

White Box Testing - Assessment 4

Green = Team Craig Additions

Test Class	Test Name	Test ID	Description	Result	Req
CharacterTest	charactersWithSamePositionShouldCollide	1.1	Creates one character and checks it collides with itself.	PASS	
	touchingCharactersShouldCollide	1.2	Creates two characters than are just touching to ensure they collide. This is the most likely situation to occur in game.	PASS	
	nonTouchingCharactersShouldNotCollide	1.3	Creates two characters that should not collide and checks this is the case.	PASS	
	getCenterOnCharacterWithPositivePosition	1.4	Tests the calculation for getCenter(), on a character in the top right quadrant, which should return the absolute position of the centre of the character's sprite. It assumes the character sprite is 32x32, as it should be.	PASS	
	getCenterOnCharacterWithNegativePosition	1.5	Negative test as no character should ever have a negative position in the x or y direction. Again testing the calculation for getCenter().	PASS	
	getDirectionInTopRightQuadrant	1.6	Testing the getDirection method for each possible quadrant. All of which are possible as the direction is a bearing relative to the	PASS	
	getDirectionInBottomRightQuadrant	1.7	character's center.	PASS	
	getDirectionInBottomLeftQuadrant	1.8		PASS	
	getDirectionInTopLeftQuadrant	1.9		PASS	
	charactersTakeSpecifiedDamage	1.10	Calling takeDamage() on a character and testing if the expected hit points are lost from the player.	PASS	
	getDirNormVectorToNegativePosition	1.11	Testing that getDirNormVector calculates the correct normalized vector from the character's center to a negative coordinate.	PASS	

	getDirNormVectorToPositivePosition	1.12	Testing that getDirNormVector calculates the correct normalized vector from the character's center to a positive coordinate.	PASS	
PlayerTest	playerPositionResetsWhenRespawned	2.1	Check that the original position of the player, is the same as the position after moving it then respawning it.	PASS	
	playerDoesNoDamageToZombieWhenAtMaxRange	2.2	Creates a player and a zombie where the zombie at the maximum range away from the player in the direction that the player is facing. Then the player attacks the zombie and we check that the health does not decrease.	PASS	F10
	playerDoesDamageToZombieWhenInRange	2.3	The same as above but the zombie's distance from the player is less than the maximum range. We expect the zombie's health to decrease.	PASS	F10
	playerDoesNoDamageToZombieWhenOutOfRange	2.4	The same as above but the zombie's distance from the player is greater than the maximum range. In this case the player should not do damage to the player.	PASS	F10
	playerTypesHaveDifferentHealth	2.5	Save the health of a nerdy student in a variable then respawn the player as a sporty student and check that they have a different amount of hit points.	PASS	F3
	playerTypesHaveDifferentSpeed	2.6	Save the speed of a nerdy student in a variable then respawn the player as a sport student and check that they have different speed.	PASS	F3
ZombieTest	zombieDoesNoDamageToPlayerWhenAtMaxRange	3.1	The same as 2.1.1 but switch player and zombie positions and checking that the zombie does not do damage to the player.	PASS	F9
	zombieDoesDamageToPlayerWhenInRange	3.2	The same as 2.1.2 but switch player and zombie positions and checking that the zombie does damage to the player when in range.	PASS	F9
	zombieDoesNoDamageToPlayer	3.3	The same as 2.1.3 but switch player and zombie positions and checking	PASS	F9

	WhenrOutOfRange		that the zombie does no damage to the player.		
	zombieCannotAttackBeforeCooldownComplete	3.4	The zombie tries to attack the player twice in rapid succession. The player should only take damage from the first attack.	PASS	F9
	zombieCanAttackAfterCooldownComplete	3.5	The zombie tries to attack the player twice but with a pause longer than the zombies cooldown time between the attacks. The player should take damage from both attacks.	PASS	F9
	differentZombieTypesHaveDifferentHealthStats	3.6	Certain zombie types have a variation in health attribute value.	PASS	F4
	differentZombieTypesHaveDifferentSpeedStats	3.7	Zombie types have a variation in health attribute value.	PASS	F4
	differentZombieTypesHaveDifferentSpeedStats	3.8	Zombie types have a variation in health attribute value.	PASS	F4
PowerUpTest	powerUpHealthAddsHPToPlayer	4.1	Reduces player health then activates a health power up and checks that the player's health goes up by the amount specified by Constant.HEALUP.	PASS	
	powerUpSpeedIncreasesPlayersSpeed	4.2	Compares the player's speed before and after activating a speedpower up to make to sure the speed increases by the amount specified by Constant.SPEEDUP.	PASS	
	powerUpSpeedDeactivatesAfter10s	4.3	Compares the player's speed before activating it and 11 seconds after it has been activated to make sure the speed is the same as the original speed.	PASS	
	powerUpSpeedDoesNotDeactivateBefore10s	4.4	Compares the player speed before activating and 9s after activating to make sure the speed is still different.	PASS	
	powerUpSpeedDeactivateMethodResetsPlayerSpeed	4.5	Tests that the speed power ups effect can be cancelled at any time by calling deactivate 0 manually.	PASS	
	playerCannotPickUpFarAwayPowerUp	4.6	Checks the player can't pick up a power up that is out of reach(must be overlapping) by using the overlapsPlayer() method of the PowerUp class.	PASS	

	playerCanPickUp ClosePowerUp	4.7	Checks the player can pick up a power up that is in reach (must be overlapping) by using the overlapsPlayer() method of the PowerUp class.	PASS	
	powerUpImmunityStopsThePlayer TakingDamage	4.8	Activates an immunity power up and calls takeDamage on the player. Checks that the players health before and after take Damage remains the same.	PASS	
	powerUpImmunityDeactivatesAfter 5s	4.9	Activates an immunity power up and calls takeDamage before and after 5 seconds. Checks the the player only lost hit points from take Damage called after 5 seconds.	PASS	
	powerUpImmunityDeactivateMethod CancelsImmunity	4.10	Activates an immunity power up and calls takeDamage on the player before and after calling deactivate on the power up. Checks the player only lost hit points from the takeDamage called after deactivate.	PASS	