SFMLResource

Jack Featherstone

Version 1.0

06/29/2019

Class Index

Class List

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

Class Documentation

ResourceManager Class Reference

Static Public Member Functions

static sf::Texture * **getTexture** (const std::string filePath)

Get the Texture that is at the given file path. If this texture has been previously loaded, a pointer to the previous object will be returned. Otherwise, a new pointer entry will be created in the texture map and returned.

static int **getNumberOfTextures** ()

Returns the size of the m_textureMap object.

static void **preLoadTextures** (const std::string folderPath, bool recurse=true)

Load all of the files in a given folder (whose file extensions appear in TEXTURE_EXTENSIONS) into the texture map.

static void **setInvalidTexturePath** (const std::string filePath)

Set the Invalid Texture Path to a new value. Can be relative to the project folder or an absolute path.

static std::string **getInvalidTexturePath** ()

Get the Invalid Texture Path.

static void **clearTextures** ()

Delete all of the pointers held in m_textureMap and clear the respective entries.

static sf::SoundBuffer * **getSoundBuffer** (const std::string filePath)

Get the SoundBuffer that is at the given file path. If this sound has been previously loaded, a pointer to the previous object will be returned. Otherwise, a new pointer entry will be created in the sound map and returned.

static int **getNumberOfSoundBuffers** ()

Returns the size of the m_soundMap object.

static void **preLoadSoundBuffers** (const std::string folderPath, bool recurse=true)

Load all of the files in a given folder (whose file extensions appear in SOUND_EXTENSIONS) into the sound map.

static void **setInvalidSoundPath** (const std::string filePath)

Set the Invalid Sound Path to a new value. Can be relative to the project folder or an absolute path.

static std::string **getInvalidSoundPath** ()

Get the Invalid Sound Path.

static void **clearSoundBuffers** ()

Delete all of the pointers held in m_soundMap and clear the respective entries.

static sf::Font * **getFont** (const std::string filePath)

Get the Font that is at the given file path. If this sound has been previously loaded, a pointer to the previous object will be returned. Otherwise, a new pointer entry will be created in the font map and returned.

static int **getNumberOfFonts** ()

Returns the size of the m_fontMap object.

static void **preLoadFonts** (const std::string folderPath, bool recurse=true)

Load all of the files in a given folder (whose file extensions appear in FONT_EXTENSIONS) into the sound map.

static void **setInvalidFontPath** (const std::string filePath)

Set the Invalid Font Path to a new value. Can be relative to the project folder or an absolute path.

static std::string **getInvalidFontPath** ()

Get the Invalid Font Path.

static void **clearFonts** ()

Delete all of the pointers held in m_fontMap and clear the respective entries.

static bool **contains** (std::vector< std::string > vec, std::string str)

A simple method to find whether a string is contained in a vector.

Member Function Documentation

bool ResourceManager::contains (std::vector< std::string > *vec*, std::string *str*)[static]

A simple method to find whether a string is contained in a vector.

Parameters:

vec	The vector of reference strings
str	The string that may or may not be present in the vector

Returns:

true The string is found in the vector
false The string is not found in the vector

sf::Font * ResourceManager::getFont (const std::string *filePath*)[static]

Get the Font that is at the given file path. If this sound has been previously loaded, a pointer to the previous object will be returned. Otherwise, a new pointer entry will be created in the font map and returned.

Parameters:

filePath	The (relative to project folder or absolute) location of the font file

Returns:

sf::Font* A pointer to the font at the given file path

std::string ResourceManager::getInvalidFontPath ()[static]

Get the Invalid Font Path.

Returns:

std::string The file that will be referenced when a font is not found.

std::string ResourceManager::getInvalidSoundPath ()[static]

Get the Invalid Sound Path.

Returns:

std::string The file that will be referenced when a sound is not found.

std::string ResourceManager::getInvalidTexturePath ()[static]

Get the Invalid Texture Path.

Returns:

std::string The file that will be referenced when a texture is not found.

int ResourceManager::getNumberOfFonts ()[static]

Returns the size of the m_fontMap object.

Returns:

int The number of font entries loaded in the resource manager

int ResourceManager::getNumberOfSoundBuffers ()[static]

Returns the size of the m_soundMap object.

Returns:

int The number of sound entries loaded in the resource manager

int ResourceManager::getNumberOfTextures ()[static]

Returns the size of the m_textureMap object.

Returns:

int The number of texture entries loaded in the resource manager

sf::SoundBuffer * ResourceManager::getSoundBuffer (const std::string *filePath*)[static]

Get the SoundBuffer that is at the given file path. If this sound has been previously loaded, a pointer to the previous object will be returned. Otherwise, a new pointer entry will be created in the sound map and returned.

Parameters:

filePath	The (relative to project folder or absolute) location of the sound file

Returns:

sf::SoundBuffer* A pointer to the sound at the given file path

sf::Texture * ResourceManager::getTexture (const std::string *filePath*)[static]

Get the Texture that is at the given file path. If this texture has been previously loaded, a pointer to the previous object will be returned. Otherwise, a new pointer entry will be created in the texture map and returned.

Parameters:

filePath	The (relative to project folder or absolute) location of the texture file

Returns:

sf::Texture* A pointer to the texture at the given file path

void ResourceManager::preLoadFonts (const std::string *folderPath*, bool *recurse* = true) [static]

Load all of the files in a given folder (whose file extensions appear in FONT_EXTENSIONS) into the sound map.

Parameters:

folderPath	The The (relative to project folder or absolute) location of the folder where fonts are to be loaded from
recurse	Whether or not the manager should search for files recursively i.e. below the given folder. Default is true

void ResourceManager::preLoadSoundBuffers (const std::string *folderPath*, bool *recurse* = true)[static]

Load all of the files in a given folder (whose file extensions appear in SOUND_EXTENSIONS) into the sound map.

Parameters:

folderPath	The The (relative to project folder or absolute) location of the folder where sounds are to be loaded from
recurse	Whether or not the manager should search for files recursively i.e. below the given folder. Default is true

void ResourceManager::preLoadTextures (const std::string *folderPath*, bool *recurse* = true) [static]

Load all of the files in a given folder (whose file extensions appear in TEXTURE_EXTENSIONS) into the texture map.

Parameters:

folderPath	The The (relative to project folder or absolute) location of the folder where textures are to be loaded from
recurse	Whether or not the manager should search for files recursively i.e. below the given folder. Default is true

static void ResourceManager::setInvalidFontPath (const std::string *filePath*)[static]

Set the Invalid Font Path to a new value. Can be relative to the project folder or an absolute path.

Parameters:

filePath	The new file that will be referenced when a font is not found.

static void ResourceManager::setInvalidSoundPath (const std::string *filePath*)[static]

Set the Invalid Sound Path to a new value. Can be relative to the project folder or an absolute path.

Parameters:

filePath	The new file that will be referenced when a sound is not found.

static void ResourceManager::setInvalidTexturePath (const std::string *filePath*)[static]

Set the Invalid Texture Path to a new value. Can be relative to the project folder or an absolute path.

Parameters:

filePath	The new file that will be referenced when a texture is not found.

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

src/ResourceManager.hpp
src/ResourceManager.cpp