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_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

Scene Menu

std::string m_title1

std::string m_title2

sf::Text m_titleText1

sf::Text m_titleText2

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

+ Scene_Menu(GameEngine

*gameEngine=nullptr)

> m texts

Scene Credits

std::string m_title

sf::Text m_titleText

std::vector< sf::Text

std::vector< std::string

> m_menuStrings

sf::Text m_menuText

void sDoAction(const

Action &action)

+ Scene Credits(GameEngine

*gameEngine=nullptr)

> m texts

+ void sRender()

void update()

void onEnd()

void init()

- # size_t m_spawnRateFrame
- # std::vector< sf::Rectangle Shape > m_shopRectangles
- # std::vector< sf::Rectangle
 Shape > m_roadRectanglesGrid
 # std::vector< sf::Rectangle</pre>
- Shape > m_grassRectanglesGrid
- # 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 sMovement()+ void sEnemySpawner()
- + void sSpawnEnemy(size _t line)
- + void sCollision()
- + void sRender()+ void sShop()
- + void sHealth()
- + void sDoAction(const Action &a)
- + void sPlacement()
- + void spawnPlayer()
- + void sUpgrade()
- + void spawnBarricade (const Vec2 &, size_t)+ void sInfo()
- + void attack
- + void attack(std::shared _ptr< Entity > a, std ::shared_ptr< Entity > b)
- + void generateRoadRectangles()
- + void generateGrassRectangles()
- + void spawnSpikes(const std::string &type, const Vec2 &pos)

- 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()