**SAAUREN M – ASSIGNMENT 3 DESIGN RATIONALE**

**Mambo** **Marie**

The Mambo Marie feature was implemented into the game. To represent a MamboMarie object, a class named “MamboMarie” was created. This class extends from the “ZombieActor” class. By doing this, we can easily access and use critical methods that are already defined for us in the ZombieActor class, and therefore also the Actor class. An alternative to this design decision may be to have the MamboMarie class inherit Zombie rather than ZombieActor.

This was not done as the Zombie class does not have a constructor (by default or from previous assignments) where we can choose the display character, hitpoints and other attributes. We could overload the constructor in Zombie which will let us do this, however I decided that the Zombie class did not have any methods that would be useful to MamboMarie and ZombieActor already had this sort of constructor defined.

The class MamboMarie, like other classes such as Zombie and Farmer uses an array to hold a set of behaviours. These behaviours are MarieSpawnBehaviour, ChantBehaviour, and WanderBehaviour.

In order to implement MamboMarie appearing 5 percent of the time, the behaviour MarieSpawnBehaviour is created. This class implements the Behaviour interface. By implementing an interface, we can easily attain methods that we will need in order to develop this behaviour such as the getAction() method. This is good practice as this interface requires that all methods that implement it are required to have it’s methods, so we can uphold consistency amongst all Behaviours.

The MarieSpawnBehaviour class implements the getAction method which returns a MarieSpawnAction, or null if it is not. A helper class to determine a random edge location getRandEdgeLocation is utilized which determines and returns a random Location on the top edge of the map. This method is called in the getAction action method. An alternative for this would be to determine the random location inside of getAction, however this would be bad design practice as this means that the code is not modularised and therefore more difficult to interpret. In this manner, using a helper method makes the code more readable.

**Ending the game**