Recommendations for extensions to the game engine

Group Name: Tute08Team139

Members: Tan Jin En, Vernise Ang Veng Yan

World Class

- ## Getter method for GameMaps

  Inside world class, the game map for the game is defined as protected instance variable which does not allow classes outside the package to access the game map. This causes us to use another arraylist of gameMaps in Application class to store the same maps and to allow retrieval of the maps. This is considered as duplication of codes which is unnecessary and it can be avoided if we have a getter method for gameMaps in World class. This also abide by the rules of DRY principle not to repeat codes with same purpose or functionality.

- ## Getter method for actorLocations

  ActorLocations is a protected instance variable that is a hashmap that stores all the location of all actors in the game. However, the lack of getter method causes us to use nested for loop to loop through the map in order to perform finding or searching of an actor. For example, when we need to find a potential dinosaur to mate with another dinosaur in map. Using nested for loop in this case is very inefficient and it will result in the game processing each turn very slow if the map is much larger than the current map. Furthermore, adding a getter method for actorLocations will apply the DRY principle to reduce repeated code to find specific actor when needed.

- stillRunning() to integrate with winAction()

  stillRunning() is used to determine whether the game is still running, which is suitable to be used for detecting the win condition for player. It functions by detecting if player is still in map, if not it will break the loop and game is over, hence the code in player class which calls for winAction() should be coded inside stillRunning(). Only one method should be detecting if the game is running or over, the current implementation of having the checking in both player class and world class is inefficient and is not a good design as only one method should be doing the same purpose.

Actor Class, Ground Class and Item Class

- a class checking method for actor

  Throughout the assignment, we have to check if an actor is the instance of an actor's child class prior to perform specific actions such as mating action for dinosaur. In the forum it is stated that the use of instanceof is a bad practice however it could not be avoided in this case. We can only reduce the repeated code by implementing a method in actor class to perform the instanceof checking. For the same reason as mentioned, we can have methods to

perform the instanceof checking in the ground class and item class as well. This will greatly reduce repeated codes for the instanceof checking.

Display Class

- method to read string

  In the current display class, there is only one method to read user input and it is only reading the first character entered. In the future if we are going to let user input their own choices of game turn limit and EcoPointsGoal, then a method to read string in display class is necessary. Hence, a method to read string is highly recommended to be included in display class so that the future maintenance in getting user input will be easier.