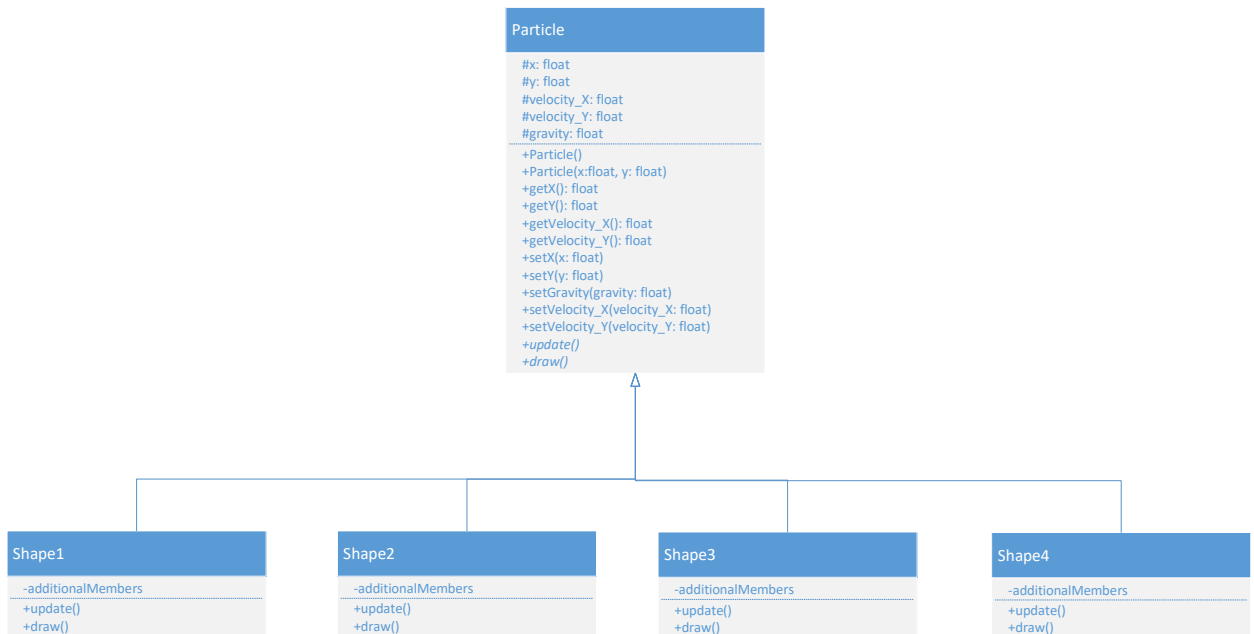


Lab 8: Fireworks Particle System

This lab lets you use **Inheritance and Polymorphism** coding skill combined openFrameworks (OF) library in C++ to create a simple **Particle Engine System** implementing Fireworks effect on your screen.

A particle system is a technique in game physics, motion graphics, and computer graphics that uses a large number of very small sprites, 3D models, or other graphic objects to simulate certain kinds of "fuzzy" phenomena, which are otherwise very hard to reproduce with conventional rendering techniques - usually highly chaotic systems, natural phenomena, or processes caused by chemical reactions. Wikipedia: https://en.wikipedia.org/wiki/Particle_system

1. Create a super class named "Particle" with speed and location attributes, corresponding getters and setters, default constructor, overloaded constructor and two **virtual** public functions: update and draw.
2. There are at least 4 different kinds of 2D objects in the screen. You can design each object freely. For example: some fundamental polygons like rectangle, circle, triangle and at least one object is designed by yourself using ofBeginShape and ofEndShape methods.
3. Each object should have its own class inherited from the super class "Particle". The UML class diagram would look like below.



4. All classes should be defined in best practice (like in separate header and source files, using **initializer list** to define constructors, declare **const member functions** if necessary...).
5. Use any kinds of **resizable sequential data structure** (you are free to use **C++ std library containers** like vector, forward_list, list... or the **ArrayList** or **LinkedList** we have already developed in former labs) to store the particles to create a Particle Engine System simulate a Fireworks effect.
6. Each time your program running, it will have random number of objects with random sizes, speeds moving in the screen and bouncing from the edge of the screen.

7. Design a background.

Grading Rubric:

1. Create at least 5 classes according to the UML class diagram: 25 pts.
2. Define classes in best practice: 10 pts.
3. Design a background: 5 pts.
4. Use a resizable sequential data structure to create a Particle Engine System to simulate Fireworks effect: 30 pts.
5. Proper memory management: 10 pts.
6. No compiling errors: 10 pts.
7. Good Submission: 10 pts.