* **This document will contain the answers to the most frequently asked questions in phase 2.**
* **Make sure to check it regularly. We may add more questions in it later.**
* **The document name will be updated regularly to contain the date of the last modification.**
* **We will add any new part at the end of the document, so check anything new written at the end (not in the middle).**
* **Note: all the mentioned solutions are not the only correct solutions. Any other solutions are also accepted if they do NOT break the constraints mentioned in Phase 2’s document (for example, class responsibilities).**

**[ Question #1 ]:**

**How can we implement the overlapping check of ladders with other ladders without breaking the class responsibilities?**

**Answer:**

Ask yourself one question, who is responsible for checking if a newLadder is overlapping with another existing Ladder? The answer is the **Ladder class** itself.

**I suggest the following solution:**

1. Define a **virtual** function in class **GameObject** as follows:



It takes a pointer to another GameObject, **newObj**, that should be checked if it overlaps with the current GameObject (the calling game object). **Think if we should make the function pure virtual or not.**

1. **Override** the previous function in class **Ladder** as follows: first check if the passed “**newObj**” is a Ladder object too and if yes, check if the **passed Ladder** overlaps with the **current Ladder**. → **Check of Overlapping is made Here**
2. Define a function in class **Grid** as follows:



It simply **loops** on the GameObject of each cell in **CellList**, calls the **IsOverlapping()** function of class **GameObject** and returns true if the function call returns true. ***No further logic is made inside the Grid function*** (just looping and calling).