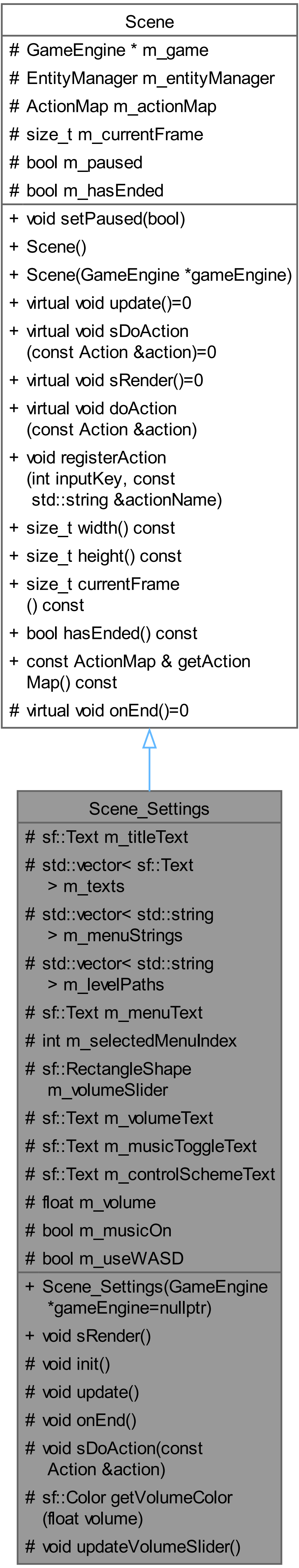
Scene
# GameEngine * m_game
# EntityManager m_entityManager
# ActionMap m_actionMap
# size_t m_currentFrame
# bool m_paused
# bool m_hasEnded
+ void setPaused(bool)
+ Scene()
+ Scene(GameEngine *gameEngine)
+ virtual void update()=0
+ virtual void sDoAction (const Action &action)=0
+ virtual void sRender()=0
+ virtual void doAction (const Action &action)
+ void registerAction (int inputKey, const std::string &actionName)
+ size_t width() const
+ size_t height() const
+ size_t currentFrame () const
+ bool hasEnded() const
+ const ActionMap & getActionMap() const
# virtual void onEnd()=0
Scene_Settings
# sf::Text m_titleText
# std::vector< sf::Text > m_texts
# std::vector< std::string > m_menuStrings
# std::vector< std::string > m_levelPaths
# sf::Text m_menuText
# int m_selectedMenuIndex
# sf::RectangleShape m_volumeSlider
# sf::Text m_volumeText
# sf::Text m_musicToggleText
# sf::Text m_controlSchemeText
# float m_volume
# bool m_musicOn
# bool m_useWASD
+ Scene_Settings(GameEngine *gameEngine=nullptr)
+ void sRender()
# void init()
# void update()
# void onEnd()
# void sDoAction(const Action &action)
# sf::Color getVolumeColor (float volume)
# void updateVolumeSlider()