DigiPen
INSTITUTE OF TECHNOLOGY

Final Programming Exam

Module 2 – High-level Programming II

Purpose

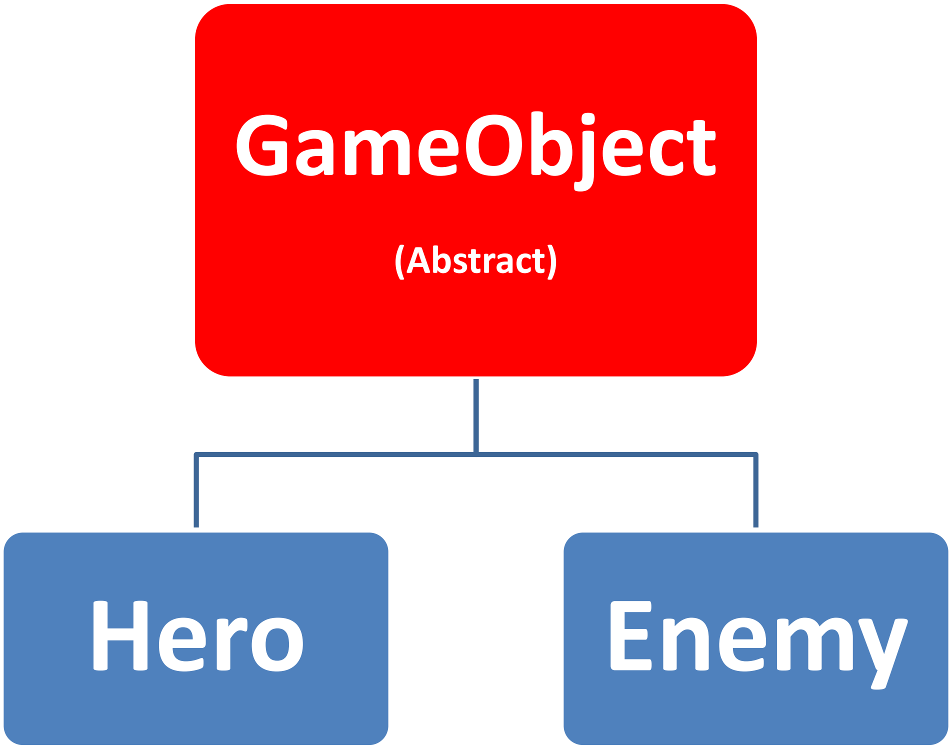
This exam is testing you on most concepts we covered so far in class in object-oriented design and coding (classes, objects, constructors, operator overloading, inheritance, polymorphism, strings, STL containers).

Information

The task is to define and implement the following classes:

- GameObject
- Hero
- Enemy
- ObjectManager

Here is a diagram that shows you the hierarchy that you are required to follow:



Note: Red nodes represent abstract classes.

Here is a description on the above classes:

GameObject Class	
Description	**Abstract** class that serves as the parent of all game objects (Hero and Enemy). All GameObject instances will contain the following properties and methods: ➢ private ***center*** of type Point ➢ private ***name*** of type string ➢ Update ➢ Display method

Hero Class	
Description	Inherits from the ***GameObject*** class. In addition to what is inherited from the ***GameObject*** class, all ***Hero*** instances will contain the following extra properties: ➢ A dynamically allocated array of integers called ***items***. Every element in the array serves as counter that represents how many times the hero collected a certain item. ➢ ***items_count*** that represents the size of the ***items*** array. ***PS: The items array should be dynamically allocated in the Hero's constructor and all its elements set to 0.***

Enemy Class	
Description	Inherits from the ***GameObject*** class. No additional properties.

ObjectManager Class	
Description	The ObjectManager is a class that will hold game objects. It will contain a vector of GameObject pointer. The user of this class will have the ability to: ➢ Add objects to the container ➢ Update all objects in the container ➢ Display all objects in the container ➢ Remove the first occurrence of an object by name ➢ Remove all objects by name ➢ Clear all objects All the above should be possible without having any memory leaks or crashes.

NOTE:

- It is your responsibility to specify if the methods are ***const***, ***virtual*** or ***pure virtual***.
- The tests provided in ***main.cpp*** will guide you in your above decisions and will hint on what methods to add in your classes.
- The provided properties **MUST** stay **private**.

Testing your code

Diff your code's output against the provided outputs. If it matches and you implemented everything sensibly, you're good to go!

What to submit

You must submit the following CPP and Header files:

- ObjectManager.cpp, ObjectManager.h
- GameObject.cpp, GameObject.h
- Hero.cpp, Hero.h
- Enemy.cpp , Enemy.h

in a single .zip file (go to the class moodle page and you will find the submition link).

Do not submit any other files than the ones listed.