```
Scene
# GameEngine * m_game
# EntityManager m_entityManager
# ActionMap m actionMap
# size_t m_currentFrame
# bool m_paused
+ 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 currentFrame
  () const
+ const ActionMap & getAction
  Map() const
# virtual void onEnd()=0
            Scene_Play
  # sf::Text m_scoreText
  # size_t m_score
  # sf::View m_view
  # size t m currentBackground
  # size_t m_pastBackground
  # size_t m_currentFrame
  # size_t m_highestScore
  # int m_alpha
  # bool m drawTextures
  # bool m_drawCollision
  # bool m_move
  # bool m_transition
  # bool m_end
  # Vec2 m_gridSize
  # Vec2 m targetViewPosition
  # size t m_platformSpacing
  # std::shared_ptr< Entity
     > m_player
  + Scene_Play(GameEngine
     *gameEngine)
  + void init()
  + void update()
  + void onEnd()
  + void replay()
  + void sRemoveDeadPlatforms()
  + void sPlatformGeneration()
  + void sAnimation()
    void sMovement()
  + void sCollision()
  + void sRender()
  + void sDoAction(const
     Action &a)
  + void spawnPlayer()
  + void loadLevel()
```