UF UNIVERSITY of FLORIDA
Herbert Wertheim
College of Engineering

COP3504C

Lab 11: Particle Factory

Overview

In this lab, students will develop a generic singleton-pattern inventory factory for use in a particle system. A program entrypoint (main) will act as the driver of the program. The purpose of this assignment is for student to demonstrate an understanding of and ability to implement the singleton and factory design patterns in C++.

Structure

The provided driver for the program uses **SFML** to display particles on-screen, up to a defined maximum. Particles will be requested from, and returned to, a generic singleton-pattern inventory factory. A cycle occurs on predefined time intervals (in milliseconds). At the beginning of each cycle, a new particle requested from the factory and is displayed on the screen at a random location. If the maximum number of particles has been reached, the oldest particle is removed and returned to the factory for later reuse.

Specification

The `ParticleFactory<T>` singleton class must have these following methods. Its **constructor**, **destructor**, **copy constructor**, and **copy assignment operator** must all be private.

`private ParticleFactory()`
Constructs the singleton instance. If `DEFAULT_INVENTORY_SIZE` is defined, the inventory should be prefilled with that number pointers to dynamically allocated instances of `T`; otherwise, it should be prefilled with `100` pointers to dynamically allocated instances of `T`.

`public ParticleFactory& getInstance()`
Returns a reference to the instance of the `ParticleFactory` singleton.

`public T* getParticle()`
Returns a pointer to an instance of `T`. If the inventory has available elements, an element should be removed from the inventory and returned; if the inventory is empty, a new instance of `T` should be allocated and returned.

`public T* returnParticle(T* element)`
Accepts a pointer to an instance of `T` for recycling, which should be stored in the inventory for later use.

Submissions

NOTE: Your output must match the example output. If it does not, ***you will not receive full credit for your submission***! (Note that matching sample output is necessary, but not sufficient, for full credit.)

Files: ParticleFactory.h
Method: Submit on Canvas