Tue 9:00 Group 8

Analysis of the current design

The original code has rarely comment, it is hard for people to understand the code. In order to improve, more comments should be added to the code.
There are lots of magic numbers in the original code

In the monsterType, the original code has two attribute Troll and T-X5. When getting the image from the monsterType, if else sentences is needed. When creating more monsterType, the original code will be modified which violets the open closed principle and results in high coupling. However, the UI is highly depended on the monsterType, so the monsterType class is kept in the modified code for simplicity.
To apply the GRASP principles of low coupling and polymorphism, different monster should extend from the same abstract class monster. By doing that, new monster can be easily added as child class of monster class.
Array List can be used to storage all the monster type, in the original code, each monster is set up independently (location and so on), for loop can be used if there is an array list storing all the monsters.
According to information expert, TX5 should be responsible for knowing he should start moving 5 seconds after games starts

New design of the simple version

A new method called setUpSimpleGame is added in the Game class, abstract class "Monster" is created and encapsulates all information about monster. An abstract method WalkApproach is created. Troll and TX5 extends Monster class and implements the method WalkApproach. It reduces coupling between classes. Monster class doesn't need to know how the method WalkApproach is implemented in Troll and TX5. But it can call the WalkApproach method in Troll if the monster is a Troll and vice versa due to polymorphism.
An common method WalkRandom is created in the Monster class as Troll will walk randomly all the time and TX5 will walk random if it hit a wall or visit visited place. Troll and TX5 can call the method WalkRandom in their parent class if needed. This reduces redundant code.

New design of extended version

In the extended version, GameMode interface is added, SimpleGameMode and MultiverseGameMode is created and extends the GameMode. If the game mode is multiverce, if pacMan eats gold, Game will call the abstract method eatGold in GameMode and due to polymorphism, implemented eatGold method in MultiverseGameMode will be called. Same for SimpleGameMode, this results in high cohesion and low coupling between classes.
However, we failed to complete the walkApproach in Wizard class and let it walk randomly like a Troll, this should be improved in the future.