

Scene
<pre># GameEngine * m_game # EntityManager m_entityManager # ActionMap m_actionMap # size_t m_currentFrame # bool m_paused # bool m_hasEnded</pre>
<pre>+ 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</pre>



Scene_Play
<pre># sf::Text m_gridText # std::string m_levelPath # size_t m_coins # sf::Text m_coinsText # std::vector< sf::Text > m_infoVector # bool m_drawInfo # 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</pre>
<pre>+ 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 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(std::shared _ptr< Entity > a, std ::shared_ptr< Entity > b) + void generateRoadRectangles() + void generateGrassRectangles() + void spawnSpikes(const std::string &type, const Vec2 &pos)</pre>