Test Assignment

Hello,

Your task is to add a couple of new features to the already existing project. The project is a variation of an Asteroids game. Some parts of the code are pretty advanced while others are fairly simple. We leave it to you to decide what is good and bad and feel free to modify or improve the codebase if you think it is necessary. However, if possible please provide at least a short explanation of why you decided to do so.

This test covers a variety of topics and **you do not need to finish all tasks**. You can do just a few to be good however obviously the more the better.

A few rules:

- Please use all the good programming practices you know.
- If something is hard to implement but you understand the concept please provide an explanation. This is almost as valuable as a fully functional implementation itself.
- You can also modify this list or implement something extra however please be reasonable.
- You can ask questions via email but it may take me some time to respond.
- Please use Unity 2019.4.x (the project is in 2019.4.6f1).

Task 1: Add new functionality

- Currently, the player name is modified through editor. We don't like it. We would like to use JSON instead. We provided you with a proper file and a basic infrastructure. If it is possible to handle the case when the name is equal to null (null means null, not an empty string).
- Our level designers beg us for new tools. We need an editor window that allows us to add an extra asteroid on the screen. Position and rotation should be provided through that window and all other values set to default or random (your choice). Instantiation happens in the editor (not in the runtime). This extra asteroid should take part in the simulation. Additionally, the rotation should be possible to be provided either as degrees or radians. We have provided you with an example method signature in line 349. You can use that or make your own.
- We need to allow the player to control the ship via mouse. Add functionality that makes ship to go to the click location. Make it looks natural as it would be controlled via the keyboard.
- Is it easy to add shooting? If so please add it. If it is technologically possible do not use coroutines.
- If we have shooting then also add a score label (one asteroid = 1 point).
- If it is reasonable, add a unit test or an assertion (one is enough), do not do both unless you are extremely smart.

Task 2: Refactorization and cleanup

- Do you think there is a place to use a Factory pattern? Implement it if it is useful and regardless please provide an explanation of your reasoning.
- What about using Singleton pattern or adding additional layers similar to the DataLayer. Maybe we could make use of that too? Same as above provide reasoning.

Good Luck and Have Fun!