**Comprised of three parts:**

* **Rendering Engine**
* **Sprite Manager**
* **Animated Sprite**

**Rendering Engine:  
Creation of a Rendering Engine is REQUIRED before ANY sprites can be created:**

RenderEngine rendEng;

The rendering engine will automatically load config from the settings file, which will be automatically created when needed with default values

The rendering engine has several up front commands:

* render(); This command will render all created AnimatedSprites so far created
* addSprite(sf::Drawable\*); This will add any drawable thing to the spritelist, this includes non-animated sprites, shapes, texture, etc
* removeSprite(sf::Drawable\*); will attempt to remove this sprite from the rendering list
* flush(); Removes ALL sprites from the rendering list, effectively causing rendering to end until it's repopulated
* window; This is the SFML window that is used in the engine
* view; The SFML view used in the engine, by the window

**Sprite Manager**

**Loads and holds sprites**

SpriteManager spriteMan;

Multiple SpriteManagers can be created, they will share one list of sprites so don't worry about it

* loadFile(string); Loads an XML file (Look under assets for examples) and converts it to sprites
* getSprite(string); Returns an animated sprite of that name
* addSprite(string, AnimatedSprite); Adds an animated sprite, if you're adding one without loading it from a file (Why?!)

**Animated Sprite**

**A sprite with animations**

AnimatedSprite aniSprite(float speed, bool paused, bool looped);

**Definitions:**

* speed: Modifier which speeds up animations, a speed of 2 causes the animations to go twice as fast
* paused: If the animation is currently playing. When paused the first frame of the animation is rendered
* looped: If not looped the animation will stop after playing once

**Functions:**

* update(sf::Time deltaTime); Must be called once every frame with the difference in time between frames
* setAnimation(animation); if you have a custom animation, not loaded into the sprite, it loads it
* setAnimation(string); Loads an animation by that name loaded into the sprite
* addAnimation(animation); Adds an animation
* doesAnimationExist(string); checks if an animation by that name exists in the sprite
* render(bool); sets if the sprite is rendered or not (Using Rendering Engine "Add/Remove sprite")
* setSpeed(float); Sets the speed of the sprite (must be greater than 0)
* getSpeed(); Gets the current speed
* play(); Plays the currently loaded animation
* play(Animation); Plays a specific animation
* pause(); Pauses the animation on the current frame
* stop(); Pauses the animation on the FIRST frame
* setLooped(bool); Sets if the animation is looped
* setColor(Color); sets the color mask
* getAnimation(); Gets currently loaded animation
* getLocalBounds/ getGlobalBounds : See SFML sprite documentation
* isLooped(); returns true if loop is set to true
* isPaused(); returns true if the animation is paused
* setFrame(size\_t, bool);
* setRenderingEngine(RenderingEngine\*); Sets the current rendering engine, should be automatically set