Drako Engine

Release 0.1.0

Drew "Drako0812" Wibbenmeyer

Jun 04, 2020

CONTENTS

1 Documentation **3**

2 License **5**

2.1 Drako Engine API 5

Index **15**

A modern C++ game engine using a high-performance Entity-Component-System architecture. Also, *Drako Engine* should be considered a programmer-first engine, not to say that designer and artist features won't be added, they are just a secondary concern. This is mostly because the initial target audience for *Drako Engine* is for lone developers like myself.

CHAPTER

ONE

DOCUMENTATION

The documentation is hosted on Read the Docs

CHAPTER

TWO

LICENSE

Drako Engine is licensed under the MIT license found on GitHub

2.1 Drako Engine API

2.1.1 Class Hierarchy

2.1.2 File Hierarchy

2.1.3 Full API

Namespaces

Namespace DrakoEngine

Contents

- *Classes*
- *Typedefs*

Classes

- *Class App*

Typedefs

- *Typedef DrakoEngine::byte*
- *Typedef DrakoEngine::f32*
- *Typedef DrakoEngine::f64*
- *Typedef DrakoEngine::Float*
- *Typedef DrakoEngine::i16*
- *Typedef DrakoEngine::i32*

- *Typedef DrakoEngine::i64*
- *Typedef DrakoEngine::i8*
- *Typedef DrakoEngine::Int*
- *Typedef DrakoEngine::pstr*
- *Typedef DrakoEngine::pvec*
- *Typedef DrakoEngine::str*
- *Typedef DrakoEngine::u16*
- *Typedef DrakoEngine::u32*
- *Typedef DrakoEngine::u64*
- *Typedef DrakoEngine::u8*
- *Typedef DrakoEngine::UInt*
- *Typedef DrakoEngine::vec*

Namespace sf

Classes and Structs

Class App

- Defined in file_DrakoEngineLib_app.hpp

Class Documentation

class `DrakoEngine::`**App**

Entry point class of a Game/Application.

Public Types

enum Result

Signals success or failure of some functions.

Values:

enumerator Failure = 0

Indicates some kind of failure.

enumerator Success

Indicates successful execution.

Public Functions

App()

~App()

Result **Startup**()
Runs before entering the game loop.

Result **Shutdown**()
Runs after exiting the game loop.

void **HandleEvent**(sf::Event **const** *&event*)
Handles individual events.

void **Update**()
Updates game or application state once per game loop.

void **Render**()
Renders graphics to the screen.

Result **Run**(int *argc*, char **argv*[])
Starts the game loop.

Starts the game loop using the following execution order:

i. *Startup()*
ii. Game Loop:
   A. *Update()*
      I. HandleEvent(event)
   B. *Render()*
iii. *Shutdown()*

Public Members

vec<*str*> **Arguments**
Holds the command-line arguments.

Protected Functions

Result **ClientRun**()
Resets the state of the game/application without destroying the window or unloading major engine components.

Resets the game similarly to closing the game and restarting it manually. Execute client Run code such as handling command-line arguments.

Protected Attributes

sf::RenderWindow **Window**

bool **ShouldClose**

bool **DoSoftReset**

bool **DoHardReset**

Defines

Define DRAKOENGINE_COMPONENT_MEMORY_BYTES

- Defined in file_DrakoEngineLib_config.hpp

Define Documentation

DRAKOENGINE_COMPONENT_MEMORY_BYTES

Define DRAKOENGINE_DEFAULT_FLOAT_TYPE

- Defined in file_DrakoEngineLib_config.hpp

Define Documentation

DRAKOENGINE_DEFAULT_FLOAT_TYPE

Define DRAKOENGINE_DEFAULT_SIGNED_INTEGER_TYPE

- Defined in file_DrakoEngineLib_config.hpp

Define Documentation

DRAKOENGINE_DEFAULT_SIGNED_INTEGER_TYPE

Define DRAKOENGINE_DEFAULT_UNSIGNED_INTEGER_TYPE

- Defined in file_DrakoEngineLib_config.hpp

Define Documentation

DRAKOENGINE_DEFAULT_UNSIGNED_INTEGER_TYPE

Define DRAKOENGINE_STRING_MEMORY_BYTES

- Defined in file_DrakoEngineLib_config.hpp

Define Documentation

DRAKOENGINE_STRING_MEMORY_BYTES

Define DRAKOENGINELIB_EXPORTED

- Defined in file_DrakoEngineLib_exported.hpp

Define Documentation

DRAKOENGINELIB_EXPORTED

Define DRAKOENGINELIB_NOT_EXPORTED

- Defined in file_DrakoEngineLib_exported.hpp

Define Documentation

DRAKOENGINELIB_NOT_EXPORTED

Define NOMINMAX

- Defined in file_DrakoEngineLib_del_pch.hpp

Define Documentation

NOMINMAX

Typedefs

Typedef DrakoEngine::byte

- Defined in file_DrakoEngineLib_DrakoEngine.hpp

Typedef Documentation

using `DrakoEngine::`**`byte`** = std::byte

Typedef DrakoEngine::f32

- Defined in file_DrakoEngineLib_DrakoEngine.hpp

Typedef Documentation

using `DrakoEngine::`**`f32`** = float

Typedef DrakoEngine::f64

- Defined in file_DrakoEngineLib_DrakoEngine.hpp

Typedef Documentation

using `DrakoEngine::`**`f64`** = double

Typedef DrakoEngine::Float

- Defined in file_DrakoEngineLib_config.hpp

Typedef Documentation

using `DrakoEngine::`**`Float`** = *f64*

Typedef DrakoEngine::i16

- Defined in file_DrakoEngineLib_DrakoEngine.hpp

Typedef Documentation

using `DrakoEngine::i16` = std::int16_t

Typedef DrakoEngine::i32

- Defined in file_DrakoEngineLib_DrakoEngine.hpp

Typedef Documentation

using `DrakoEngine::i32` = std::int32_t

Typedef DrakoEngine::i64

- Defined in file_DrakoEngineLib_DrakoEngine.hpp

Typedef Documentation

using `DrakoEngine::i64` = std::int64_t

Typedef DrakoEngine::i8

- Defined in file_DrakoEngineLib_DrakoEngine.hpp

Typedef Documentation

using `DrakoEngine::i8` = std::int8_t

Typedef DrakoEngine::Int

- Defined in file_DrakoEngineLib_config.hpp

Typedef Documentation

using `DrakoEngine::Int` = *i64*

Typedef DrakoEngine::pstr

- Defined in file_DrakoEngineLib_DrakoEngine.hpp

Typedef Documentation

using `DrakoEngine::pstr` = std::pmr::string

Typedef DrakoEngine::pvec

- Defined in file_DrakoEngineLib_DrakoEngine.hpp

Typedef Documentation

using `DrakoEngine::pvec` = std::pmr::vector<Type>

Typedef DrakoEngine::str

- Defined in file_DrakoEngineLib_DrakoEngine.hpp

Typedef Documentation

using `DrakoEngine::str` = std::string

Typedef DrakoEngine::u16

- Defined in file_DrakoEngineLib_DrakoEngine.hpp

Typedef Documentation

using `DrakoEngine::u16` = std::uint16_t

Typedef DrakoEngine::u32

- Defined in file_DrakoEngineLib_DrakoEngine.hpp

Typedef Documentation

using `DrakoEngine::u32` = std::uint32_t

Typedef DrakoEngine::u64

- Defined in file_DrakoEngineLib_DrakoEngine.hpp

Typedef Documentation

using `DrakoEngine::u64` = std::uint64_t

Typedef DrakoEngine::u8

- Defined in file_DrakoEngineLib_DrakoEngine.hpp

Typedef Documentation

using `DrakoEngine::u8` = std::uint8_t

Typedef DrakoEngine::UInt

- Defined in file_DrakoEngineLib_config.hpp

Typedef Documentation

using `DrakoEngine::UInt` = *u64*

Typedef DrakoEngine::vec

- Defined in file_DrakoEngineLib_DrakoEngine.hpp

Typedef Documentation

using `DrakoEngine::vec` = std::vector<Type>

INDEX

D

DrakoEngine::App (*C++ class*), 6
DrakoEngine::App::~App (*C++ function*), 7
DrakoEngine::App::App (*C++ function*), 7
DrakoEngine::App::Arguments (*C++ member*), 7
DrakoEngine::App::ClientRun (*C++ function*), 7
DrakoEngine::App::DoHardReset (*C++ member*), 8
DrakoEngine::App::DoSoftReset (*C++ member*), 8
DrakoEngine::App::HandleEvent (*C++ function*), 7
DrakoEngine::App::Render (*C++ function*), 7
DrakoEngine::App::Result (*C++ enum*), 6
DrakoEngine::App::Result::Failure (*C++ enumerator*), 6
DrakoEngine::App::Result::Success (*C++ enumerator*), 6
DrakoEngine::App::Run (*C++ function*), 7
DrakoEngine::App::ShouldClose (*C++ member*), 8
DrakoEngine::App::Shutdown (*C++ function*), 7
DrakoEngine::App::Startup (*C++ function*), 7
DrakoEngine::App::Update (*C++ function*), 7
DrakoEngine::App::Window (*C++ member*), 8
DrakoEngine::byte (*C++ type*), 10
DrakoEngine::f32 (*C++ type*), 10
DrakoEngine::f64 (*C++ type*), 10
DrakoEngine::Float (*C++ type*), 10
DrakoEngine::i16 (*C++ type*), 11
DrakoEngine::i32 (*C++ type*), 11
DrakoEngine::i64 (*C++ type*), 11
DrakoEngine::i8 (*C++ type*), 11
DrakoEngine::Int (*C++ type*), 11
DrakoEngine::pstr (*C++ type*), 12
DrakoEngine::pvec (*C++ type*), 12
DrakoEngine::str (*C++ type*), 12
DrakoEngine::u16 (*C++ type*), 12
DrakoEngine::u32 (*C++ type*), 13
DrakoEngine::u64 (*C++ type*), 13
DrakoEngine::u8 (*C++ type*), 13
DrakoEngine::UInt (*C++ type*), 13
DrakoEngine::vec (*C++ type*), 13
DRAKOENGINE_COMPONENT_MEMORY_BYTES (*C macro*), 8
DRAKOENGINE_DEFAULT_FLOAT_TYPE (*C macro*), 8
DRAKOENGINE_DEFAULT_SIGNED_INTEGER_TYPE (*C macro*), 8
DRAKOENGINE_DEFAULT_UNSIGNED_INTEGER_TYPE (*C macro*), 9
DRAKOENGINE_STRING_MEMORY_BYTES (*C macro*), 9
DRAKOENGINELIB_EXPORTED (*C macro*), 9
DRAKOENGINELIB_NOT_EXPORTED (*C macro*), 9

N

NOMINMAX (*C macro*), 9