Assignment 2 REQ6: Monologue

Overview:

Due to the design changes for achieving Assignment 3 features, there will be some classes created or modified in Assignment 2's design. (i.e., Monologue and Speakable created in the extended system, Toad and SpeakAction will be modified due to the new design.) Besides, Monologue and Speakable is part of Assignment 3 REQ5, so detailed explanation will be state in Assignment 3 design rationale. The design rationale for each new or modified class is shown on the following pages.

1) SpeakAction

What changed in the design rationale between Assignment 2 and Assignment 3 and Why:

There is no changes in this class of REQ6 between Assignment 2 and Assignment 3.

2) Toad

What changed in the design rationale between Assignment 2 and Assignment 3 and Why:

In Assignment 2, I added all the sentence belongs to Toad into the arraylist: toadTalk, and generate the monologue by using getReplyString() to modify which sentence to be talk depends on different capabilities the player had and different index to be used.

But In Assignment 3, the design is changed, all of the actors can talk every even turn automatically, so the way we get corresponding monologue is change to use Speakable interface and Monologue class instance.

Why I choose to do it that way:

To reduce the repetitive code in those actor's class, I created an interface Speakable and Monologue class to store all the monologues of this game. I let toad implements Speakable, so it can generate monologues by using the default method stated in Speakable and get the correct monologues by using boundary index. By doing so, we don't have to repeat all the methods in every actor's class, this obeys DRY principle.

3) Monologue

Why I choose to do it that way:

By create the monologue class, I can store all the monologues belongs to Toad inside the arraylist of this class. For Toad, it has two ways of speaking, one is when Mario interact with it and another one is generated automatically every evern turn, so by creating an instance of this class, we can get the instance of the monologue class inside the Speakable interface in order to use the boundary index to generate the correct monologues of Toad. For the case of speaking under interaction, we generate the monologue depends on the capability of Mario. For the case of speaking automatically, we generate the monologues randomly from the four sentences. By doing so, we follow the SRP(single responsibility principle) as this class is focus on generate monologues of actors.

4) Speakable

Why I choose to do it that way:

By create a interface called Speakable, toad will implement this interface in order to use the default methods inside to get the correct monologues and speak automatically.

By using this interface, it makes our code less repetitive as we can use its default method in any class which implement this interface. It follows the DRY(don't repeat yourself) principle.

5) Application

What changed in the design rationale between Assignment 2 and Assignment 3 and Why:

There is no changes in this class of REQ6 between Assignment 2 and Assignment 3.

6) Status Enum

What changed in the design rationale between Assignment 2 and Assignment 3 and Why:

There is no changes in this class of REQ6 between Assignment 2 and Assignment 3.