**INFO1113 Assignment 2 Report**

**SID: 500508121**

In the process of designing my code a primary demonstration of object-oriented programming is exhibited in the use of class hierarchy. As can be seen in the UML diagram present in the appendix, the use of multilayered inheritance allowed for a great code efficiency in reducing repeated logic in subtypes. The GameObject class sits at the top of the hierarchy and extends to all objects which may be in the map, including Fruit, Walls, and MoveableObject. The Fruit class further extends to the SuperFruit class and the Waka and Ghost classes both inherit from the MoveableObject super class. Through such implementation, repeated code was reduced through method reusability.

Building onto this, the MoveableObject class is declared as an abstract class as it behaves as a generalisation for the Waka and Ghost classes however there are no instances in which the program needs to instantiation an object of type MoveableObject. The abstract class rather enforces a method implementation for the subtypes within the characteristics which the Waka and Ghost hold common.

Within the abstract class MoveableObject, I have also declared abstract methods for the ‘hasCollided’ and ‘reset’ methods. While the methods are common to both Ghost and Waka types, the logic of the methods differ slightly for each subtype and thus is defined within the subtypes themselves. Without the use of abstract methods, the methods would simply be overwritten by each subtype and would result in a section of unnecessary code in the super class. Each mapObject has the properties; representation (char) which is the value it would be represented with on the map (e.g. ‘1’ = horizontal wall) as well as a sprite path, which is the file path to access the sprite of the object. After implementing this enum, I was able to iterate through the map and for each iteration, iterate through the enum and check whether the character matches the mapObject representation.

**Appendix:**

**Diagram, schematic

Description automatically generated**