Design Rationale

1. Tree and Bush are connected to Fruit even though Location has its own item attribute

Tree and bush will store fruit that it has growing on itself and the Location class will store items that are on the ground.

1. Dinosaur is an interface

Each type of dinosaur has different attributes and interacts differently with the rest of the world (e.g. how it eats fruit). It is easier to having an interface to ensure each dinosaur has the required functionality expected from each dinosaur but the exact details can be separated between each type of dinosaur.

1. Dependency between Player and Location

Our diagram includes that to show that Player will interact with the item ArrayList at Location.

1. Eat is a separate class to EatDino, FeedDino and EatFruit

This is to ensure code is not repeated between the different eat actions (DRY) and to make code easier to maintain in case the implementation of feeding and hunger levels change. Instead of changing all the classes, we can just change the Eat class.

1. Dinosaurs store an age tracker to determine whether it is a baby or not

This will minimise repeated code (DRY) as the baby dinosaur is very, very similar to the adult dinosaur.

1. Egg parent class for each type of egg

Each egg is very similar. This will minimise repeated code (DRY) and make it easier to maintain.