Aura Game Engine

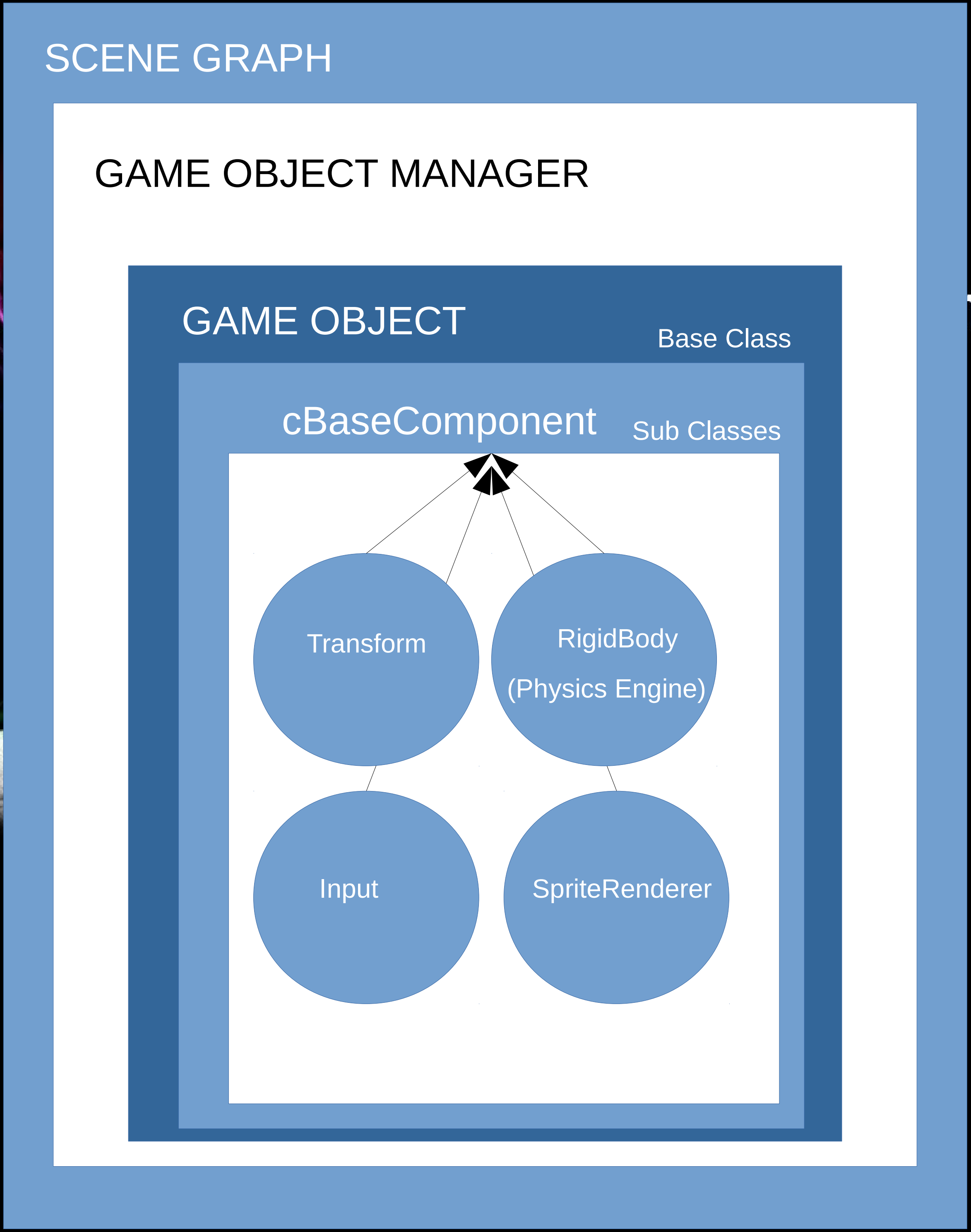
By Robert Esposito and Anton Zeitger

Architecture

Aura Game Engine

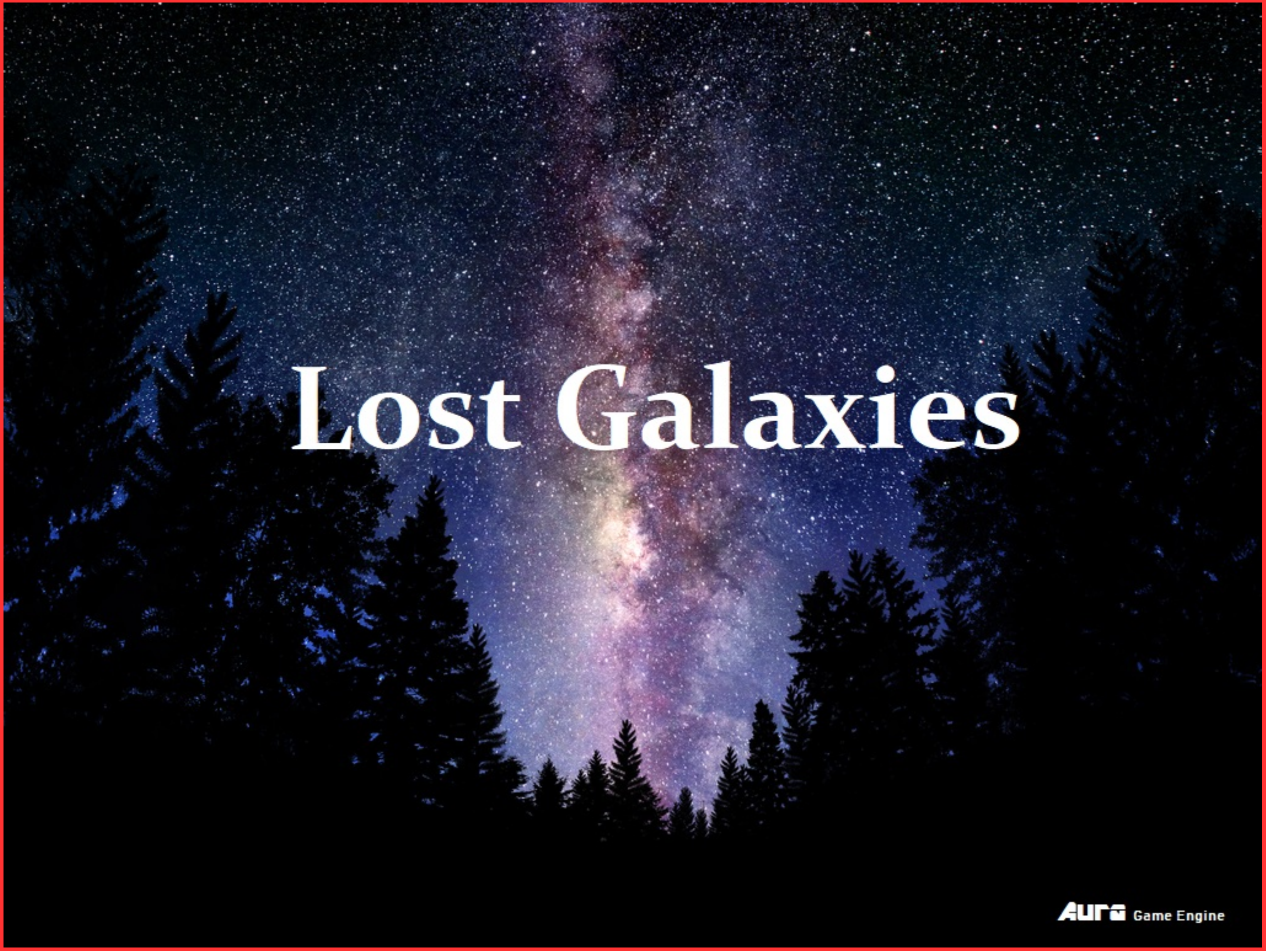
Discussion Points:

- 2D game engine, in C++ and using SFML Library
- Component Architecture implementation
- SceneGraph, GameObjManager, GameObj, cBaseComponent
- Aura:: Static Library implementation
- XML example


SCENE GRAPH
GAME OBJECT MANAGER
GAME OBJECT
Base Class
cBaseComponent
Sub Classes
Transform
RigidBody
(Physics Engine)
Input
SpriteRenderer

me Engine

Aura Game Engine



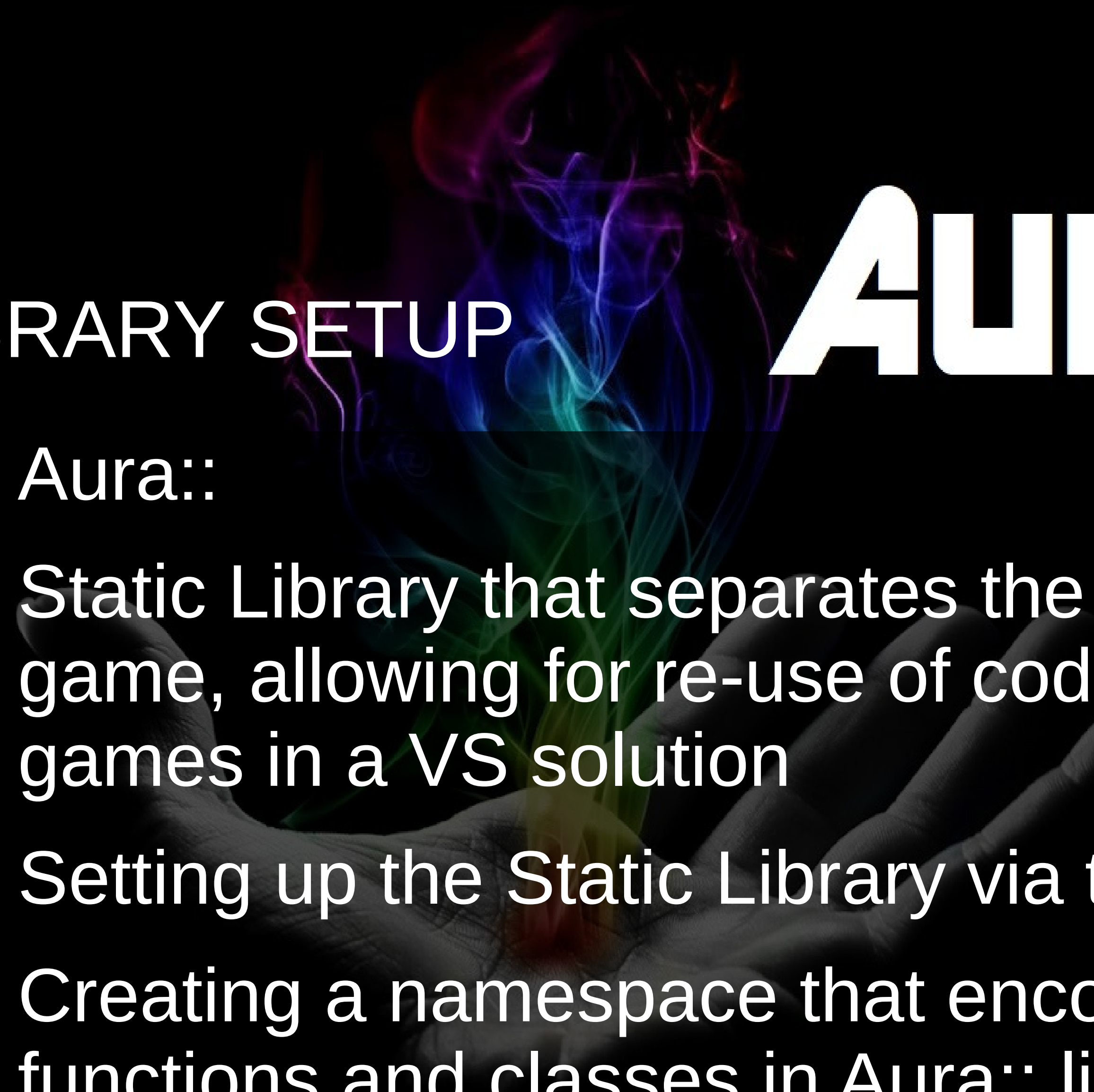
Our Game Project

Aura Game Engine

Lost Galaxies

* 2D Single player game
* Asteroid type game, avoid and shoot asteroids to win
* Mechanics: 2D physics (collisions, transform, input controls, sprite renderer)

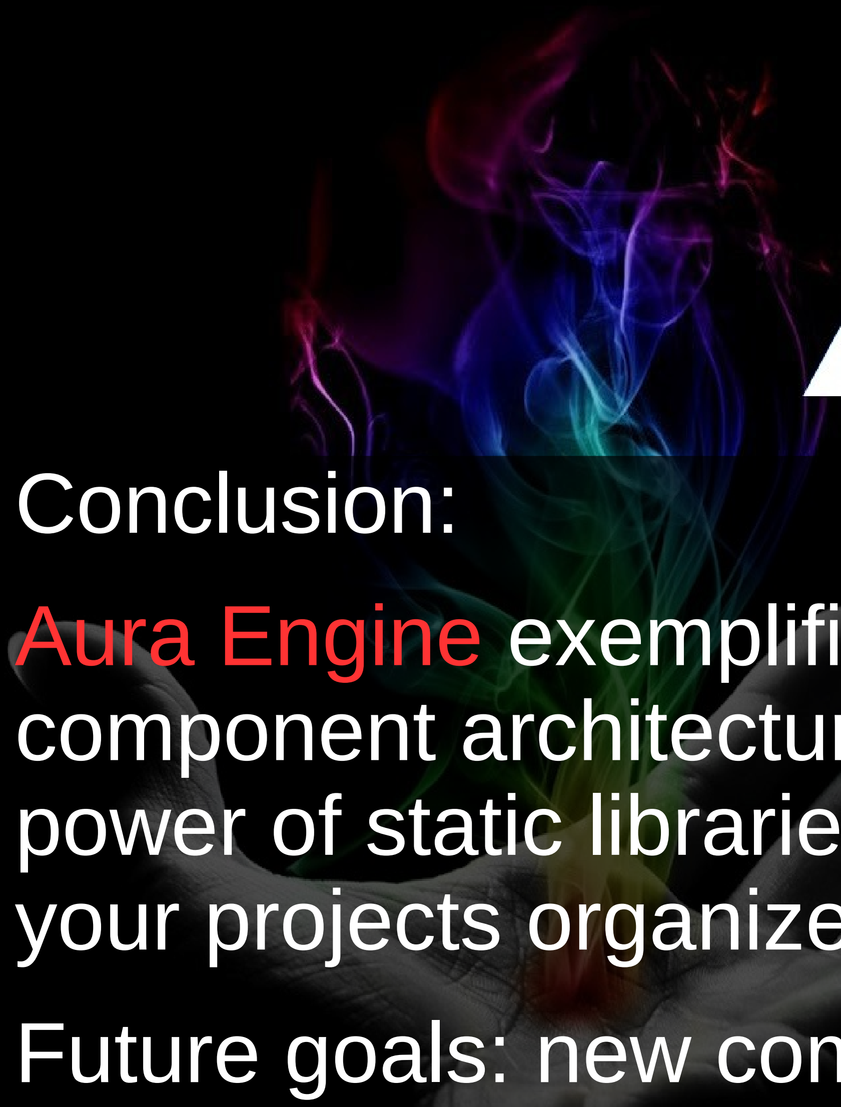
LIBRARY SETUP



- Aura::
- Static Library that separates the engine from the game, allowing for re-use of code and multiple games in a VS solution
- Setting up the Static Library via the project settings
- Creating a namespace that encompasses all functions and classes in Aura:: library
- Linker settings to connect projects to the library

XML/JSON

- Reading from an XML or JSON file (Scene Manager)



Aura Game Engine

Conclusion:

- Aura Engine exemplifies the usefulness of the component architecture design pattern and the power of static libraries to reuse code and make your projects organized and expandable.
- Future goals: new components, physics engine, 3D environment, optimization.