
Alien ECS

Release 0.0.2

Jul 29, 2019

Contents

1	Getting Started	1
1.1	Bare Minimum Code	1
2	Creating a Component	3
3	Creating a Custom Component System	5
4	Creating an Entity Component System	7

ComponentSystemManager -> ComponentSystems -> Components

A component system manager manages component systems, which act on components.

1.1 Bare Minimum Code

The bare minimum to get you started can be found below.

```
#include "ECS.h"

class CustomComponent : public Component
{
public:
    void Update(float dt) override
    { /* Update component */ }
};

int main()
{
    // Init ECS
    ComponentSystemManager<CustomComponent> componentSystemManager;

    // Add a Component
    componentSystemManager.AddComponent(0, new CustomComponent());

    // Update Components
    for (int i = 0; i < 10; ++i)
        componentSystemManager.UpdateSystems(1/60.0f);

    return 0;
}
```

No user-defined component systems need to be defined for basic behavior.

CHAPTER 2

Creating a Component

Extend from the **Component** class and overload the **Update** function.

```
class CustomComponent : public Component
{
public:
    void Update(float dt) override
    { /* Update component */ }
};
```

Creating a Custom Component System

Create a template specialization of the **ComponentSystem** class and inherit from **ComponentSystemInterface**, both templated on your custom component type.

It is important to inherit from the **ComponentSystemInterface** class, so any custom system will plug-and-play with the **ComponentSystemManager**.

```
template<>
class ComponentSystem<CustomComponent> : public ComponentSystemInterface
    ↳<CustomComponent>
{
    /* Custom system code */
};
```

Note: It is important that the file defining the **ComponentSystemManager** object can see your template specializations.

To use a user-defined **ComponentSystem**:

```
auto componentSystem = componentSystemManager.GetComponentSystem<CustomComponent>();
componentSystem->SomeMethod();
```

Creating an Entity Component System

To work with component systems and their components, create a component system manager.

ComponentSystemManager is a variadic templated class that takes user defined components to manage.

As seen below, the class will manage the user defined **CustomComponentA** and **CustomComponentB**.

```
ComponentSystemManager<CustomComponentA, CustomComponentB> componentSystemManager;
```