Design Rationale

REQ 1:

I designed the MetalSheet and LargeBolt such that they extend the abstract class Item, since both of them are considered Item in the game. This design helps to achieve the DRY principle by avoiding repeating the code in the Item abstract class. Single-Responsibility Principle is also achieved because each concrete class manages a single specific item.

New item can also be easily added to the game by extending the Item abstract class following this design. It also helps to categorize game entities that are in the item category, this allows us to follow the Liskov Substitution Principle, where the children of Item have all the methods their parent already implemented and those that their parent declared as abstract. Thus, we can easily substitute any child class of Item into any part of the code that expects Item.

There’s no significant downside to this design, as I believe it’s the optimal design to be achieved.