

ED1 Components

# Overview

This lab is designed to introduce you to the FSGDEngine and to component design.

# Objective

Create new test objects using the component manager.

There is also a scavenger hunt that is going to require you to read through the game engine code.

# Scavenger Hunt

When working with an existing code base it is important to be able to find your way around, this scavenger hunt is designed to get you to search around the code and look for specific things; you can write your answers here in this document.

1. Find the entry point for the game engine
   1. What is the name of this function?
      * APIENTRY \_tWinMain
   2. Which systems are initialized before the game class is created?
      * EDMemoryManager
      * EDEventManager
2. Find the GameObject class
   1. What is the parent of the game object class?
      * IComponent
   2. What does the enum COMPONENT\_COUNT signify?
      * The number of types of components in the game (7)
3. Find the Thread Pool class
   1. What is the default size for a thread pool (number of threads)?
      * 8
   2. What is the worker example class DoWork() function doing?
      * Find prime numbers in the radix variable.
4. Find the class that represents a vector (the math kind)
   1. There is a structure that allows the data in the vector to be accessed in different ways, what is it and how does it work?
      * Union.
      * Access the vector as an array
   2. What are the preprocessors doing in this file?
      * Including headers and preventing that this class is included more than once in the whole solution

#pragma once

#include <math.h>

include <utility>

1. Find the location of the game engine on your computer
   1. Where is the main executable built at?
      * ..\FSGDEngine-Student\FSGDGame
   2. Where are game assets stored (models, textures, etc)?
      * ..\ FSGDEngine-Student\FSGDGame\Resources\GDAssets
2. Find the class that controls the input console (~)
   1. What kind of container are the commands stored in?
      * set
   2. Which function processes user input?
      * void InputConsole::ProcessKey(char key)
3. Find the class that contains information about the World (game time, screen sizes).
   1. What is the qualifier used for every member?
      * static
   2. What does this usage imply?
      * Every instance share the same variables
      * The variables need to be created before the class and these variables will “last or live” until the program ends
      * You don’t need an instance to access the variable

# Rubric | Grading Breakdown

Commenting should be appropriate and used where necessary.

Your completed lab should be warning free.

Your completed lab should be error/crash free, a submission that crashes will result in a 0%.

Your project should be free of memory leaks.

Failure to follow directions will result in the loss of additional points.

|  |  |
| --- | --- |
| **Objectives** |  |
| Test Objects | 80% |
| Scavenger Hunt | 20% |

# Hints

* Control + F
* Google

# Submissions

Lab is due at **the end** of the lab period. Have a lab instructor grade your assignment; you still need to turn in the assignment on VFILER, grades will be posted on LMS. If you are off campus, late assignments can be turned in through LMS.

This lab must be turned in using the .zip file format using as *LastName.FirstName.lab\_name.zip*.

To create the zip file, run the “make submission folder.bat” file. It will create a folder one level higher called “zip\_this\_and\_turn\_in,” leave it named that. Zip it, rename the zip file and turn it in on VFILER