***Sequence diagram - RestAction.execute()***

This execute() method of the RestAction class interacts with three other classes, including ResetManager, Status, and Resettable. The ResetManager class interacts with the lifeline of this RestAction class through the invocation of its getInstance() method, which returns an object of its own class, named “resetManager”. The enum class Status is involved in this lifeline since REST is passed in as an argument of the run() method of the ResetManager class. The RestAction class’s lifeline interacts with this object by invoking its run() method. Throughout the lifeline of this resetManager object, the self message symbol is used to represent the invocation of its own method, cleanUp(). The loop fragment is used to model the iteration through resettableList. Since each resettable element of resettableList implements the Resettable interface, this interface interacts with the object resetManager’s lifeline.

***Sequence diagram - SoftResetAction.execute()***

All interactions between SoftResetAction and other classes, including ResetManager, Status, and Resettable, are the same as above. The only difference is that SOFT\_RESET is passed in as an argument of the run() method of the ResetManager class.

***Sequence diagram - Player.resetInstance()***

This resetInstance() method of Player class interacts with three other classes, including GameMap, Location, and Token. The self message symbol represents the invocation of its own attributes, methods or local objects that are from the same lifeline as Player class. Those attributes include “hitPoints”, “maxPoints” and “lastBonfire”, methods include “transferSouls()”, and local objects include “x”, “y” and “location” which are created within this lifeline. The loop fragment is used to model the iteration through this.getInventory(). The optional execution fragments symbolize that if the condition is met, the code within this fragment will be executed, else, continue. The lifeline of Estus Flask class begins and ends within this loop fragment since it interacts with the Player class’s lifeline only when the right item is found to perform downcasting and invoke its resetChargeCount() method. The object of class GameMap which is passed in as an argument and named “map”, interacts with the Player class’s lifeline through the invocation of locationOf(), at(), and moveActor() method. This method returns an object of the Location class, named “location”. The Player class’s lifeline interacts with this location object by invoking its x(), y(), and addItem(token) method. The alternative fragment describes only one of the options, either the status is REST or SOFT\_RESET, will be executed. The Token class interacts with this lifeline as its getInstance() method is invoked and returns an object of its class, which is named as “token”. Lastly, the attribute lastToken that is of Location class is involved in this lifeline through the invocation of its removeItem() method. Self message symbol is not used here since we are not only calling on the attribute itself, but the method of the attribute’s type class. Enum class is not presented here.