

# CS-224 Object Oriented Programming and Design Methodologies

Spring 2020

Assignment 4, Feb 23, 2020

## 1 Guidelines

This is the last assignment in this course. You need to present this assignment's work during the class on 17th March 2020.

- You need to do this assignment in a group of two persons
- Only one member of the group will submit the assignment to LMS
- You need to follow the best programming practices as given in the accompanying document and it is also present on LMS. Failure in doing so will have your marks deducted.
- Submit assignment on time; late submissions will not be accepted.
- Some assignments will require you to submit multiple files. Always Zip and send them.
- It is better to submit incomplete assignment than none at all.
- It is better to submit the work that you have done yourself than what you have plagiarized.
- It is strongly advised that you start working on the assignment the day you get it. Assignments WILL take time.
- Every assignment you submit should be a single zipped file containing all the other files. Suppose your group member's name are Sara Khan, id 0022 and Ali Haider, id 033 so the name of the submitted file should be SaraKhan0022\_AliHaider033.zip

- DO NOT send your assignment to your instructor, if you do I will just mark your assignment as ZERO for not following clear instructions.
- You can be called in for Viva for any assignment that you submit

## 2 Game of Pigeons

Pigeons at HU fly around the campus all the time. This game is a fiction play with the pigeons. The attached exe file is running sample of intended game, where the following rules can be observed.

- Clicking on the upper half of screen produces an adult pigeon.
- Clicking on the lower half of screen produces a nest.
- A pigeon lays egg on random interval, currently 2% chances are to lay egg on every frame.
- If an egg is dropped inside a nest, it produces a baby pigeon, that starts flying up-rightwards and increases its size.
- If an egg is dropped on the floor, it just vanishes.
- A baby pigeon starts laying egg when its size is same as adult pigeon.
- Pigeons stop flying as they reach to a certain height in the screen.
- A pigeon is died after laying 5 eggs.
- No object is going beyond the boundaries of the screen.

### 2.1 Core Task

You have to implement this game by using best OOP approaches. In particular, since **Pigeon**, **Egg** and **Nest** are some objects to be drawn on the screen, so a super class, call it **Unit**, can be implemented to take care of drawing aspects for all these objects. Following list of items are the core task for this assignment:

- A link list should be maintained for every type of objects.
- When certain object is destroyed, remove it from its particular link list.
- Inherit the **Pigeon**, **Egg** and **Nest** classes from **Unit** class, and use best OOP approaches to providing the functionality.
- Follow all the game rules given above.

## 2.2 Open-Ended Task

To make the game interesting and fun, apart from implementing the core tasks, you have to implement at least two other features in the game. You have to design these features yourselves. Some possible (not limited to) pointers are:

- When a pigeon reaches the right-most position, it should start flying in left direction.
- As the egg is falling down, it wiggles.
- Eggs take some time to hatch while placed in the nest.
- Play an animation as a pigeon ends its life.
- Play a breaking egg animation when an egg hatches.
- If an egg hits a pigeon, it decreases the pigeon's life.
- Game can be paused and resumed.

## 2.3 Bonus Task (Optional)

If you are really having fun doing this assignment, you can do some extra to earn bonus points. This task will give you absolute 2 marks as bonus. In the upcoming lectures, we'll be studying Polymorphism and Operator-Overloading. If you implement the both in this assignment, you can earn the bonus marks. However, it's up-to you to decide where to implement Polymorphism and Operator-Overloading.

## 3 Some Useful Pointers

- Sample code is there for your benefit. If you are going to use it, understand how it works.
- You do not need to follow the code given exactly. You can make changes where you see fit provided that it makes sense.
- A general rule of thumb is that all attributes in a class are private/protected.
- Where necessary, declare your own functions inside classes. Make sure why you would keep a function as private or public.

- You need to define separate `*.hpp` and `*.cpp` files for all the classes.
- Lazy Foo tutorial are helpful to learn working in SDL <https://lazyfoo.net/tutorials/SDL/index.php>.
- You should take [www.cplusplus.com](http://www.cplusplus.com) and [www.cppreference.com](http://www.cppreference.com) as primary web source to search about C++.
- If you want, you can use pre-built link list in C++. <https://en.cppreference.com/w/cpp/container/list>