Test_ID	Input	Expected Output	Actual Output	Description	Related Requirements	Category	Author	Status
BulletDispenser_1.1	N/A	Null	Null	Testing that the BulletDispenser constructor is working as intended with standard input	UR_FIRETRUCK_REPAIR UR_ET_UNIQUE_SPEC UR_ET_IMPROVEMENT UR_LOSS_CONDITION UR_DIFFICULTY_LEVEL UR_INTUITIVE UR_FORTRESS_ATTACK	Functional, Unit	Bruno Davies	Pass
BulletDispenser_1.2	A mocked Pattern class	The time between firing patterns - 'PatternTime'	1f - the time given	Testing that when passed a pattern and with one pattern in the List, the PatternTime is set to the correct value	UR_FIRETRUCK_REPAIR UR_ET_UNIQUE_SPEC UR_LOSS_CONDITION UR_DIFFICULTY_LEVEL UR_INTUITIVE UR_FORTRESS_ATTACK	Functional, Unit	Bruno Davies	Pass
BulletDispenser_1.3	A mocked Pattern class	The same mocked Pattern Class as inputted - 'FiringPattern'	The mocked Pattern class	Testing that the pattern passed is added and that the correct FiringPattern	UR_FIRETRUCK_REPAIR UR_ET_UNIQUE_SPEC UR_LOSS_CONDITION UR_DIFFICULTY_LEVEL UR_INTUITIVE UR_FORTRESS_ATTACK	Functional, Unit	Bruno Davies	Pass
BulletDispenser_1.4	Two mocked Pattern classes	The 'PatternTime' from the first passed Pattern class	1f - the time given	Testing that the if a second pattern was added the 'patternTime' do not change	UR_FIRETRUCK_REPAIR UR_ET_UNIQUE_SPEC UR_LOSS_CONDITION UR_DIFFICULTY_LEVEL UR_INTUITIVE UR_FORTRESS_ATTACK	Functional, Unit	Bruno Davies	Pass
BulletDispenser_1.5	Two mocked Pattern classes	The first passed Pattern class - 'FiringPattern'	The first passed Pattern class	Testing that the if a second pattern was added the 'firingPattern' do not change	UR_FIRETRUCK_REPAIR UR_ET_UNIQUE_SPEC UR_LOSS_CONDITION UR_DIFFICULTY_LEVEL UR_INTUITIVE UR_FORTRESS_ATTACK	Functional, Unit	Bruno Davies	Pass
Entity_1.1	N/A	Correct initialized variables	The correct initialized variables - True for isAlive - 100 for health - (1,2) for position	Testing that the Entity constructor is working as intended with standard input	UR_FIRETRUCKS_UNIQUE_SPEC UR_FIRETRUCKS_REFILL UR_FIRETRUCK_REPAIR UR_ET_UNIQUE_SPEC UR_ET_IMPROVEMENT UR_FIRETRUCK_MIN_START UR_ET_MIN_START UR_WIN_CONDITION UR_LOSS_CONDITION UR_ET_DESTROYS_STATION UR_MINIGAME UR_HIGHSCORE UR_PATROLS UR_FORTRESS_ATTACK	Functional, Intergrated, Unit	Bruno Davies	Pass
Entity_1.2	Int value into .applyDamage	Entities health minus the input value	entityTest health minus the input value - 50	Testing that applyDamage removes the amount of health with standard value - with not killing	UR_FIRETRUCKS_UNIQUE_SPEC UR_FIRETRUCKS_REFILL UR_FIRETRUCK_REPAIR UR_ET_IMPROVEMENT UR_WIN_CONDITION UR_LOSS_CONDITION UR_ET_DESTROYS_STATION UR_HIGHSCORE UR_FORTRESS_ATTACK	Functional, Intergrated, Unit	Bruno Davies	Pass
Entity_1.3	Int value into .applyDamage greater than Entities max health	Entities health minus the input value - negative value	entityTest health minus the input value - '-10'	Testing that applyDamage removes the amount of health with standard value - does kill - negative health	UR_FIRETRUCKS_UNIQUE_SPEC UR_FIRETRUCKS_REFILL UR_FIRETRUCK_REPAIR UR_ET_IMPROVEMENT UR_WIN_CONDITION UR_LOSS_CONDITION UR_ET_DESTROYS_STATION UR_HIGHSCORE UR_FORTRESS_ATTACK	Functional, Intergrated, Unit	Bruno Davies	Pass
Entity_1.4	Int value into .applyDamage equal to the Entities max health	Zero	Zero	Testing that applyDamage removes the amount of health with standard value - does kill - at zero health	UR_FIRETRUCKS_UNIQUE_SPEC UR_FIRETRUCKS_REFILL UR_FIRETRUCK_REPAIR UR_ET_IMPROVEMENT UR_WIN_CONDITION UR_LOSS_CONDITION UR_ET_DESTROYS_STATION UR_HIGHSCORE UR_FORTRESS_ATTACK	Functional, Intergrated, Unit	Bruno Davies	Pass

Entity_1.5	Int value into .applyDamage that is negative	An IllegalArgumentException to be thrown	IllegalArgumentException	Testing that applyDamage should throw IllegalArgumentException when passed a negative number	UR_FIRETRUCKS_UNIQUE_SPEC UR_FIRETRUCKS_REFILL UR_FIRETRUCK_REPAIR UR_ET_IMPROVEMENT UR_WIN_CONDITION UR_LOSS_CONDITION UR_ET_DESTROYS_STATION UR_HIGHSCORE UR_FORTRESS_ATTACK	Functional, Intergrated, Unit	Bruno Davies	Pass
Entity_1.6	N/A (positive health)	True when check if it isAlive()	True	Testing that isAlive works with a standard positive health	UR_WIN_CONDITION UR_LOSS_CONDITION UR_ET_DESTROYS_STATION UR_HIGHSCORE UR_FORTRESS_ATTACK	Functional, Intergrated, Unit	Bruno Davies	Pass
Entity_1.7	Int value between 0 and max health but not equal	True when check if it isAlive()	True	Testing that isAlive works after Entity takes damage but not kills	UR_WIN_CONDITION UR_LOSS_CONDITION UR_ET_DESTROYS_STATION UR_HIGHSCORE UR_FORTRESS_ATTACK	Functional, Intergrated, Unit	Bruno Davies	Pass
Entity_1.8	Int value equal to max health	False when check if it isAlive()	False	Testing that isAlive returns false when health is zero	UR_WIN_CONDITION UR_LOSS_CONDITION UR_ET_DESTROYS_STATION UR_HIGHSCORE UR_FORTRESS_ATTACK	Functional, Intergrated, Unit	Bruno Davies	Pass
Entity_1.9	Int value greater than max health	False when check if it isAlive()	False	Testing that isAlive returns false when health is negative	UR_WIN_CONDITION UR_LOSS_CONDITION UR_ET_DESTROYS_STATION UR_HIGHSCORE UR_FORTRESS_ATTACK	Functional, Intergrated, Unit	Bruno Davies	Pass
StatBar_1.1	N/A	Correct initialized variables	The correct initialized variables - height of 3	Testing that the constructor initializes StatBar correctly by checking values	UR_FIRETRUCK_REPAIR UR_ET_UNIQUE_SPEC UR_ET_IMPROVEMENT UR_INTUITIVE		Bruno Davies	Pass
WaterStream_1.1	N/A	Correct initialized variables	The correct initialized variables - position of (0,0)	Testing that the constructor for WaterStream works correctly	UR_FIRETRUCKS_UNIQUE_SPEC UR_FIRETRUCKS_REFILL UR_ET_UNIQUE_SPEC UR_MINIGAME UR_INTUITIVE		Bruno Davies	Pass
GameObject_1.1	N/A	Correct initialized variables	The correct initialized variables	Testing that the GameObject Constructor is working as intended with a standard input	UR_FIRETRUCKS_UNIQUE_SPEC UR_ET_UNIQUE_SPEC UR_ET_MIN_START UR_DIFFICULTY_LEVEL UR_INTUITIVE UR_PATROLS	Functional, Intergrated, Unit	Bruno Davies	Pass
GameObject_1.2	N/A	The 'middle' between the position vector and the top-right corner (calculated by width and height)	A vector between the position vector and the width and height - (25,25)	Testing that when the GameObject is at location (0,0) .getCentre() calculates the correct centre	UR_FIRETRUCKS_UNIQUE_SPEC UR_ET_UNIQUE_SPEC UR_ET_MIN_START UR_PATROLS	Functional, Intergrated, Unit	Bruno Davies	Pass
GameObject_1.3	New position vector	The 'middle' between the new position vector and the top-right corner (calculated by width and height)	A vector between the position vector and the width and height - (75,75)	Testing that wen GameObject is at any 'middle' location .getCentre() calculates the correct centre	UR_FIRETRUCKS_UNIQUE_SPEC UR_ET_UNIQUE_SPEC UR_ET_MIN_START UR_PATROLS	Functional, Intergrated, Unit	Bruno Davies	Pass
GameObject_1.4	N/A - Another instance of GameObject with (0,0) as dimensions	The centre calculated when dimensions (width / height) are (0,0)	A vector between the position vector and the width and height - (0,0)	Testing that when the dimensions are (0, 0) . getCentre() calculates the correct centre	UR_FIRETRUCKS_UNIQUE_SPEC UR_ET_UNIQUE_SPEC UR_ET_MIN_START UR_PATROLS	Functional, Intergrated, Unit	Bruno Davies	Pass
GameObject_1.5	New position of positive standard numbers - (500, 500)	The new position of GameObject to be what is set	The input - (500, 500)	Testing that .setPosition() accepts standard values and changes accordingly	UR_FIRETRUCKS_UNIQUE_SPEC UR_ET_UNIQUE_SPEC UR_ET_MIN_START UR_PATROLS	Functional, Intergrated, Unit	Bruno Davies	Pass
GameObject_1.6	New position of two negative numbers - (-10, -10)	The new position of GameObject to be what is set	The input - (-10, -10)	Testing that .setPosition() accepts negative values and changes accordingly	UR_FIRETRUCKS_UNIQUE_SPEC UR_ET_UNIQUE_SPEC UR_ET_MIN_START UR_PATROLS	Functional, Intergrated, Unit	Bruno Davies	Pass
GameObject_1.7	changePosition with input of standard standard positive numbers - position (50, 50) and the change (100, 100)	The position before changePosition plus the values passed	The addition of position and change - (150, 150)	Testing that .changePosition() changes correctly with standard values (.changePosition changes current position by vector v)	UR_ET_MIN_START UR_PATROLS	Functional, Intergrated, Unit	Bruno Davies	Pass

GameObject_1.8	changePosition with input of negative numbers - position (50, 50) and the change (-25, -25)	The position before changePosition plus the values passed	The addition of position and change - (25, 25)	Testing that .changePosition() changes correctly with an all negative vector input	UR_ET_MIN_START UR_PATROLS	Functional, Intergrated, Unit	Bruno Davies	Pass
GameObject_1.9	changePosition with input of one positive and one negative number - position (50, 50) and the change (25, -25)	The position before changePosition plus the values passed	The addition of position and change - (75, 25)	Testing that .changePosition() changes correctly with one part of the vector being negative	UR_ET_MIN_START UR_PATROLS	Functional, Intergrated, Unit	Bruno Davies	Pass
FireTruck_1.1	N/A	Correct initialized variables	The correct initialized variables - via getCentre()	Testing that the FireTruck constructor is working as intended with standard input	UR_FIRETRUCKS_UNIQUE_SPEC UR_FIRETRUCKS_REFILL UR_FIRETRUCK_REPAIR UR_FIRETRUCK_MIN_START UR_ET_MIN_START UR_WIN_CONDITION UR_ET_DESTROYS_STATION UR_DIFFICULTY_LEVEL UR_HIGHSCORE	Functional, Intergrated, Unit	Bruno Davies	Pass
FireTruck_1.2	N/A (The initialized 'Direction')	New centre/position in relation to speed, direction and time in only the Y direction	A shift in the Y, not X, direction the correct amount to predetermined calculations - (12.5,25.01648)	Testing that moveInDirection changes the Y position but not the X when no collision when Direction set to N (0)	UR_FIRETRUCKS_UNIQUE_SPEC UR_WIN_CONDITION UR_ET_DESTROYS_STATION UR_INTUITIVE	Functional, Intergrated, Unit	Bruno Davies	Pass
FireTruck_1.3	East ('E') direction - 270	New centre/position in relation to speed, direction and time in only the X direction	A shift in the X, not Y direction the correct amount to predetermined calculations - (12.517002,25.0)	Testing that moveInDirection changes the X position but not the Y when no collision when Direction set to E (270)	UR_FIRETRUCKS_UNIQUE_SPEC UR_WIN_CONDITION UR_ET_DESTROYS_STATION UR_INTUITIVE	Functional, Intergrated, Unit	Bruno Davies	Pass
FireTruck_1.4	North East ('NE') direction - 315	New centre/position in relation to speed, direction and time in both the X and Y direction	A shift in the X and Y direction the correct amount to predetermined calculations - (12.511982,25.011982)	Testing that moveInDirection changes the X and Y position when no collision when Direction set to NE (270)	UR_FIRETRUCKS_UNIQUE_SPEC UR_WIN_CONDITION UR_ET_DESTROYS_STATION UR_INTUITIVE	Functional, Intergrated, Unit	Bruno Davies	Pass
FireTruck_1.5	New position where neither X not Y can move - (2903, 3211)	Same positon as Input	Same positon as Input - (2903, 3211)	Testing that movePosition does not move when another object is in the way	UR_FIRETRUCKS_UNIQUE_SPEC UR_WIN_CONDITION UR_ET_DESTROYS_STATION UR_INTUITIVE	Functional, Intergrated, Unit	Bruno Davies	Pass
FireTruck_1.6	New position where in range of Fortress or Alien - (2903, 3211)	Decrease in water level by 1f	A decrese in the water by 1f - to 99f	Testing that when a FireTruck is in range of a 'enemy' the FireTruck will attack	UR_FIRETRUCKS_REFILL UR_WIN_CONDITION UR_ET_DESTROYS_STATION UR_MINIGAME UR_HIGHSCORE UR_FORTRESS_ATTACK	Functional, Intergrated, Unit	Bruno Davies	Pass
FireTruck_1.7	N/A	Nothing to waterLevel	Nothing	Testing that when a FireTruck is not in range of an 'enemy' it does not attack	UR_FIRETRUCKS_REFILL UR_ET_DESTROYS_STATION UR_MINIGAME UR_HIGHSCORE UR_FORTRESS_ATTACK	Functional, Intergrated, Unit	Bruno Davies	Pass
FireTruck_1.8	applyDamage(10f) onto the FireTruck	FireTruck to heal the -10+2 amount	Healed from 90 to 92	Testing that replenish adds to health when not max	UR_FIRETRUCKS_UNIQUE_SPEC UR_FIRETRUCK_REPAIR UR_ET_IMPROVEMENT UR_DIFFICULTY_LEVEL UR_FORTRESS_ATTACK	Functional, Intergrated, Unit	Bruno Davies	Pass
FireTruck_1.9	New position where in range of Fortress or Alien - (2903, 3211)	Refills from not max to max water	Max water - 100f	Testing that refillWater refills water to max water	UR_FIRETRUCKS_UNIQUE_SPEC UR_FIRETRUCKS_REFILL UR_MINIGAME	Functional, Intergrated, Unit	Bruno Davies	Pass
FireTruck_2.0_USER	User playing game - Attacks alien or fortress to use water - returns to FireStation - Completes Minigame - Checks waterLevels	The user will lose water and then refill it after completing minigame	Max water in the bar after using some	Testing that user can complete a the most crusial game-life cycle to complete the game	UR_FIRETRUCKS_UNIQUE_SPEC UR_FIRETRUCKS_REFILL UR_FIRETRUCK_REPAIR UR_FIRETRUCK_MIN_START UR_ET_MIN_START UR_ET_DESTROYS_STATION UR_MINIGAME UR_HIGHSCORE UR_FORTRESS_ATTACK	Functional, Intergrated, Unit	Bruno Davies	Pass
FireTruck_2.1_USER	User playing game - Does not attack - moves in FireStation - checks waterLevels	No change in water levels and no activation of minigame	Nothing happened	Testing that there are no areas to automatically trigger minigame	UR_FIRETRUCKS_UNIQUE_SPEC UR_FIRETRUCKS_REFILL UR_FIRETRUCK_REPAIR UR_FIRETRUCK_MIN_START UR_ET_MIN_START UR_MINIGAME	Functional, Intergrated, Unit	Bruno Davies	Pass
Alien_1.1	N/A	Correct initialized variables	The correct initialized variables - via getPosition()	Testing that the Alien constructor is working as intended with standard input	UR_LOSS_CONDITION UR_DIFFICULTY_LEVEL UR_INTUITIVE UR_PATROLS	Functional, Intergrated, Unit	Bruno Davies	Pass

Alien_1.2	update() once (movementCountDown > 0)	currentWayPoint and position unchanged	No change in position and currentWayPoint	Testing that update will not move when movementCountdown is above 0	UR_LOSS_CONDITION UR_DIFFICULTY_LEVEL UR_INTUITIVE UR_PATROLS	Functional, Intergrated, Unit	Bruno Davies	Pass
Alien_1.3.	update() twice (movementCountDown == 0)	currentWayPoint and position unchanged	No change in position and currentWayPoint	Testing that update will not move when movementCountdown is 0	UR_LOSS_CONDITION UR_DIFFICULTY_LEVEL UR_INTUITIVE UR_PATROLS	Functional, Intergrated, Unit	Bruno Davies	Pass
Alien_1.4	update() thrice (movementCountDown < 0)	currentWayPoint+1 and position changed in only X directions base on calculations	currentWayPoint went from 0 to 1 and position changed in X direction	Testing the patrol moves when below 0 on the movementCountdown - X direction	UR_LOSS_CONDITION UR_DIFFICULTY_LEVEL UR_INTUITIVE UR_PATROLS	Functional, Intergrated, Unit	Bruno Davies	Pass
Alien_1.5	new alien that needs only one update() to move	currentWayPoint+1 and position changed in only Y directions base on calculations	currentWayPoint went from 0 to 1 and position changed in Y direction	Testing the patrol moves when below 0 on the movementCountdown - Y direction	UR_LOSS_CONDITION UR_DIFFICULTY_LEVEL UR_INTUITIVE UR_PATROLS	Functional, Intergrated, Unit	Bruno Davies	Pass
Alien_1.6	set currentWayPoint to last element, the alien position to that last element location and update()	currentWayPoint to reset to 0	currentWayPoint is 0	Testing that when at last wayPoint the currentWayPoint is reset to 0	UR_LOSS_CONDITION UR_DIFFICULTY_LEVEL UR_INTUITIVE UR_PATROLS	Functional, Intergrated, Unit	Bruno Davies	Pass
Alien_1.7	set currentWayPoint to last element, the alien position NOT to that last element location and update()	currentWayPoint remains at last element	currentWayPoint is the same - 5	Testing that when currentWayPoint is at the last wayPoint but the position of alien is not - do not reset	UR_LOSS_CONDITION UR_DIFFICULTY_LEVEL UR_INTUITIVE UR_PATROLS	Functional, Intergrated, Unit	Bruno Davies	Pass
Alien_1.8	USER: Controlled truck to be in path of Alien and take damage	Alien to fire at FireTruck	Bullets fired toward the truck and the user taking damage	Testing that the Aliens will stop and attack the truck visually	UR_LOSS_CONDITION UR_COLOUR_ACCESSIBILITY	Functional, Intergrated, Unit	Bruno Davies	Pass
Key for colours:								
Original Code:								
Altered Code:								