UFCFGL-30-1

Generated by Doxygen 1.8.14

Contents

1 UFCFGL-30-1 Programming in C++ Support Libraries **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 uwe::EntranceInfo Class Reference 7

4.1.1 Constructor & Destructor Documentation 7

4.1.1.1 EntranceInfo() 7

4.1.2 Member Function Documentation 8

4.1.2.1 getDirection() 8

4.1.2.2 getTo() 8

4.1.2.3 isLocked() 8

4.2 uwe::ItemInfo Class Reference 9

4.2.1 Constructor & Destructor Documentation 9

4.2.1.1 ItemInfo() 9

4.2.2 Member Function Documentation 9

4.2.2.1 getHtml() 9

4.2.2.2 getName() 10

4.3 uwe::RoomInfo Class Reference 10

4.3.1 Constructor & Destructor Documentation 10

4.3.1.1 RoomInfo() 10

4.3.2 Member Function Documentation . . . 11
4.3.2.1 getDescription() . . . 11
4.3.2.2 getEntrances() . . . 11
4.3.2.3 getItems() . . . 11
4.3.2.4 getName() . . . 12
4.3.2.5 getZombieCount() . . . 12
4.4 uwe::WorldLoader Class Reference . . . 12
4.4.1 Member Function Documentation . . . 12
4.4.1.1 getEnd() . . . 13
4.4.1.2 getInfo() . . . 13
4.4.1.3 getInventoryHtml() . . . 13
4.4.1.4 getItems() . . . 13
4.4.1.5 getRooms() . . . 14
4.4.1.6 getStart() . . . 14
4.4.1.7 getStartHtml() . . . 14
4.5 uwe::ZombieBot Class Reference . . . 14
4.5.1 Member Function Documentation . . . 15
4.5.1.1 begin() . . . 15
4.5.1.2 currentScore() . . . 15
4.5.1.3 disableTimer() . . . 15
4.5.1.4 enableTimer() . . . 15
4.5.1.5 processCmd() . . . 16
4.5.1.6 shouldQuit() . . . 17
4.6 uwe::ZombieServer Class Reference . . . 17
4.6.1 Constructor & Destructor Documentation . . . 17
4.6.1.1 ZombieServer() . . . 17

5 File Documentation 19

5.1 include/graphics.h File Reference . . . 19

5.1.1 Detailed Description . . . 20

5.1.2 Function Documentation . . . 20

5.1.2.1 drawCircle() . . . 20

5.1.2.2 drawFilledCircle() . . . 21

5.1.2.3 drawFilledRect() . . . 21

5.1.2.4 drawImageAt() . . . 21

5.1.2.5 drawLine() . . . 22

5.1.2.6 drawRect() . . . 22

5.1.2.7 drawText() . . . 23

5.1.2.8 fillAll() [1/2] . . . 23

5.1.2.9 fillAll() [2/2] . . . 23

5.1.2.10 getKeyCode() . . . 25

5.1.2.11 getTextCharacter() . . . 25

5.1.2.12 getWindowHeight() . . . 25

5.1.2.13 getWindowWidth() . . . 26

5.1.2.14 initialiseGraphics() . . . 26

5.1.2.15 loadImage() . . . 26

5.1.2.16 loop() . . . 27

5.1.2.17 setColour() [1/2] . . . 27

5.1.2.18 setColour() [2/2] . . . 27

5.1.2.19 setFontSize() . . . 28

5.2 include/ufcfgl-30-1.h File Reference . . . 28

5.2.1 Detailed Description . . . 28

5.2.2 Function Documentation . . . 29

5.2.2.1 getch() . . . 29

5.2.2.2 getche() . . . 29

5.2.2.3 getInt() [1/2] . . . 29

5.2.2.4 getInt() [2/2] . . . 29

5.2.2.5 getString() . . . 30

5.2.2.6 pause() . . . 30

5.2.2.7 readFile() . . . 30

5.2.2.8 str2int() . . . 31

Index 33

Chapter 1

UFCFGL-30-1 Programming in C++ Support Libraries

Introduction

To aid the development of programs as part of UFCFGL-30-1 a small selection of libraries are provided to be used in your programs.

ufcfgl-30-1.h

Utility library, providing functions to input data from the command line and files, for example. The documentation for the library is in ufcfgl-30-1.h.

To use the library in your programs add the line:

```
#include <ufcfgl-30-1.h>
```

Additionally you will need to link the library, this can be achieved by modifying your programs Makefile and adding:

```
$(LIBDIR)/libufcfgl-30-1.a
```

to the LIBS variable.

graphics.h

Provides a simple 2D graphics library, documentation is in graphics.h. The library provides the ability to open a window and draw shapes and lines, draw text, and handle keyboard input.

To use the graphics library you must add the line:

```
#include <graphics.h>
```

to your source code.

Additionally you will need to link the library, this can be achieved by modifying your programs Makefile and adding:

```
$(LIBDIR)/libgraphics.a $(JUCELIB)
```

to the LIBS variable. The variable `JUCELIB` references the JUCE library, which is used by the graphics library underhood.

Zombie Assignment Library

In the 2nd half of the course you will asked to complete an assignment. This library can be used to support that work.

The library is broken out into the following files (click classes in the main menu to see the corresponding documentation):

- zombies/EntranceInfo.h
- zombies/ItemInfo.h
- zombies/RoomInfo.h
- zombies/WorldLoader.h
- zombies/ZombieBot.h
- zombies/ZombieServer.h

Additionally you will need to link the library, this can be achieved by modifying your programs Makefile and adding:

```
$(LIBDIR)/$(LIBDIR)/libzombies.a
```

to the LIBS variable.

Chapter 2

Class Index

2.1 Class List

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

uwe::EntranceInfo 7
uwe::ItemInfo 9
uwe::RoomInfo 10
uwe::WorldLoader 12
uwe::ZombieBot 14
uwe::ZombieServer 17

Chapter 3

File Index

3.1 File List

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

include/graphics.h
An easy-to-use library for drawing geometric shapes, such as points, lines, and circles. Support for displaying text and images, is also provided . . . 19
include/ufcfgl-30-1.h
Support library for ufcfgk-30-1 . . . 28
include/zombies/**EntranceInfo.h** . . . **??**
include/zombies/**ItemInfo.h** . . . **??**
include/zombies/**RoomInfo.h** . . . **??**
include/zombies/**WorldLoader.h** . . . **??**
include/zombies/**ZombieBot.h** . . . **??**
include/zombies/**ZombieServer.h** . . . **??**

Chapter 4

Class Documentation

4.1 uwe::EntranceInfo Class Reference

Public Member Functions

- EntranceInfo (std::string dir, std::string to, bool locked)

  constructor to create info about an entrance
- std::string getDirection () const

  direction entrance is reachable by moving in this direction
- std::string getTo () const

  room that is reached, when moving through entrance
- bool isLocked () const

  is room locked

4.1.1 Constructor & Destructor Documentation

4.1.1.1 EntranceInfo()

```
uwe::EntranceInfo::EntranceInfo (
            std::string dir,
            std::string to,
            bool locked )
```

constructor to create info about an entrance

Parameters

dir	direction entrance leads to
to	room extrance leads to
locked	is the entrance locked, i.e. needs a key to open

4.1.2 Member Function Documentation

4.1.2.1 getDirection()

```
std::string uwe::EntranceInfo::getDirection ( ) const  [inline]
```

direction entrance is reachable by moving in this direction

Returns

direction

4.1.2.2 getTo()

```
std::string uwe::EntranceInfo::getTo ( ) const  [inline]
```

room that is reached, when moving through entrance

Returns

room

4.1.2.3 isLocked()

```
bool uwe::EntranceInfo::isLocked ( ) const  [inline]
```

is room locked

Returns

true if tool is locked, otherwise entrance is open

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

- include/zombies/EntranceInfo.h

4.2 uwe::ItemInfo Class Reference

Public Member Functions

- ItemInfo (std::string name, std::string html)
  constructor to create an item
- std::string getName ()
  name of item
- std::string getHtml ()
  html to display item in client

4.2.1 Constructor & Destructor Documentation

4.2.1.1 ItemInfo()

```
uwe::ItemInfo::ItemInfo (
            std::string name,
            std::string html )
```

constructor to create an item

Parameters

name	of item
html	for displaying item in client

4.2.2 Member Function Documentation

4.2.2.1 getHtml()

```
std::string uwe::ItemInfo::getHtml ( )  [inline]
```

html to display item in client

Returns

HTML

4.2.2.2 getName()

```
std::string uwe::ItemInfo::getName ( )  [inline]
```

name of item

Returns

name

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

- include/zombies/ItemInfo.h

4.3 uwe::RoomInfo Class Reference

Public Member Functions

- RoomInfo (std::string name, std::string desc, std::vector< EntranceInfo > entrances, std::vector< std::string > items, unsigned int zombieCount)
  constructor
- std::string getName () const
  room name
- std::string getDescription () const
  room description
- std::vector< EntranceInfo > getEntrances ()
  room entrances
- std::vector< std::string > getItems ()
  rooms items
- unsigned int getZombieCount ()
  rooms zombie count

4.3.1 Constructor & Destructor Documentation

4.3.1.1 RoomInfo()

```
uwe::RoomInfo::RoomInfo (
            std::string name,
            std::string desc,
            std::vector< EntranceInfo > entrances,
            std::vector< std::string > items,
            unsigned int zombieCount )
```

constructor

Parameters

name	room's name
desc	room's description
entrances	list of room's entrances
items	list of room's items
zombieCount	number of zombies in room

4.3.2 Member Function Documentation

4.3.2.1 getDescription()

```
std::string uwe::RoomInfo::getDescription ( ) const  [inline]
```

room description

Returns

description

4.3.2.2 getEntrances()

```
std::vector<EntranceInfo> uwe::RoomInfo::getEntrances ( )  [inline]
```

room entrances

Returns

list of entrances

4.3.2.3 getItems()

```
std::vector<std::string> uwe::RoomInfo::getItems ( )  [inline]
```

rooms items

Returns

list of items

4.3.2.4 getName()

```
std::string uwe::RoomInfo::getName ( ) const  [inline]
```

room name

Returns

name

4.3.2.5 getZombieCount()

```
unsigned int uwe::RoomInfo::getZombieCount ( )  [inline]
```

rooms zombie count

Returns

number of zombies in room

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

- include/zombies/RoomInfo.h

4.4 uwe::WorldLoader Class Reference

Public Member Functions

- **WorldLoader** (std::string world)
- std::string getStart () const
  starting room
- std::string getEnd () const
  end room
- std::string getInfo () const
  game info
- std::string getInventoryHtml () const
  HTML for displaying with player's inventory.
- std::string getStartHtml () const
  HTML to display when client makes connection to server.
- std::vector< ItemInfo > getItems () const
  items contained within the world
- std::vector< RoomInfo > getRooms () const
  items contained within the world

4.4.1 Member Function Documentation

4.4.1.1 getEnd()

```
std::string uwe::WorldLoader::getEnd ( ) const  [inline]
```

end room

Returns

room

4.4.1.2 getInfo()

```
std::string uwe::WorldLoader::getInfo ( ) const  [inline]
```

game info

Returns

info

4.4.1.3 getInventoryHtml()

```
std::string uwe::WorldLoader::getInventoryHtml ( ) const  [inline]
```

HTML for displaying with player's inventory.

Returns

HTML for inventory image

4.4.1.4 getItems()

```
std::vector<ItemInfo> uwe::WorldLoader::getItems ( ) const  [inline]
```

items contained within the world

Returns

list of items in world

4.4.1.5 getRooms()

```
std::vector<RoomInfo> uwe::WorldLoader::getRooms ( ) const  [inline]
```

items contained within the world

Returns

list of items in world

4.4.1.6 getStart()

```
std::string uwe::WorldLoader::getStart ( ) const  [inline]
```

starting room

Returns

room

4.4.1.7 getStartHtml()

```
std::string uwe::WorldLoader::getStartHtml ( ) const  [inline]
```

HTML to display when client makes connection to server.

Returns

html to display

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

- include/zombies/WorldLoader.h

4.5 uwe::ZombieBot Class Reference

Public Member Functions

- virtual bool shouldQuit ()=0
  should game quit
- virtual std::string begin ()=0
  text to be displayed at beginning of game
- virtual int currentScore ()=0
  compute current score
- virtual bool enableTimer ()=0
  should zombie timer be enabled
- virtual bool disableTimer ()=0
  should zombie timer be disabled
- virtual std::vector< std::string > processCmd (std::string cmd)=0
  process player commands.

4.5.1 Member Function Documentation

4.5.1.1 begin()

```
virtual std::string uwe::ZombieBot::begin ( )  [pure virtual]
```

text to be displayed at beginning of game

Returns

text to be displayed

4.5.1.2 currentScore()

```
virtual int uwe::ZombieBot::currentScore ( )  [pure virtual]
```

compute current score

Returns

current score

4.5.1.3 disableTimer()

```
virtual bool uwe::ZombieBot::disableTimer ( )  [pure virtual]
```

should zombie timer be disabled

Returns

true if timer should be disabled, otherwise false

4.5.1.4 enableTimer()

```
virtual bool uwe::ZombieBot::enableTimer ( )  [pure virtual]
```

should zombie timer be enabled

Returns

true if timer should be enabled, otherwise false

4.5.1.5 processCmd()

```
virtual std::vector<std::string> uwe::ZombieBot::processCmd (
            std::string cmd )  [pure virtual]
```

process player commands.

The set of commands are as follows:

- info - return the info string for the world.
- look - return the HTML representing the view of the current room. this should include entrances, items, and so on.
- move dir - try and move in the direction dir. if there is an extrance in that direction, then if it is not locked, move through to the connected room, return message about new room. if can't move, because no entrance in that direction, or entrance is locked, then return a message to that effect.
- one entry to a room if it contains zombies, then the next time the method enableTimer() is called it should return true, just the one time. This will cause the client to start the Zombie kill timer event, displaying a Zombie and timer, to show the player that they will die if the zombie(s) is not killed in time.
- pickup item - pick up item from room, if item exits then this should cause it to be removed from room and get added to players inventory, and return message to say picked up. if it does not exist, then return message to that effect.
  The users score should be increased by one, each time an item is picked up.
- kill - if the room contains at least one zombie and players inventory contains either a Daisy or Chainsaw, then one should be killed (i.e. zombie count drops by one), return a message to inform the user that a zombie has died. if kill is succcessful, then the player's inventory should have one less Daisy or Chainsaw.
  if no zombies, then nothing happens.
- if the zombie timer was running, i.e. a call to enableTimer() had returned true for the current room, then if the zombie count is now zero, then disableTimer() should return true, one time, to tell the client to stop the zombie timer.
- drop item - drop an item into room from inventory. if the item is not in inventory, then nothing happens, otherwise it should be removed, when added to room. return a message to say item dropped.
- timerexpired - the client's zombie timer expired before all zombies in the current room were killed and so the player is dead. shouldQuit() should now return true.
- quit - shouldQuit() should now return true.
- inventory - return HTML to display the players current inventory.
- anything else - return HTML saying "That's not a verb I recognise."

Parameters

cmd	to be processed

Returns

output to be displayed. each element in the list returned is in HTML and will be output in the clients display window

4.5.1.6 shouldQuit()

```
virtual bool uwe::ZombieBot::shouldQuit ( )  [pure virtual]
```

should game quit

Returns

return true if exit program, otherwise false

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

- include/zombies/ZombieBot.h

4.6 uwe::ZombieServer Class Reference

Public Member Functions

- ZombieServer (uint16_t port, ZombieBot &zbot)
  constructor
- void run ()
  run server, loops until user quits or instance deleted

4.6.1 Constructor & Destructor Documentation

4.6.1.1 ZombieServer()

```
uwe::ZombieServer::ZombieServer (
            uint16_t port,
            ZombieBot & zbot )
```

constructor

Parameters

port	to use for server

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

- include/zombies/ZombieServer.h

Chapter 5

File Documentation

5.1 include/graphics.h File Reference

An easy-to-use library for drawing geometric shapes, such as points, lines, and circles. Support for displaying text and images, is also provided.

```
#include <string>
#include <limits.h>
#include <stdlib.h>
#include "graphics_internal.h"
```

Typedefs

- typedef std::function< void(Graphics &)> **uwe::graphics_func**
- typedef juce::KeyPress **uwe::keyPress**

Functions

- void uwe::initialiseGraphics (int windowWidth=internal::WINDOW_WIDTH_DEFAULT, int window↩ Height=internal::WINDOW_HEIGHT_DEFAULT)

  setup graphics context.
- void uwe::shutDown (void)

  close the current graphics context
- template<class F >
  void uwe::loop (F paint_callback, internal::keypress_func keypress_callback=internal::keypress_↩ func(internal::default_keypress_callback))

  loops while application window is active, calling paint callback to draw to the component's window.
- char uwe::getTextCharacter (KeyPress keyPress)

  convert keyPress to a character
- int uwe::getKeyCode (KeyPress keyPress)

  convert keyPress to key scan code
- void uwe::drawLine (float start_x, float start_y, float end_x, float end_y)

  draws a line on the current graphics context
- void uwe::drawRect (float x, float y, float width, float height, float lineThickness=1)

  draws a rectangle on the current graphics context.

- void uwe::drawFilledRect (float x, float y, float width, float height)

  draws a rectangle on the current graphics context, which is filled with the current pen colour.
- void uwe::drawCircle (float x, float y, float radius, float lineThickness=1)

  draws a circle on the current graphics context.
- void uwe::drawFilledCircle (float x, float y, int radius)

  draws a filled circle on the current graphics context, using the current pen colour.
- void uwe::drawText (const String &text, int x, int y, int width, int height)

  draw text on current graphics context, using the current pen colour.
- void uwe::setColour (Colour newColour)

  set pen colour
- void uwe::setFontSize (float size)

  set the size of the font used to display text
- void uwe::fillAll (Colour backgroundColour)

  fill the window with a given colour
- void uwe::fillAll (uint8 red, uint8 green, uint8 blue)

  fille the window with a given colour
- void uwe::setColour (uint8 red, uint8 green, uint8 blue)

  set colour of pen
- int uwe::getWindowWidth ()

  calculate the width of the graphics window
- int uwe::getWindowHeight ()

  calculate the height of the graphics window
- Image uwe::loadImage (const char *imageFile)

  load and image into an image object from a file
- void uwe::drawImageAt (const Image &image, int x, int y)

  draw a previously loaded image onto current graphics context

5.1.1 Detailed Description

An easy-to-use library for drawing geometric shapes, such as points, lines, and circles. Support for displaying text and images, is also provided.

The graphics library is built using the excellent JUCE framework (https://juce.com/). JUCE is large and powerful framework, which relies heavely on advanced features of C++. This libary provides a simpler interface to some of JUCE's basic features, for graphics, but exposes some aspects when it makes sense.

5.1.2 Function Documentation

5.1.2.1 drawCircle()

```
void uwe::drawCircle (
            float x,
            float y,
            float radius,
            float lineThickness = 1 )  [inline]
```

draws a circle on the current graphics context.

Parameters

x	centre x position of circle
y0	centre y position of circle
radius	circle's radius

5.1.2.2 drawFilledCircle()

```
void uwe::drawFilledCircle (
            float x,
            float y,
            int radius )  [inline]
```

draws a filled circle on the current graphics context, using the current pen colour.

Parameters

x	centre x position of circle
y0	centre y position of circle
radius	circle's radius

5.1.2.3 drawFilledRect()

```
void uwe::drawFilledRect (
            float x,
            float y,
            float width,
            float height )  [inline]
```

draws a rectangle on the current graphics context, which is filled with the current pen colour.

Parameters

x	[description]
y	[description]
width	[description]
height	[description]

5.1.2.4 drawImageAt()

```
void uwe::drawImageAt (
            const Image & image,
```

```
            int x,
            int y )  [inline]
```

draw a previously loaded image onto current graphics context

Parameters

image	image to be displayed
x	top left x position to draw image
y	top left y position to draw image

5.1.2.5 drawLine()

```
void uwe::drawLine (
            float start_x,
            float start_y,
            float end_x,
            float end_y )  [inline]
```

draws a line on the current graphics context

Parameters

start_x	x start position of line
start_y	y start position of line
end_x	x end position of line
end_y	y end position of line

5.1.2.6 drawRect()

```
void uwe::drawRect (
            float x,
            float y,
            float width,
            float height,
            float lineThickness = 1 )  [inline]
```

draws a rectangle on the current graphics context.

Parameters

x	x position of top left hand corner
y	y position of top left hand corner

Parameters

width	width of rectangle
height	height of rectangle
lineThickness	optional line thickness of rectangle

5.1.2.7 drawText()

```
void uwe::drawText (
            const String & text,
            int x,
            int y,
            int width,
            int height )  [inline]
```

draw text on current graphics context, using the current pen colour.

Parameters

text	text to be displayed
x	top left x position to draw text
y	top right y position to draw text
width	max width of text
height	max height of text

5.1.2.8 fillAll() [1/2]

```
void uwe::fillAll (
            Colour backgroundColour )  [inline]
```

fill the window with a given colour

Parameters

backgroundColour	colour to fill screen

5.1.2.9 fillAll() [2/2]

```
void uwe::fillAll (
            uint8 red,
```

```
            uint8 green,
            uint8 blue )  [inline]
```

fille the window with a given colour

Parameters

red	valye of red channel
green	value of green channel
blue	valye of blue channel

5.1.2.10 getKeyCode()

```
int uwe::getKeyCode (
            KeyPress keyPress )  [inline]
```

convert keyPress to key scan code

Parameters

keyPress	representation of key pressed on keyboard

Returns

scan code of pressed key

5.1.2.11 getTextCharacter()

```
char uwe::getTextCharacter (
            KeyPress keyPress )  [inline]
```

convert keyPress to a character

Parameters

keyPress	representation of key pressed on keyboard

Returns

character pressed on keyboard

5.1.2.12 getWindowHeight()

```
int uwe::getWindowHeight ( )  [inline]
```

calculate the height of the graphics window

Returns

height

5.1.2.13 getWindowWidth()

```
int uwe::getWindowWidth ( )  [inline]
```

calculate the width of the graphics window

Returns

width

5.1.2.14 initialiseGraphics()

```
void uwe::initialiseGraphics (
            int windowWidth = internal::WINDOW_WIDTH_DEFAULT,
            int windowHeight = internal::WINDOW_HEIGHT_DEFAULT )  [inline]
```

setup graphics context.

no graphics should be preformed before calling this function.

Parameters

graphics_func	callback function, which is called each frame to paint onto the graphics context.

5.1.2.15 loadImage()

```
Image uwe::loadImage (
            const char * imageFile )  [inline]
```

load and image into an image object from a file

Parameters

imageFile	path and name of image file

Returns

loaded image

5.1.2.16 loop()

```
template<class F >
void uwe::loop (
            F paint_callback,
            internal::keypress_func keypress_callback = internal::keypress_func(internal:↩
:default_keypress_callback) )
```

loops while application window is active, calling paint callback to draw to the component's window.

Parameters

paint_callback	called when the screen is to be painted
keypress_callback	called when a key is pressed, optional parameter. if keypress_callback returns true, then the graphics context will close and return control to the application.

5.1.2.17 setColour() [1/2]

```
void uwe::setColour (
            Colour newColour )  [inline]
```

set pen colour

Parameters

newColour	colour to set pen

5.1.2.18 setColour() [2/2]

```
void uwe::setColour (
            uint8 red,
            uint8 green,
            uint8 blue )  [inline]
```

set colour of pen

Parameters

red	value of red channel
green	value of green channel
blue	value of blue channel

5.1.2.19 setFontSize()

```
void uwe::setFontSize (
            float size )  [inline]
```

set the size of the font used to display text

Parameters

size	font size

5.2 include/ufcfgl-30-1.h File Reference

Support library for ufcfgk-30-1.

```
#include <string>
#include <fstream>
#include <streambuf>
#include <memory>
#include <unistd.h>
```

Functions

- int uwe::getInt (void)
  read an integer from the command line
- int uwe::getInt (char *str)
  read an integer from a string
- char uwe::getch (void)
  read 1 character without echo
- char uwe::getche (void)
  read 1 character with echo
- int uwe::pause (useconds_t usec)
  [pause description]
- char * uwe::getString ()
  reads a string from stdin
- std::string uwe::readFile (std::string filename)
  read complete contents of file into a string
- constexpr unsigned int uwe::str2int (const char *str, int h=0)
  convert strings to a unique int, at compile time if possible

5.2.1 Detailed Description

Support library for ufcfgk-30-1.

An easy-to-use library for drawing geometric shapes, such as points, lines, and circles. Support for displaying text and images, is also provided.

5.2.2 Function Documentation

5.2.2.1 getch()

```
char uwe::getch (
            void  )
```

read 1 character without echo

Returns

[description]

5.2.2.2 getche()

```
char uwe::getche (
            void  )
```

read 1 character with echo

Returns

[description]

5.2.2.3 getInt() [1/2]

```
int uwe::getInt (
            void  )
```

read an integer from the command line

Returns

read integer

5.2.2.4 getInt() [2/2]

```
int uwe::getInt (
            char * str )
```

read an integer from a string

Parameters

str	to read integer from

Returns

read integer

5.2.2.5 getString()

```
char* uwe::getString ( )
```

reads a string from stdin

Returns

character array (note will leak if not deleted!)

5.2.2.6 pause()

```
int uwe::pause (
            useconds_t usec )
```

[pause description]

Parameters

usec	[description]

Returns

[description]

5.2.2.7 readFile()

```
std::string uwe::readFile (
            std::string filename )
```

read complete contents of file into a string

Parameters

filename	path/name of file to read

Returns

string containing file

5.2.2.8 str2int()

```
constexpr unsigned int uwe::str2int (
            const char * str,
            int h = 0 )
```

convert strings to a unique int, at compile time if possible

Parameters

str	string to be converted
h	height (used in create unique ID)

Returns

ID for string

Index

begin
    uwe::ZombieBot, 15

currentScore
    uwe::ZombieBot, 15

disableTimer
    uwe::ZombieBot, 15
drawCircle
    graphics.h, 20
drawFilledCircle
    graphics.h, 21
drawFilledRect
    graphics.h, 21
drawImageAt
    graphics.h, 21
drawLine
    graphics.h, 22
drawRect
    graphics.h, 22
drawText
    graphics.h, 23

enableTimer
    uwe::ZombieBot, 15
EntranceInfo
    uwe::EntranceInfo, 7

fillAll
    graphics.h, 23

getDescription
    uwe::RoomInfo, 11
getDirection
    uwe::EntranceInfo, 8
getEnd
    uwe::WorldLoader, 12
getEntrances
    uwe::RoomInfo, 11
getHtml
    uwe::ItemInfo, 9
getInfo
    uwe::WorldLoader, 13
getInt
    ufcfgl-30-1.h, 29
getInventoryHtml
    uwe::WorldLoader, 13
getItems
    uwe::RoomInfo, 11
    uwe::WorldLoader, 13
getKeyCode
    graphics.h, 25
getName
    uwe::ItemInfo, 9
    uwe::RoomInfo, 11
getRooms
    uwe::WorldLoader, 13
getStart
    uwe::WorldLoader, 14
getStartHtml
    uwe::WorldLoader, 14
getString
    ufcfgl-30-1.h, 30
getTextCharacter
    graphics.h, 25
getTo
    uwe::EntranceInfo, 8
getWindowHeight
    graphics.h, 25
getWindowWidth
    graphics.h, 26
getZombieCount
    uwe::RoomInfo, 12
getch
    ufcfgl-30-1.h, 29
getche
    ufcfgl-30-1.h, 29
graphics.h
    drawCircle, 20
    drawFilledCircle, 21
    drawFilledRect, 21
    drawImageAt, 21
    drawLine, 22
    drawRect, 22
    drawText, 23
    fillAll, 23
    getKeyCode, 25
    getTextCharacter, 25
    getWindowHeight, 25
    getWindowWidth, 26
    initialiseGraphics, 26
    loadImage, 26
    loop, 27
    setColour, 27
    setFontSize, 28

include/graphics.h, 19
include/ufcfgl-30-1.h, 28
initialiseGraphics
    graphics.h, 26
isLocked

uwe::EntranceInfo, 8
ItemInfo
uwe::ItemInfo, 9

loadImage
graphics.h, 26
loop
graphics.h, 27

pause
ufcfgl-30-1.h, 30
processCmd
uwe::ZombieBot, 15

readFile
ufcfgl-30-1.h, 30
RoomInfo
uwe::RoomInfo, 10

setColour
graphics.h, 27
setFontSize
graphics.h, 28
shouldQuit
uwe::ZombieBot, 17
str2int
ufcfgl-30-1.h, 31

ufcfgl-30-1.h
getInt, 29
getString, 30
getch, 29
getche, 29
pause, 30
readFile, 30
str2int, 31
uwe::EntranceInfo, 7
EntranceInfo, 7
getDirection, 8
getTo, 8
isLocked, 8
uwe::ItemInfo, 9
getHtml, 9
getName, 9
ItemInfo, 9
uwe::RoomInfo, 10
getDescription, 11
getEntrances, 11
getItems, 11
getName, 11
getZombieCount, 12
RoomInfo, 10
uwe::WorldLoader, 12
getEnd, 12
getInfo, 13
getInventoryHtml, 13
getItems, 13
getRooms, 13
getStart, 14
getStartHtml, 14
uwe::ZombieBot, 14
begin, 15
currentScore, 15
disableTimer, 15
enableTimer, 15
processCmd, 15
shouldQuit, 17
uwe::ZombieServer, 17
ZombieServer, 17

ZombieServer
uwe::ZombieServer, 17