What the game framework does

1. The application reads a city.json file which describes a “city” made of “skyscrapers” (simple cuboids) and renders them.
2. The player can fly their camera around the level (using “WSAD” keys and a mouse).
3. Pressing right / left mouse buttons changes the appearance of the crosshair.

Functional requirements – what should the application do?

1. There should be flocks of yellow balls flying among the skyscrapers. Those flocks should behave similarly to birds, meaning:
    - The balls should speed up and slow down,
    - The balls should try to keep themselves together in a flock but they shouldn’t collide with each other (occasional collisions are allowed),
    - The balls should avoid any obstacles (the skyscrapers, the ground and any additional objects that you add),
    - The movement of balls should be reminiscent of the natural movement of birds,
    - There should be more than one flock.
2. Pressing the left mouse button should result in firing a red ball. Red balls should have a big initial velocity, and should be able to bounce off the walls. Additionally red balls should start with a large amount of “energy”. Red balls should accelerate (in any direction) using this “energy” until the “energy” is depleted. The acceleration should be relatively small, with the initial velocity being dominant in the first phase of a red ball’s flight.
3. If a red ball hits a yellow ball it should consume it, increasing its energy level. Red balls should actively pursue yellow balls in order to consume them. Yellow balls should scatter when attacked by a red ball. The system should be properly balanced, for example having the red ball lose its whole energy after consuming several yellow balls. When a red ball loses all its energy, it should become a yellow ball and join the nearest flock.
4. The application should display a “ground” (terrain under the skyscrapers).

The above are the basic requirements. Adding some interesting additional features will be a plus. Your creativity is very important here. You can improve any field that you think is necessary / gives the greatest impact on the gameplay:

- **player movement,**
- **interface – UI,**
- **“action”**
- **additional features**

Use your imagination and try to create cool gameplay. The game should be fun!

Technical requirements – how should application run and how it should be built?

1. Use the framework we've provided. The application should be written in C++ and be compiled in Microsoft Visual Studio. It should run under Windows.
2. Please remember that we will look at the architecture and design of the solution.
3. Application should work smoothly (at least 30 fps – it shouldn't be a slide show). And the faster is the solution, the better.
4. You should provide source code, data and binaries. The project should compile and run without any additional effort.
5. You can use external libraries e.g. for physics to speed up the implementation process but we encourage you to write (as in the case of physics) your own simple collisions.

Additionally, please include a short description of your implementation including any significant challenges that you faced.