TestDamageValue

- This test ensures that the enemies have their correct damage value that will be used to update the monument health. This is relevant to the M4 implementation since we want to ensure enemies have damage values that can result in a gameover once the monument loses all its health and this test checks if there is such a damage value.

testMonumentHealth

- This test checks if the monument health is updated properly after an enemy deals damage to it. The test creates an enemy and has it damage the monument. It then checks if the damage was properly applied to the monument. This is important to the M4 implementation; the monument must take damage from enemies once they finish their route. This method checks if that damage is correct and updates health properly.

testMoneyValues

- This test checks if the enemies have their money value attributed to them that will be given to the player once defeated. This is important to the overall game since enemies will be the main source of income for the player as they progress rounds.

testEnemyPosition

- Ensures that enemies are properly initialized to the beginning of the path. This is to make sure that enemies do not begin off the path or take paths that are not intended. This is relevant since we want the enemy to be on a known path that we can then check to see if an enemy exists there.

testEnemyHealth

- This test checks to see if enemies are initialized with proper health values. The test creates each enemy type and then checks to see if the health values associated with each enemy are correct. This is relevant to the M4 implementation because enemies can only exist when the health is above zero, so health values need to be correct so enemies can spawn.

testEnemyScore
testEnemyAddScore
testEnemyRemoveScore

- These series of tests deal with a score that is added to the player or removed when dealing with an enemy. The first checks and ensures that all enemies have a score value attributed to them. The second checks and sees that when an enemy is defeated, it adds to the players score properly and the third checks that once an enemy makes it to the end of the path, will remove some points from the score. This is important and relevant to the M4 implementation as we plan to have a total score screen the game over part of the game that displays how well or poorly the player did which depends on enemies having this score value.

testLoseGame

- This test checks if the game can be lost. This is done by having damage dealt to the monument until it is below or equal to 0, then checking if the game has been lost. This is relevant to the M4 milestone because once a monument reaches 0 or less health, the game is meant to end. This test ensures this occurs.

testPaths

- This test checks if the paths of each enemy are correct. This is done by taking the paths of each of the three enemy types and comparing it to a premade correct path. This is relevant to the M4 milestone because enemies are only supposed to travel along a specific path. This test ensures this is the case.