```
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 & getAction
  Map() const
# virtual void onEnd()=0
          Scene Settings
  # sf::Text m titleText
  # std::vector< sf::Text
     > m texts
  # std::vector< std::string
     > m_menuStrings
  # 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()
```