Test method	Associated requirement(s)	Class tested	Test case objective	Result
armorDamageOnBoulder	UR_OBSTACLE_ENCOUNTER FR_OBSTACLES	Player	Test if player’s armour decreases upon collision with a boulder	PASS
healthDamageOnMonster			Test if player’s health decreases upon collision with a boulder	PASS
captureTest	UR_CAPTURING_COLLEGES FR_XP_UPDATE FR_MONEY_UPDATE FR_POINTS_UPDATE		Test when the player college, the amount of plunder and xp increases	PASS
destroyTest			Test if when the player destroys a college, the amount of plunder and xp increases	PASS
outOfAmmo	UR_FIRE_WEAPONS		Test if upon ammo exhaustion the player is still capable of shooting cannonballs	PASS
isAlive	FR_ENEMY_HEALTH	Ship	Test the ship is destroyed in the event health is reduced to 0 and only 0	PASS
destroy			Test if the destroy method changes the status of the ship from alive	PASS
plunder	FR_MONEY_UPDATE FR_MONEY_TRACKING		Test the plunder method is able to permanently increment the amount of plunder of a ship by any amount between 1 and 100	PASS
setFaction	FR_COLLEGE_ENTITY_TRACKING		Ensure upon the initialisation of a new ship, constants are able to be updated by appropriate amounts with set methods.	PASS
setHealth	FR_POWERUP_EFFECTS			PASS
setArmor				PASS
setSpeed				PASS
setPlunder	FR_MONEY_UPDATE			PASS
setPlunderBonus				PASS
setXp	FR_XP_UPDATE			PASS
setXPBonus				PASS
setAmmo	FR_POWERUP_EFFECTS			PASS
setDamageDelt	FR_COLLEGE_ENTITY_TRACKING			PASS
setBulletSpeed	FR_BULLET_TRAVEL FR_BUFFS			PASS
setPosition	FR_COLLEGE_ENTITY_TRACKING			PASS
getAmmo	FR_PLAYER_AMMO		Ensure upon the initialisation of a new ship, the associated constants are set to the correct default value (Constants : Attack Range, Bullet Speed, Faction, Health, Plunder, Plunder Bonus, Position, Speed, Velocity, XP and XP bonus)	PASS
getArmor				PASS
getAttackRange	FR_COLLEGE_ENTITY_TRACKING			PASS
getBulletSpeed	FR_BULLET_TRAVEL FR_BUFFS			PASS
getFaction	FR_COLLEGE_ENTITY_TRACKING			PASS
getHealth	FR_ENEMY_HEALTH FR_PLAYER_AMMO			PASS
getPlunder	FR_MONEY_TRACKING			PASS
getPlunderBonus	FR_MONEY_TRACKING			PASS
getPosition	FR_COLLEGE_ENTITY_TRACKING			PASS
getSpeed	FR_BUFFS			PASS
getVelocity	FR_COLLEGE_ENTITY_TRACKING			PASS
getXp	FR_XP_TRACKING			PASS
getXpBonus				PASS
shootTest	FR_PLAYER_FIRE FR_BULLET_TRAVEL	CannonBall	Test if when a cannonball is created, the cannonball travels by the appropriate amount in a specified	PASS
KillCompleteTest			Test if when a new quest with the objective of destroying a college is able to be completed by a player only after achieving the objective and rewards an appropriate amount of plunder	PASS

Test method	Test case objective	Result
testAspectRatio	Ensure display constants have been assigned the correct default values in order ensuring graphics are rendered correctly	PASS
setBackgroundColour		PASS
testBuildingScale		PASS
testDimensions		PASS
testFullscreen		PASS
testHalfDimensions		PASS
testHalfDimensions		PASS
testHalfViewportHeight		PASS
testHalfViewportWidth		PASS
testOperatingSystem		PASS
testPhysicsTimeStep		PASS
testScreenHeight		PASS
testScreenWidth		PASS
testTileSize		PASS
testVSync		PASS
testViewportHeight		PASS
testViewportTitle		PASS
testViewportWidth		PASS
testZoom		PASS
testObstacleID	Ensure certain tile map cell IDs refer to the correct display items	PASS
testPassableCost		PASS
testPassableID		PASS
testAngleBetween	Ensure vector and tile calculations are computed accurately to ensure correctness of the game	PASS
testAngleToVector		PASS
testCheckProximity		PASS
testContains		PASS
testDistanceToTiles		PASS
testFloor		PASS
testRandomChoice		PASS
testRandomPos		PASS
testRound		PASS
testScale		PASS
testTilesToDistance		PASS
testVectorToAngle		PASS
testArialTTFAsset		PASS

LocateCompleteTest	FR_QUEST_TRACKING FR_QUEST_OBJECTIVE	Quest QuestManager GameManager	Test if when a new quest with the objective of locating a chest is able to be completed by a player only after achieving the objective and rewards an appropriate amount of plunder	PASS
getPosition	FR_HOSTILE_AI FR_FRIENDLY_AI FR_COLLEGE_ENTITY_TRACKING	Node	Ensure upon the initialisation of a new node, position can be assigned	PASS
set			Ensure upon the initialisation of a new node, its position is able to be updated by the set method	PASS
estimate		NodeHeuristic	Ensure the correct heuristic value (squared distance between nodes) is calculated between two different nodes	PASS
estimate		Path	Ensure path cost calculates correct values	PASS
getFromNode			Ensure the getFromNode method refers to the correct node given any path	PASS
getToNode			Ensure the getToNode method refers to the correct node given any path	PASS
vectorToAngle		AINavigation	Ensure upon the initialisation of a new AI navigation control for a steerable, the angle to a placed vector can be calculated correctly	PASS
angleToVector			Ensure upon the initialisation of a new AI navigation control for a steerable, a vector can be placed at any given angle	PASS
isTagged			Ensure upon the initialisation of a new AI navigation control for a steerable, the associated constants are set to the correct default value (Constants : isTagged, Bounding radius, Minimum linear speed, Linear speed, Linear acceleration, Maximum angular speed, Maximum angular acceleration)	PASS
getBoundingRadius				PASS
getZeroLinearSpeedThreshold				PASS
getMaxLinearSpeed				PASS
getMaxLinearAcceleration				PASS
getMaxAngularSpeed				PASS
getMaxAngularAcceleration				PASS
setTagged			Ensure upon the initialisation of a new AI navigation control for a steerable, the correct constants can be updated with appropriate values (Constants : isTagged, Linear speed, Linear acceleration, Maximum angular speed, Maximum angular acceleration)	PASS
setMaxLinearSpeed				PASS
setMaxLinearAcceleration				PASS
setMaxAngularSpeed				PASS
setMaxAngularAcceleration				PASS
addPlunder	FR_MONEY_UPDATE		Test if player stats can be correctly updated given any amount (Stats : Plunder and XP)	PASS
addXp	FR_XP_UPDATE			PASS
addTarget	FR_HOSTILE_AI FR_FRIENDLY_AI FR_COLLEGE_ENTITY_TRACKING		Test if ships can be added as targets to the pirate objects' queue for targets successfully	PASS
getAmmo	FR_PLAYER_AMMO		Ensure upon the initialisation of a pirate, associated constants are set to the correct default value (Constants : Ammunition, Armour, Faction, Health, Plunder and XP)	PASS
getArmor				PASS
getFaction	FR_COLLEGE_ENTITY_TRACKING			PASS
getHealth	FR_COLLEGE_ENTITY_TRACKING FR_ENEMY_HEALTH FR_PLAYER_AMMO			PASS
getPlunder	FR_MONEY_TRACKING			PASS
getXp	FR_XP_TRACKING			PASS
setAmmo	FR_PLAYER_AMMO			PASS

testBeachTilesetPNGAssetExists	Ensure the required external assets used by the game are located in the native files	PASS
testBeachTilesetTSXAssetExists		PASS
testBoatsPNGAssetExists		PASS
testBoatsTXTAssetExists		PASS
testBuildingsTXTAssetExists		PASS
testChestPNGAssetExists		PASS
testGameSettingsJSONAssetExists		PASS
testMapTMXAssetExists		PASS
testMenuBGJPGAssetExists		PASS
testOtherPNGAssetExists		PASS
testShipPNGAssetExists		PASS
testUISkinFolderAssetExists		PASS

setArmor	FR_POWERUP_EFFECTS	Pirate	Ensure upon the initialisation of a pirate, the correct constants can be updated with appropriate values (Constants : Ammunition, Armour, Faction, Health, Plunder and XP)	PASS				
setFactionId	FR_COLLEGE_ENTITY_TRACKING			PASS				
setHealth	FR_COLLEGE_ENTITY_TRACKING FR_ENEMY_HEALTH FR_PLAYER_AMMO			PASS				
setPlunder	FR_MONEY_UPDATE			PASS				
setXp	FR_XP_UPDATE			PASS				
canAttack	FR_HOSTILE_AI FR_PLAYER_FIRE		A newly spawned isolated pirate should not be able to attack	PASS				
isAgro	FR_HOSTILE_AI		Ensure pirates are unable to attack outside their attack range	PASS				
isAlive	FR_ENEMY_HEALTH FR_PLAYER_AMMO		Ensure newly spawned pirates have full health	PASS				
kill	FR_ENEMY_HEALTH FR_PLAYER_AMMO		Test the kill method causes the pirate to no longer alive	PASS				
reload	FR_PLAYER_AMMO		Test the reload method is able to increment the current amount of ammunition to a greater amount	PASS				
takeDamage	FR_ENEMY_HEALTH		Test the takeDamage method causes a pirates health to decrease by the corresponding amount, where the maximum amount of reduction is capped at the ships current health	PASS				
getFontColour	NFR_COLOURBLINDNESS	Text	Ensure upon the initialisation of new text, associated constants are set to the correct default value (Constants : Font Colour and Position)	PASS				
getPosition				PASS				
setFontColour			Ensure upon the initialisation of new text, the correct constants can be updated with appropriate values (Constants : Font Colour and Position)	PASS				
setPosition				PASS				
angleToVector		Transform	Ensure upon the initialisation of a new transform component the angle to a placed vector can be calculated correctly	PASS				
vectorToAngle			Ensure upon the initialisation of a new transform component, a vector can be placed at any given angle	PASS				
getOrientation			Ensure attributes associated with a transform component are set to the correct default values (Attributes : Orientation, Position, Rotation and Scale)	PASS				
getPosition				PASS				
getRotation				PASS				
getScale				PASS				
setPosition			Ensure attributes associated with a transform component can be updated with appropriate values; displaying consistent and accurate change (Attributes : Position, Rotation and Scale	PASS				
setRotation				PASS				
setScale				PASS				

getMaxTime	UR_SCALING	CaptionManager GameManager	Ensure upon the initialisation of a new CaptionManager the maximum time until a caption is removed prior to being placed is set to the correct default value	PASS
getdisplay			Ensure upon the initialisation of a new CaptionManager the current text displayed is empty	PASS
setMaxTime			Ensure upon the initialisation of a new CaptionManager the maximum time until a caption is removed prior to being placed can be updated with appropriate values	PASS
setDisplay			Ensure upon the initialisation of a new CaptionManager the current text being displayed can be updated with appropriate strings	PASS
easyPlayerHandler	FR_DIFFICULTY FR_CHANGE_DIFFICULTY	DifficultyManager GameManager	Test if the chosen difficulty can be initialised by DifficultyManager to modify the constants of the game using GameManager (Constants : health, armour, starting plunder, speed, bullet speed and ammo)	PASS
mediumPlayerHandler				PASS
hardPlayerHandler				PASS
easyCaptureHandler	UR_CAPTURING_COLLEGES UR_HOSTILE_COLLEGE_CAPTURE FR_DIFFICULTY FR_CHANGE_DIFFICULTY FR_XP_UPDATE FR_MONEY_UPDATE	DifficultyManager CaptureManager	Test if the chosen difficulty can be initialised by DifficultyManager to modify the amount of XP and plunder a player receives upon capturing a college	PASS
mediumCaptureHandler				PASS
hardCaptureHandler				PASS
easyDestroyHandler			Test if the chosen difficulty can be initialised by DifficultyManager to modify the amount of XP and plunder a player receives upon capturing a college	PASS
mediumDestroyHandler				PASS
hardDestroyHandler				PASS
easyEnhancementHandler	UR_POWERUPS FR_CHANGE_DIFFICULTY	DifficultyManager EnhancementManager	(Constants : health increase, speed increase, armour increase, ammo increase, immunity time and infinite ammo time)	PASS
mediumEnhancementHandler				PASS
hardEnhancementHandler				PASS
easyEnhancementCostHandler			Test if the chosen difficulty can be initialised by DifficultyManager to modify the cost power up constants of the game using EnhancementManager	PASS
mediumEnhancementCostHandler				PASS
hardEnhancementCostHandler				PASS
saveColleges	FR_SERIALISATION FR_SAVEGAME	SaveManager GameManager	Ensure that when the game state can be saved by the save manger correctly, storing locations of all entities and current player stats into saved_preference	PASS
savePlayerStats				PASS
saveShips				PASS
loadColleges	FR_SERIALISATION FR_LOAD		Ensure that saved components can be applied into the newly initialised game correctly	PASS
loadPlayerStats				PASS
loadShips				PASS
saveDifficulty	FR_SERIALISATION FR_SAVEGAME	SaveManager GameManager DifficultyManager	Ensure the current difficulty can be retrieved from DifficultyManager and saved into saved_preference	PASS
loadDifficulty	FR_SERIALISATION FR_LOAD		Ensure that saved difficulty can be applied into the newly initialised game correctly	PASS
getAmmo			Ensure enhancement related attributes can be	PASS
getArmor				PASS
getHealth				PASS

getImmunity	FR_BUFFS FR_SHOP		retrieved by EnhancementManger in order to be modified Correct captions should be displayed given only certain powerups can be purchased at the start of the game (Attributes :Ammunition, Armour, Health, Invincibility Time, Infinite Ammunition time, Speed,Costs of all Power-ups and Current assurance player can purchase power up)	PASS
getInfiniteAmmo				PASS
getSpeed				PASS
getTax				PASS
getValidation				PASS
setAmmo			Ensure a enhancement related attributes can be updated with appropriate values; displaying consistent and accurate change Correct captions should be displayed upon update events (Attributes :Ammunition, Armour, Health, Invincibility Time, Infinite Ammunition time, Speed and Costs of all Power-ups)	PASS
setArmor				PASS
setHealth				PASS
setImmunity				PASS
setInfiniteAmmo				PASS
setSpeed				PASS
setTax				PASS
ammoHandlerFree	FR_POWERUPS FR_BUFFS FR_POWERUP_EFFECTS FR_SHOP FR_PLAYER_AMMO		Ensure when a player picks up a power-up, the effect of the power-up is successful and instant Captions should be displayed to notify player of pickup (Power-ups : Ammunition, Armour, Health, Immunity, Infinite Ammunition and Speed)	PASS
armorHandlerFree				PASS
healthHandlerFree				PASS
immunityHandlerFree				PASS
infiniteAmmoHandlerFree				PASS
speedHandlerFree				PASS
ammoHandlerPaid	FR_PURCHASE FR_BUFFS FR_POWERUP_EFFECTS FR_SHOP FR_PLAYER_AMMO FR_MONEY_TRACKING		Ensure upon the purchase of a powerup from a substantial amount of plunder, the effect is instant and the amount of player plunder is updated accordingly Captions should be displayed to verify the purchase (Power-ups : Ammunition, Armour, Health, Immunity, Infinite Ammunition and Speed)	PASS
armorHandlerPaid				PASS
healthHandlerPaid				PASS
immunityHandlerPaid				PASS
infiniteAmmoHandlerPaid				PASS
speedHandlerPaid				PASS
taxation	FR_PURCHASE FR_MONEY_TRACKING FR_SHOP	EnhancementManager GameManager CaptionManager	Ensure accurate shop transactions occur between the shop and player	PASS