Unit Tests

S. No	Action	Inputs	Expected Output	Actual Output	Test Result
Player					
PU_1	Test the players position methods (setters and getters)	int x and int y	100, 100	100, 100	Pass
PU_2, PU_3	Testing the players life system by checking if the player is out of lives	void	False, True	False, True	Pass
PU_4, PU_5, PU_6, PU_7	Testing the players score system by using the addScore, decreaseScore, and resetScore methods	Int score (the score to be addded/decreased)	0, 100, 50, 0	0, 100, 50, 0	Pass
PU_8, PU_9	Testing the players movement by setting the setUp, setDown, setRight, and setLeft methods then calling update() to advance the players position one cycle	Boolean b (true/false)	The players position: After moving down: 102 After moving right: 102 After moving up: 100 After moving left: 100	102, 102, 100, 100	Pass
Reward					
RU_1 RU_2	Testing the player's score increased by reaching the reward and bonus reward,	Int score(the score to be increased)	Player reach reward and get: 1 Player reach bonus reward and get: 1	1 1	Pass

Map					
MU_1 MU_2	Testing the map’s ability in detecting walls Testing if the map is recognizes not a wall	Int x and int y Int x and int y	True False	T1. Null Pointer exception T2. True False	Error Passed
Enemy					
EU_1	Test the enemies position(setters and getters)	Int x and int y	4, 8	4, 8	Passed
EU_2	Testing correct respawn positions	void	4, 8	4, 8	Passed
ExitPoint					
EPU_1	Testing if the exit point is correctly spawning	Void	4,8	4,8	Passed
EPU_2	Testing if the exit point becomes non-active	Void	False	False	Passed
EPU_3	Testing if the exit point becomes set to active	Void	True	True	Passed
EPU_4	Testing if the exit point is returning the correct coordinates	Void	4, 2	4, 2	Passed
GameTimerTest					
GMU_1	Testing that the Timer is initialized	void	Timer initialized	Timer initialized	Passed

GMU_2	Testing that the time given in seconds making	void	Returns 0	Returns 0	passed
GMU_3	Testing that the time is not negative	void	True	True	Passed
GMU_4	Testing that the timer is working and is not null	void	Return not null value, timer created	Returns not null value	Passed
Door					
usedoor	Testing that door is closed	void	False	False	Passed
setTrap	Testing that trap is setup	void	True	True	Passed
hasTrap	Testing that whether there has a trap	void	False	False	Passed
respawnDoor	Testing after door respawn and it is closed	void	True	True	Passed
isClosed	Testing if the door isclosed after door respawn	void	True	True	Passed
Get door location	Testing if the door location is correct	void	x=10 y=10	x=10 y=10	Passed
getDoorID	Testing if the ID is correct	void	0	0	Passed

Integration Tests

S.no	Action	Inputs	Expected Output	Actual Output	Test result
Player					
1	Player touches reward	Reward	Reward collected Score increase Inventory increase	The score increased	Passed
2	Player touches bonus reward	BonusReward	Bonus reward collected Score increase	The score increased	Passed
3	Player touches trap	Trap	Trap Collected Decrease score	The score decreased	Passed
	Player touches trap from above	Player move from above	Trap trigger Trap collected and decrease score, player lost life	Trap trigger Trap collected and decrease score, player lost life	Passed
	Player touches trap from below	Player move from below	Trap trigger Trap collected and decrease score, player lost life	Trap trigger Trap collected and decrease score, player lost life	Passed
	Player touches trap from left	Player move from left	Trap trigger Trap collected and decrease score, player lost life	Trap trigger Trap collected and decrease score, player lost life	Passed
	Player touches trap from right	Player move from right	Trap trigger Trap collected and decrease score, player	Trap trigger Trap collected and decrease score, player	Passed

			lost life	lost life	
P_I4	Test that the player detects the door from below, test if not from below above, test if not from above right, test if not from right left, test if not from left	Door	Return True Return False Return True Return False Return True Return False Return True Return False	Return True Return False Return True Return False Return True Return False Return True Return False	Passed
5	Test the players interaction with the ExitPoint from all directions (up, down, left, right)	ExitPoint	Game endable Condition is true		
Enemy					
E_I1 E_I2 E_I3	Testing Enemy Hit Player so Player position respawns Enemy position respawns and Player decrease life	Player	4,4 200, 201 2	4,4 200,201 2	Passed
E_I4 E_I5 E_I6 E_I7	Test that the enemy detects the door from below, test if not from below above, test if not from above right, test if not from right left, test if not from left	Door	Return True Return False Return True Return False Return True Return False Return True Return False	Return True Return False Return True Return False Return True Return False Return True Return False	Passed

E_I8	Test Enemy moving when player moving	Player	X != 4 Y != 8 X != 4 Y != 8	T1. Null Pointer Exception T2. Enemy moves	Failed Passed
E_I9	Test The enemy not moving when player is moving	Player	4,8 4,8	4,8 4,8	Passed
MenuState					
MS_1	Player selects item	Game Manager	Gm.currentstate should be specified state		
Applies to all others					