SWEN30006 Project 1

Name1 Yichen Zhang 1299406

Name2 Ruipu Cui 1298198

Name3 Yuhao Liu

This project is mainly based on the game ‘OreSim’. In the graph diagraph given, I can find that the OreSim class controls the game loop, checks for ore completion, and also manages actor movements and UI events. There are many classes connect to the OreSim class which shows a high coupling in GRASP. Also, we think that many function in OreSim is not strongly related to this class and this will lead to low cohesion in GRASP. To maintain the program's operational integrity, new classes were introduced. The following modules were created: Machine and Obstacles.

Machine class is created to help manage the 3 machines we used in the game which are pusher, Bulldozer and Excavator. The reason we create this class is to lower the coupling in OreSim class. This creation of OreSim class to the principle of Pure Fabrication and increases the cohesion of MailItem by reducing its responsibilities. We assign the responsibility to an intermediate object to mediate between other components or services. In our design model, We move some method which managing movement of the character to the machine class. and this will help future reuse the codes. Cause we use extend method to machine class, so it also reduces the need to repeat code in each subclass. And this will also increase cohesion in child class(pusher, buildozer, excavator) because these class extends machine so they only need to focus on their specific behavior. And also we can easily add more machines in the project and we can fix the issue more easily. For machine class, it helps reduce coupling and increase cohesion and is easy to expand and maintain.

In order to improve the management of obstacles in the game, we designed a new class called Obstacle, which allows players to destroy obstacles and implement a variety of other functions. This design is based on the principle of "polymorphism", which allows us to flexibly deal with different obstacles through different implementations of subclasses. And we can extend new types of obstacles without modifying existing code. In addition, this approach improves testability because the behavior of each subclass is independent. This class also helps us to reduce the pressure of OceSim class.It also help lower coupling and higher cohesion

In this assignment, we added some functions which is high related to the OreSim class cohesion and remove some class to machine class and obstacles class. The main idea of doing this is to make it has a high cohesion and a lower coupling.

写代码时特殊case：

Design model



Design Sequence Model

