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_Play # sf::Text m_gridText # std::string m levelPath # size_t m_coins # sf::Text m_coinsText # std::vector< sf::Text > m_infoVector # bool m_drawlnfo # bool m_upgrade # bool m_drawTextures # bool m_drawCollision # bool m_roadGrid # bool m_grassGrid # bool m_mouseItem # bool m_attack # bool m_nightFall # Vec2 m_attackPos # sf::RectangleShape m_lightningSquare # sf::RectangleShape m defenseSquare # sf::RectangleShape m_nightFilter # Vec2 m_gridSize # size t m currentFrame # size_t m_selectedItem # size_t m_lastFrameDefense Spawn # size_t m_opacity # size_t m_spawnRateFrame # std::vector< sf::Rectangle Shape > m_shopRectangles # std::vector< sf::Rectangle Shape > m_roadRectanglesGrid # std::vector< sf::Rectangle Shape > m_grassRectanglesGrid # std::map< size_t, bool > m_usedRectanglesIndex # std::shared_ptr< Entity > m player # sf::Clock m_clock # sf::Clock m_weatherClock # sf::Clock m_nightClock + Scene Play(GameEngine *gameEngine, const std ::string &levelPath) + void init(const std ::string &path) + void update() + void onEnd() + void sAnimation() + void sMovement() + void sEnemySpawner()

+ void sSpawnEnemy(size

_t line)

+ void sInfo()

> tower)

Vec2 &pos)

+ void sCollision()+ void sRender()+ void sShop()+ void sHealth()

+ void sDoAction(const Action &a)+ void sPlacement()+ void spawnPlayer()

+ void attack(std::shared
 _ptr< Entity > enemy,
 std::shared_ptr< Entity</pre>

+ void spawnSpikes(const std::string &type, const

+ void generateRoadRectangles()+ void generateGrassRectangles()

Scene_Menu

std::string m_title

sf::Text m_titleText

std::vector< sf::Text

std::vector< std::string

> m_menuStrings

std::vector< std::string

> m levelPaths

+ void sRender()

void update()

void onEnd()

void sDoAction(const

Action &action)

void init()

sf::Text m menuText

int m_selectedMenuIndex

*gameEngine=nullptr)

+ Scene_Menu(GameEngine

> m_texts

```
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()
```