

```
nSocketEvent
et, nbBytes: unsigned int): void
cket *,
signed int): void
et *): void
ce>>
ic
&,delta:uint64 t,
nt32 t): bool
,strn:std::string const &,
nt32 t=12): bool
&, onLoop:bool=false): bool
&,onLoop:bool=true): bool
&)
&, volume:float=100)
t &
bool
                                ContentManager
      -mTextures: std::shared ptr<IResourceManager<sf::Texture>>
      -mSprites: std::shared ptr<IResourceManager<SpriteModel>>
     -mFonts: std::shared ptr<IResourceManager<sf::Font>>
     -mMusics: std::shared_ptr<IResourceManager<std::string>>
      -mSounds: std::shared_ptr<IResourceManager<sf::sSound>>
     +loadTextures()
     +loadSprites()
     +loadFonts()
     +loadMusics()
     +loadSounds()
     +getTextures(): std::shared_ptr<IResourceManager<sf::Texture>> const &
     +getSprites(): std::shared_ptr<IResourceManager<SpriteModel>> &
     +getFonts(): std::shared_ptr<IResourceManager<sf::Font>> const &
     +getMusics(): std::shared_ptr<IResourceManager<std::string>> const &
      +getSounds(): std::shared_ptr<IResourceManager<sf::sSound>> const &
                                        <<interface template>>
                                        IRessourceManager
                  +loadResources(key:std::string const &,path:std::string const &)
                  +unloadResources()
                  +getResource(key:std::string const &): T
                                                                                        MusicManager
                                                                    -mListResources: std::map
'extureManager
                                                                   +loadResources(std::string const &, std::str
::map
                                                                   +unloadResources()
```

+getResource(key:std::string const &): std:

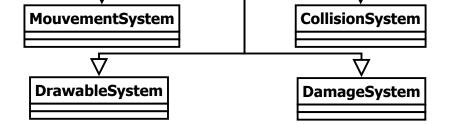
ace>>

string const &, std::string const &)

::string const &): sf::Texture &

ing const &)





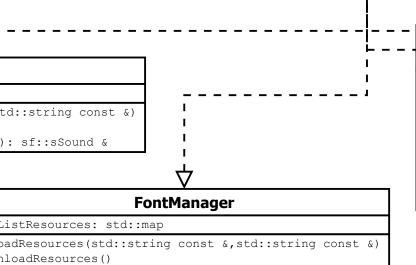
SoundManager

-mListResources: std::map

+loadResources(std::string const &,s

+unloadResources()

+getResource(key:std::string const &



etResource(key:std::string const &): sf::Font &

```
Sprite inager
```

-mListResources: std::map

+loadResources(key:std::string const &,path:std::str columns:uint32_t,lines:uint32_t,

so:SpriteOffset const &,loop:bool=fal

+unloadResources()

+getResource(key:std::string const &): SpriteModel &

. &)

ing const &,

ing const &,

.se)