GameEngine + GameEngine(const std ::string &path, const std::string &path2) + void changeScene(const std::string &, std::shared _ptr< Scene > scene, bool endCurrentScene) + void run() EntityManager + void quit() + Assets & getAssets() + EntityManager() + sf::RenderWindow & + void init() window() map< int, std::string > size_t bool + void update() + bool isRunning() + std::shared_ptr< Entity + size_t getCoins() > addEntity(const std + size_t getVolume() ::string &tag) + size_t getHighestScore() + EntityVec & getEntities() + bool getMusic() + EntityVec & getEntities (const std::string &tag) + bool getWASD() + void setCoins(size _t coins) + void setVolume(size _t volume) + void setHighestScore (size t highestScore) + void setMusic(bool music) + void setWASD(bool wasd) #m_game #m_entityManager #m_actionMap #m_currentFrame #m_paused Scene + void setPaused(bool) + Scene() + Scene(GameEngine *gameEngine) + virtual void update()=0 + virtual void sDoAction (const Action &action)=0 vector< sf::Text > Text vector< std::string > int + 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 #m_levelPaths #m menuText #m_texts #m_selectedMenuIndex #m_titleText #m_menuStrings Scene_Shop + Scene_Shop(GameEngine *gameEngine=nullptr) + void sRender() # void init() # void update() # void onEnd()

void sDoAction(const Action &action)