Reference	18
	Index	19

1 Class Index

1.1 Class List

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

[CannonBall](#)

4

2 File Index

2.1 File List

Here is a list of all files with brief descriptions:

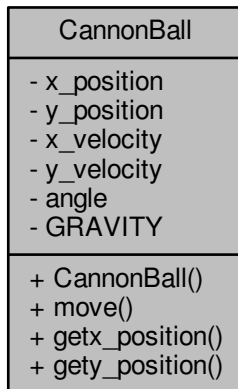
cannonball.cpp	13
CannonBall.h	18

3 Class Documentation

3.1 CannonBall Class Reference

```
#include <CannonBall.h>
```

Collaboration diagram for CannonBall:



Public Member Functions

- [CannonBall](#) (double x_posit, double velo, double ang)
- void [move](#) (double sec)
- double [getX_position](#) () const
- double [getY_position](#) () const

Private Attributes

- double [x_position](#)
- double [y_position](#)
- const double [x_velocity](#)
- const double [y_velocity](#)
- const double [angle](#)
- const double [GRAVITY](#)

3.1.1 Detailed Description

A cannonball simulator with a primary function to compute the x and y position based on the initial velocity, angle, and height and the time.

Definition at line 6 of file CannonBall.h.

3.1.2 Constructor & Destructor Documentation

3.1.2.1 CannonBall()

```
CannonBall::CannonBall (  
    double x_posit,  
    double velo,  
    double ang )
```

Constructs new [CannonBall](#) with GRAVITY initialized and sets *velo* to *x_velocity* and *y_velocity*

Parameters

<i>x_posit</i>	the initial <i>x_position</i> of the cannonball
<i>velo</i>	the initial velocity
<i>ang</i>	the initial angle from which the cannonball is shot from

Cannonball launching program with bug targer



Figure 1 Bug



Figure 2 CannonBall

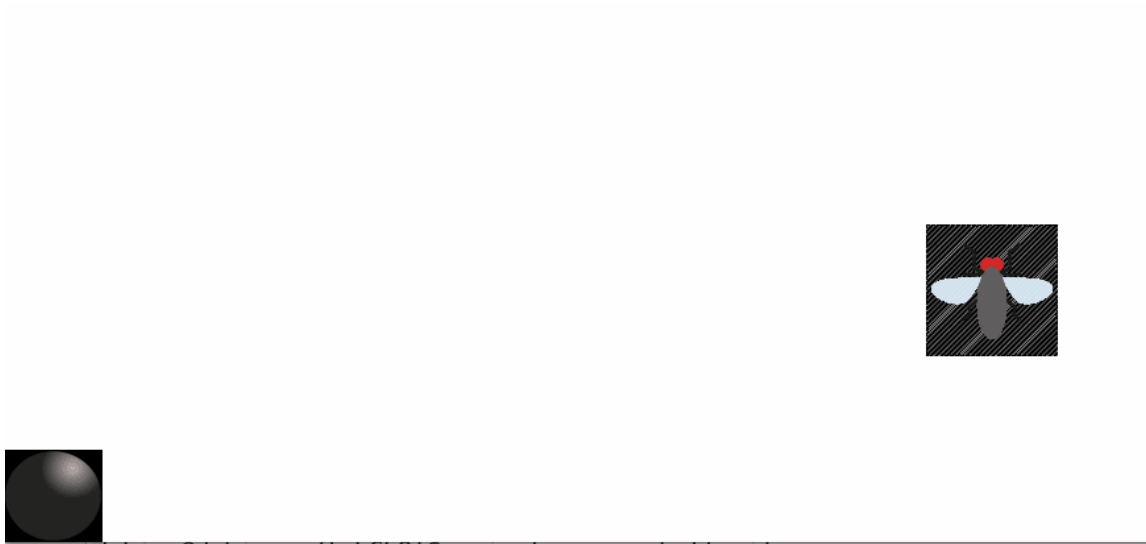


Figure 3 Gif of cannonball

Definition at line 13 of file cannonball.cpp.

```
14     :angle(ang), GRAVITY(9.81), x_position(x_posit),  
15     x_velocity(velo), y_position(0), y_velocity(velo)  
16 {}
```

3.1.3 Member Function Documentation

3.1.3.1 getx_position()

```
double CannonBall::getx_position ( ) const
```

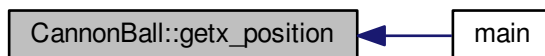
Returns

returns the current x_position

Definition at line 22 of file cannonball.cpp.

```
23 {  
24     return x_position;  
25 }
```

Here is the caller graph for this function:



3.1.3.2 gety_position()

```
double CannonBall::gety_position ( ) const
```

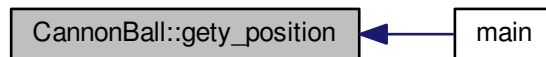
Returns

returns the current y_position

Definition at line 26 of file cannonball.cpp.

```
27 {  
28     return y_position;  
29 }
```

Here is the caller graph for this function:



3.1.3.3 move()

```
void CannonBall::move (  
    double sec )
```

Function which calculates the x and y position of the ball based on the time sent as a parameter and the equation with the initalized velocity, x_position, and angle

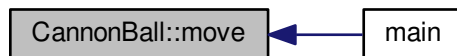
Parameters

<code>sec</code>	the seconds at which the x and y position will be calculated
------------------	--

Definition at line 17 of file cannonball.cpp.

```
18 {  
19     y_position = y_velocity * sin(angle) * sec -  
    GRAVITY * pow(sec, 2)/2;  
20     x_position = x_velocity * cos(angle) * sec;  
21 }
```

Here is the caller graph for this function:



3.1.4 Member Data Documentation

3.1.4.1 angle

```
const double CannonBall::angle [private]
```

Definition at line 40 of file CannonBall.h.

3.1.4.2 GRAVITY

```
const double CannonBall::GRAVITY [private]
```

Definition at line 41 of file CannonBall.h.

3.1.4.3 x_position

```
double CannonBall::x_position [private]
```

Definition at line 36 of file CannonBall.h.

3.1.4.4 x_velocity

```
const double CannonBall::x_velocity [private]
```

Definition at line 38 of file CannonBall.h.

3.1.4.5 y_position

```
double CannonBall::y_position [private]
```

Definition at line 37 of file CannonBall.h.

3.1.4.6 y_velocity

```
const double CannonBall::y_velocity [private]
```

Definition at line 39 of file CannonBall.h.

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

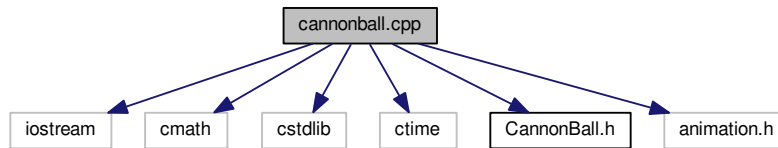
- [CannonBall.h](#)
- [cannonball.cpp](#)

4 File Documentation

4.1 cannonball.cpp File Reference

```
#include <iostream>
#include <cmath>
#include <cstdlib>
#include <ctime>
#include "CannonBall.h"
#include "animation.h"
```

Include dependency graph for cannonball.cpp:



Functions

- int `main` ()

Variables

- const double `DELTA_T` = 1
- const double `PI` = 3.1415
- const int `HEIGHT` = 500
- const int `WIDTH` = 800
- const double `BUG_WIDTH` = 700

4.1.1 Function Documentation

4.1.1.1 main()

```
int main ( )
```

Definition at line 38 of file cannonball.cpp.

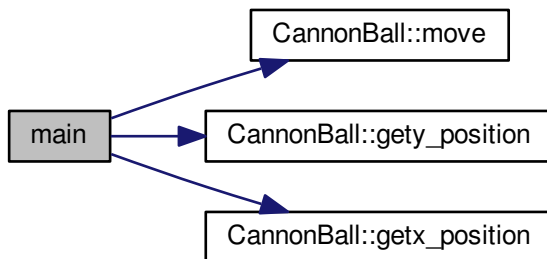
```
39 {
40     Picture pic("cannonball.png");
41     Picture bug("bugsy2.png");
42     //creating random height for the bug
43     unsigned int seed = time(0);
44     srand(seed);
45
46     double randbug_height = (rand() % (HEIGHT - bug.height() + 1)) + bug.height();
47     double velocity, angle, start_height;
48     //creating animation
49     Animation anim("cannonball.gif",
50         WIDTH + pic.width(), HEIGHT + pic.height());
51     //entering data
52     std::cout << "Enter angle of shot (in degrees):" << std::endl;
53     std::cin >> angle;
54     angle *= PI / 180.0;
55     std::cout << "Enter velocity" << std::endl;
56     std::cin >> velocity;
57     std::cout << "Enter intital height" << std::endl;
58     std::cin >> start_height;
59 }
```

```
60     double iHeight, iWidth;
61     CannonBall ball(start_height, velocity, angle);
62     bool ground = true, hit = false;
63     double second = 0;
64     anim.add(pic, 0, HEIGHT);
65     anim.add(bug, BUG_WIDTH, HEIGHT - randbug_height);
66     anim.frame();
67     while(ground)
68     {
69         second += DELTA_T;
70         ball.move(second);
71         iHeight = ball.gety_position();
72         iWidth = ball.getx_position();
73         anim.add(pic, iWidth, HEIGHT - iHeight);
74         anim.add(bug, BUG_WIDTH, HEIGHT - randbug_height);
75         //checking if CannonBall intersects with bug
76         if ((iWidth >= BUG_WIDTH - pic.width() &&
77             iWidth <= BUG_WIDTH + bug.width()) &&
78             ((iHeight >= randbug_height &&
79              iHeight <= randbug_height + bug.height()) ||
80              (iHeight + pic.height() >= randbug_height &&
81               iHeight + pic.height() <= randbug_height + bug.height()))) && !hit)
82         {
83             std::cout << "Splat goes the bug\n";
84             hit = true;
85         }
86         anim.frame();
87         // exits program if ball touches the ground or exits off screen
88         if (iHeight <= 0 || iWidth >= WIDTH)
89             ground = false;
90     }
```



```
91     if (!hit)
92         std::cout << "You missed :(. Try Again.\n";
93     anim.close();
94     //can use system("animate animation.gif") + loop for use to type in right ini
95     //maybe have bug at set x_position and rand() y position
96     return 0;
97 }
```

Here is the call graph for this function:



4.1.2 Variable Documentation

4.1.2.1 BUG_WIDTH

```
const double BUG_WIDTH = 700
```

Definition at line 36 of file cannonball.cpp.

4.1.2.2 DELTA_T

```
const double DELTA_T = 1
```

Definition at line 30 of file cannonball.cpp.

4.1.2.3 HEIGHT

```
const int HEIGHT = 500
```

Definition at line 33 of file cannonball.cpp.

4.1.2.4 PI

```
const double PI = 3.1415
```

Definition at line 31 of file cannonball.cpp.

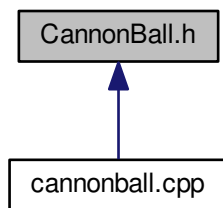
4.1.2.5 WIDTH

```
const int WIDTH = 800
```

Definition at line 34 of file cannonball.cpp.

4.2 CannonBall.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class [CannonBall](#)

Index

angle

CannonBall, [10](#)

BUG_WIDTH

cannonball.cpp, [16](#)

CannonBall, [4](#)

angle, [10](#)

CannonBall, [5](#)

GRAVITY, [11](#)

getx_position, [7](#)

gety_position, [8](#)

move, [9](#)

x_position, [11](#)

x_velocity, [11](#)

y_position, [11](#)

y_velocity, [12](#)

CannonBall.h, [18](#)

cannonball.cpp, [13](#)

BUG_WIDTH, [16](#)

DELTA_T, [17](#)

HEIGHT, [17](#)

main, [14](#)

PI, [17](#)

WIDTH, [17](#)

DELTA_T

cannonball.cpp, [17](#)

GRAVITY

CannonBall, [11](#)

getx_position

CannonBall, [7](#)

gety_position

CannonBall, [8](#)

HEIGHT

cannonball.cpp, [17](#)

main

cannonball.cpp, [14](#)

move

CannonBall, [9](#)

PI

cannonball.cpp, [17](#)

WIDTH

cannonball.cpp, [17](#)

x_position

CannonBall, [11](#)

x_velocity

CannonBall, [11](#)

y_position

CannonBall, [11](#)

y_velocity

CannonBall, [12](#)