Slow Epic, Speed Ghost

Edward Seim

December 19, 2014

1 Summary

This was my first real experience with making a powerup class for EpicMan. we created a generic PowerUp class that we could extend so we'd have a general idea of how to go about maing powerup classes that would be compatible in their implementation in the game. As shown in the Detailed image, the most interesting part of this class is the programming of a foreach loop in Blueprint. First we ask the game for all Actors of class Ghost, which returns an array of actors. Feeding that into the foreach loop, we then get each actor one at a time. An execution line is available for running each element of the array, and a final execution line runs when the loop is complete, which makes perfect sense from a visual programming persepective. The Cast to Ghost block sets up all the pointers so the editor and complier understand the variables we are trying to pull from the Ghost. This was also my first encounter with setting variables that are not in the class I am working on. To set variables that are in other classes, say an Actor, you only get access to the set by draggin out from the class pointer. This means that the set Method has a tie back to the Target class. A little arithmetic makes it so EpicMan slows down and the Ghosts speed up while the PowerUp is enabled, and a delay block lets the PowerUp take its sourse, restoring the original speeds before it is deleted.

2 Code Snapshot