**KF6017 Software Architecture for Games**

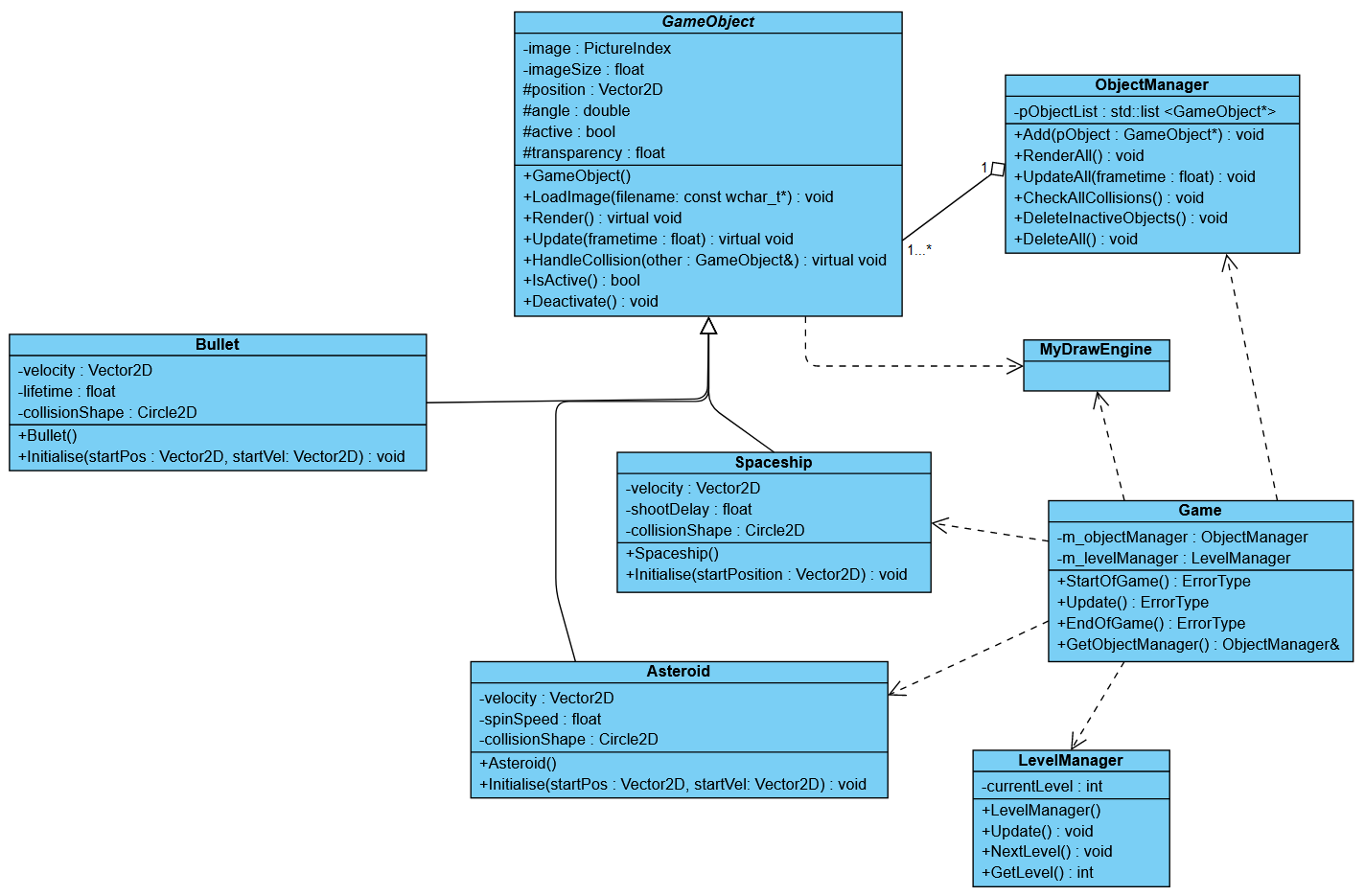
**GAME DOCUMENTATION**

**W18010859**

**Extension: Level system**

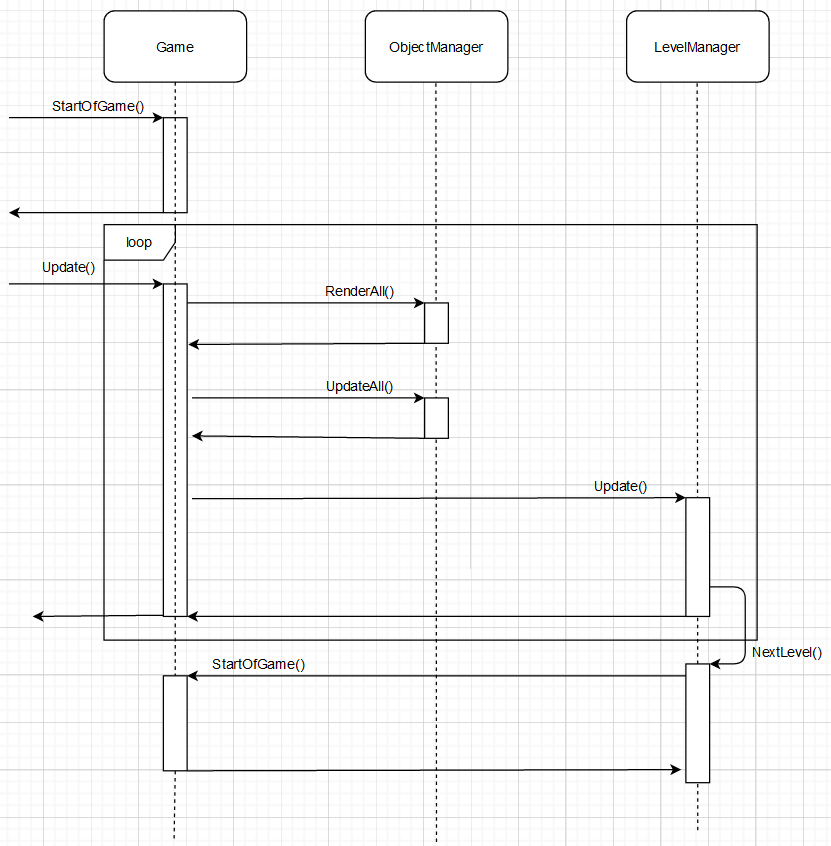
**Word count: 647**

**Class diagram**



*Note: this diagram is available in the submission folder, in the case of the above being too small/hard to read.*

**Sequence diagram**

This diagram shows the dynamic behaviour for the handling of the level system in the game.

*Note: this diagram is available in the submission folder, in the case of the above being too small/hard to read.*

**Key advantages and disadvantages**

**Advantages**

One of the most significant advantages to the implementation of this level system is the fact that it is very decoupled – as can be seen in the class and sequence diagrams above, the LevelManager class does not have any connecting dependencies for any classes except for the main Game class – this is because all that is needed for the level system to work is to return the number of GameObjects in the ObjectManager class, which can be done via the Game class thanks to the GetObjectManager() function. There is also a GetObjectList() function in the ObjectManager class to help make the returning of the size of pObjectList more efficient code-wise in LevelManager. The LevelManager class also needs to call StartOfGame() to restart the levels which can, of course, be done via the Game class.

The dependency on the Game class is also only included in the LevelManager.cpp file, making the dependency a soft dependency and therefore further reducing the coupling of this level system implementation.

Another advantage to the implementation is the scalability of this level system. The developer can easily add functionality to this LevelManager class with both the existing functions as well as new functions. The way the game handles the level incrementation is also done via the StartOfGame() function (with the number of asteroids created for each level) so the developer can also use this function to add more functionality to objects that may be generated at the start of each level. In hindsight it would be better to rename StartOfGame to StartOfLevel() to prevent confusion, but this is the way the developer would be able to tinker with the way the game is generated at the start of every level.

**Disadvantages**

One of the main disadvantages of the implementation of this level system is the fact that the level system does not receive any messages from any specific objects, and instead waits for the call from the ObjectManager that the object list now only has one object in it (the surviving ship). This is something that hinders further development of this level system as the developer is very limited in what they can do. For example, they cannot add a second player into the game, as the current LevelManager can only wait for there to be one object left in the ObjectManager, and if it were waiting for two it would not be able to differentiate between one player and an asteroid, two asteroids and two players. This is a problem in the current system too – since it is not looking for asteroids in particular, the game may progress anyway if the player dies and there is a single asteroid left, since the game thinks there is one object left and therefore advances the level.

Another disadvantage is the inflexibility of the way the game starts the level and the differences between levels – the game simply multiplies the current level number by 5 to get the number of asteroids needed for the player to destroy in that level. This limits the use of more types of objects by the developer and again adds on to the issue of there not being any solid connections between LevelManager and specific objects.

**References**

Class diagram made using Visual Paradigm Online:

<https://online.visual-paradigm.com/drive/#diagramlist:proj=0&new>

Sequence diagram made using Draw.io (could not create one properly using Visual Paradigm Online):

<https://www.draw.io/>

*These are credits to the bitmap assets used in the game.*

1. OpenGameArt.org, *Spaceship.*

<https://opengameart.org/content/spaceship-1>

1. OpenGameArt.org, *Brown Asteroid.*

<https://opengameart.org/content/brown-asteroid>

1. OpenGameArt.org, *Spaceship Bullet.*

<https://opengameart.org/content/spaceship-bullet>