Graphics Engine

1.0

Generated by Doxygen 1.8.14

Contents

Chapter 1

3802ICT

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

Class Documentation

3.1 colour Struct Reference

```
#include <structs.h>
```

Public Attributes

- double **R**
- double **G**
- double **B**

3.1.1 Detailed Description

Three doubles representing values for red, green and blue respectively

Parameters

R	red value
G	green value
B	blue value

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

- structs.h

3.2 matrix Class Reference

Public Member Functions

- matrix (int num_rows, int num_cols)
- matrix ()

- void set_row (int row_number, std::vector< double > row)
- void set_col (int col_number, std::vector< double > col)
- void set_val (int row_number, int col_number, double val)
- void add_val (int row_number, int col_number, double val)
- double get_val (int row_number, int col_number)
- int get_rows ()
- int get_cols ()
- void print ()
- matrix multiply (matrix other_matrix)
- void set_up_transformation ()

3.2.1 Constructor & Destructor Documentation

3.2.1.1 matrix() [1/2]

```
matrix::matrix (
            int rows,
            int cols )
```

Constructor for a matrix when given and number of rows and columns

Parameters

rows	an integer representing the number of rows the matrix should have
cols	an integer representing the number of columns the matrix should have

3.2.1.2 matrix() [2/2]

```
matrix::matrix ( )
```

Constructor for a matrix if not given a size. Makes a 4 by 4 matrix

3.2.2 Member Function Documentation

3.2.2.1 add_val()

```
void matrix::add_val (
            int row_number,
            int col_number,
            double val )
```

Adds a value at the given row and column with the given value

Parameters

row_number	integer representing which row to replace
col_number	integer representing which column to replace
val	double representing the value to add in the matrix

3.2.2.2 get_cols()

```
int matrix::get_cols ( )
```

Returns the number of columns the matrix has

3.2.2.3 get_rows()

```
int matrix::get_rows ( )
```

Returns the number of rows the matrix has

3.2.2.4 get_val()

```
double matrix::get_val (
             int row_number,
             int col_number )
```

Returns a value at the given row and column

Parameters

row_number	integer representing which row to replace
col_number	integer representing which column to replace

3.2.2.5 multiply()

```
matrix matrix::multiply (
             matrix other_matrix )
```

Performs matrix dot multiplication between this matrix and a given matrix

Parameters

other_matrix	the matrix to multiply with

3.2.2.6 print()

```
void matrix::print ( )
```

Prints out the matrix to stdout in a formatted way

3.2.2.7 set_col()

```
void matrix::set_col (
             int col_number,
             std::vector< double > col )
```

Replaces a col in the matrix with the given vector

Parameters

col_number	integer representing which column to replace
col	vector of doubles representing the new col values

3.2.2.8 set_row()

```
void matrix::set_row (
             int row_number,
             std::vector< double > row )
```

Replaces a row in the matrix with the given vector

Parameters

row_number	integer representing which row to replace
row	vector of doubles representing the new row values

3.2.2.9 set_up_transformation()

```
void matrix::set_up_transformation ( )
```

Sets up the matrix as a blank transformation matrix

3.2.2.10 set_val()

```
void matrix::set_val (
             int row_number,
             int col_number,
             double val )
```

Replaces a value at the given row and column with the given value

Parameters

row_number	integer representing which row to replace
col_number	integer representing which column to replace
val	double representing the value to use in the matrix

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

- matrix.h
- matrix.cpp

3.3 point Struct Reference

```
#include <structs.h>
```

Public Attributes

- int **x**
- int **y**

3.3.1 Detailed Description

Two integers representing a pixel on the screen

Parameters

x	x co-ordinate of pixel
y	y co-ordinate of pixel

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

- structs.h

3.4 Polygon Class Reference

Public Member Functions

- Polygon (std::vector< point > points, point coordinates)
- Polygon (std::vector< point > points)
- Polygon ()
- void change_points (std::vector< point > points)
- void set_colour (colour RGB)
- void draw ()
- void scale (int x_scale, int y_scale)
- void rotate (double angle)
- void additive_rotate (double angle)
- void translate (double x_offset, double y_offset)
- void additive_translate (double x_offset, double y_offset)
- void save_transformation ()
- void undo_transformation ()
- point find_top_left_point ()
- point find_bottom_right_point ()

3.4.1 Constructor & Destructor Documentation

3.4.1.1 Polygon() [1/3]

```
Polygon::Polygon (
            std::vector< point > points,
            point coordinates )
```

Constructor for a polygon when give the points and a starting point

Parameters

points	vector of points representing the points making up the polygon
coordinates	a point representing where the polygon should be drawn on the screen

3.4.1.2 Polygon() [2/3]

```
Polygon::Polygon (
            std::vector< point > points )
```

Constructor for a polygon when give the points. Draws the polygon at 0, 0

Parameters

points	vector of points representing the points making up the polygon

3.4.1.3 Polygon() [3/3]

```
Polygon::Polygon ( )
```

Constructor for a blank Polygon. Defines one point at 0, 0 and draws it at 0, 0

3.4.2 Member Function Documentation

3.4.2.1 additive_rotate()

```
void Polygon::additive_rotate (
            double angle )
```

Rotates the Polygon by the given angle (Adds to rotation)

Parameters

angle	double representing the angle in degrees

3.4.2.2 additive_translate()

```
void Polygon::additive_translate (
            double x_offset,
            double y_offset )
```

Translates the Polygon by the given dimensions (Adds to translation)

Parameters

x_scale	integer representing what to move the x value of the Polygon by
y_scale	integer representing what to move the y value of the Polygon by

3.4.2.3 change_points()

```
void Polygon::change_points (
            std::vector< point > points )
```

Replaces the vector the Polygon's points with the given vector

Parameters

points	vector of points representing the points making up the polygon

3.4.2.4 draw()

```
void Polygon::draw ( )
```

Draws the Polygon on the screen

3.4.2.5 find_bottom_right_point()

```
point Polygon::find_bottom_right_point ( )
```

Finds the highest x value and the smallest y value

3.4.2.6 find_top_left_point()

```
point Polygon::find_top_left_point ( )
```

Finds the smallest x value and the highest y value

3.4.2.7 rotate()

```
void Polygon::rotate (
             double angle )
```

Rotates the Polygon to the given angle

Parameters

angle	double representing the angle in degrees

3.4.2.8 save_transformation()

```
void Polygon::save_transformation ( )
```

Saves the currently used matrix to a stack

3.4.2.9 scale()

```
void Polygon::scale (
             int x_scale,
             int y_scale )
```

Scales the Polygon by the given dimensions

Parameters

x_scale	integer representing what to multiply the x scale of the Polygon by
y_scale	integer representing what to multiply the y scale of the Polygon by

3.4.2.10 set_colour()

```
void Polygon::set_colour (
             colour RGB )
```

Replaces the colour the Polygon is filled with with the given colour

Parameters

RGB	struct representing the 3 double values for R, G and B

3.4.2.11 translate()

```
void Polygon::translate (
             double x_offset,
             double y_offset )
```

Translates the Polygon to the given dimensions

Parameters

x_scale	integer representing where to move the x value of the Polygon to
y_scale	integer representing where to move the y value of the Polygon to

3.4.2.12 undo_transformation()

```
void Polygon::undo_transformation ( )
```

Reverts the transformation matrix to the last saved matrix

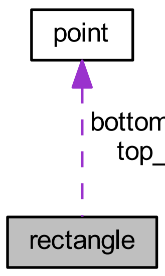
The documentation for this class was generated from the following files:

- polygon.h
- polygon.cpp

3.5 rectangle Struct Reference

```
#include <structs.h>
```

Collaboration diagram for rectangle:



Public Attributes

- point **top_left**
- point **bottom_right**

3.5.1 Detailed Description

Two points representing the top left and bottom right corners to form a rectangle

Parameters

top_left	point where the top left of the rectangle is located
bottom_right	point where the bottom right of the recatngle is located

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

- structs.h

3.6 Text Class Reference

Public Member Functions

- Text (std::string text_to_display, point bottom_left_position, colour RBG)

- void draw ()
- void update_text (std::string text_to_display)

3.6.1 Constructor & Destructor Documentation

3.6.1.1 Text()

```
Text::Text (
            std::string text_to_display,
            point bottom_left_position,
            colour RGB )
```

Constructor for text. Currently only supports numbers

Parameters

text_to_display	a string representing what text to display
bottom_left_position	a point representing the bottom left position to draw the text
RGB	a struct representing the RGB values of the colour to draw the text as

3.6.2 Member Function Documentation

3.6.2.1 draw()

```
void Text::draw ( )
```

Draws the numbers to the screen

3.6.2.2 update_text()

```
void Text::update_text (
            std::string text_to_display )
```

Replaces the text that will be replaced

Parameters

update_text	string representing the new text to display

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

- text.h
- text.cpp

3.7 velocity Struct Reference

```
#include <structs.h>
```

Public Attributes

- double **x**
- double **y**
- double **speed**

3.7.1 Detailed Description

Three doubles representing velocity as both x and y components and the magnitude of a vector

Parameters

x	x component of the velocity
y	y component of the velocity
speed	the magnitude of the velocity vector

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

- structs.h

Index