

DAVE 3605: Final Assignment - Entity Component System

Due on Monday, May 15, 2017

Aryan Naqid

Contents

3	3
Problem 1	3
Problem 2	6

3

Problem 1

Listing ?? shows a Perl script.

Listing 1: Sample Perl Script With Highlighting

```
#pragma once
#include "Entity.hpp"
#include "Component.hpp"
#include "EntityIDPool.hpp"
5 #include "ComponentStorage.hpp"
#include "System.hpp"
#include <memory>
#include <unordered_map>
#include <iostream>
10 #include "MessageHandler.hpp"

class World {
public:
    using EntityArray = std::vector<Entity>;
15    using SystemArray = std::unordered_map<TypeID, std::unique_ptr<BaseSystem>>;

    World();
    /**
     * Initialize the world with a starting number of entities ready for use
     * **/
20    explicit World(std::size_t entityPoolSize);
    /**
     * A world must be explicitly constructed, no copying allowed
     * **/
25    World(const World&) = delete;
    World(World&&) = delete;
    World& operator=(const World&) = delete;
    World& operator=(World&&) = delete;

30

    /**
     * Creates and returns a new entity for use
     * **/
35    Entity createEntity();
    /**
     * Creates a list of entities for use
     * **/
    EntityArray createEntities(std::size_t ammount);

40

    /**
     * Kills the given entity
     * **/
45    void killEntity(Entity& entity);
    void activateEntity(Entity& entity);
```

```

    void deactivateEntity(Entity& entity);

50    bool isValid(const Entity& e) const;

    /**
     * Responsible for attaching entities to their corresponding system
     * Responsible for moving entities between the different caches
55     * Responsible for cleaning entities from systems once they are dead
     *
     *
     * Must be called every update of the game
     * @Todo: Threadsafe
60     * */

    void refresh();

65    template <typename T, typename... Args>
    T& addSystem(Args&&... args);

    template <class RequireList, class ExcludeList = Excludes<>>
    EntityArray getEntities();

70    MessageHandler& messageHandler();

private:
    friend class Entity;
75    EntityIDPool m_entityIDPool;
    SystemArray m_systems;
    MessageHandler m_messageHandler;

    struct EntityAttributes {
80        struct Attribute {
            /**
             * Whether an entity is activated */
            bool activated;
            /**
85             * Whether a given system knows of the entities existence */
            std::vector<bool> systems;
        };

        explicit EntityAttributes(std::size_t ammountOfEntities) :
90            componentStorage(ammountOfEntities),
            attributes(ammountOfEntities)
        {
        }

95        ComponentStorage componentStorage;
        /** Storage of attributes for each entity */

        std::vector<Attribute> attributes;

```

```

100     }
        m_entityAttributes;

        struct cache {
            EntityArray alive;
105            EntityArray killed;
            EntityArray activated;
            EntityArray deactivated;

            void clear() {
110                alive.clear();
                killed.clear();
                activated.clear();
                deactivated.clear();
            }

115            void clearTemporaries() {
                killed.clear();
                activated.clear();
                deactivated.clear();
120            }
        } m_cache;

        void addSystem(BaseSystem* system, TypeID systemTypeId);
        EntityArray getEntities(Filter& filter);
125 };

template <typename T, typename... Args>
T& World::addSystem(Args&&...args) {
130     auto* system = new T(std::forward<Args>(args)...);
    addSystem(system, SystemTypeId<T>());
    return static_cast<T&>(*m_systems[SystemTypeId<T>()].get());
}

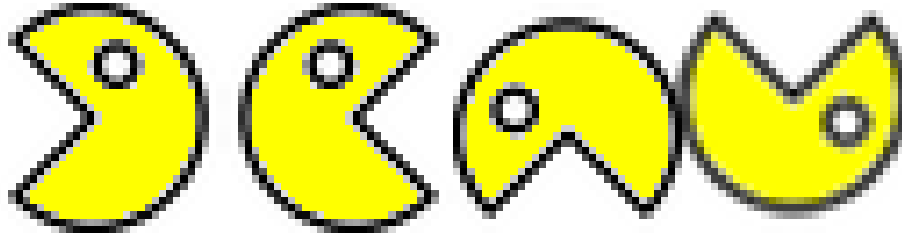
135 template <class RequireList, class ExcludeList> // = Excludes<>>
World::EntityArray World::getEntities() {
    Filter f = MakeFilter<RequireList, ExcludeList>();
    return getEntities(f);
}

```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Problem 2

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.



Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.