Unit Tests

These tests are for ensuring the back end of the game acts as it should. They have been split up into the relevant classes they are from and will be testing (e.g. GameTest class will be testing the Game class) Results are below:

Game class tests (GameTest)

Test ID	Test name	Requirement	Expected outcome	Result
G1	CreatePlayers_FourPlayersAreHuman	N3, F1	4 players are human	Pass
G2	CreatePlayers_ThreePlayersAreHumanAndOneNot	N3, F1	3 players human, one not	Pass
G3	EndGame_GameEndsCorrectlyWithNoCurrentPlayerAndNoActivePlayersAndNoTurnState		Game ends with end game variables in place	Pass
G4	GetWinner_NoWinnerWhenAPlayerHasLandmarkAndAnotherHasUnits		No winner found when only one player has landmarks but another still has units	Pass
G5	GetWinner_NoWinnerWhenMultiplePlayersOwningLandmark		No winner found if multiple players have landmarks	Pass
G6	GetWinner_NoWinnerWhenMultiplePlayersWithUnits		No winner found if multiple players have units	Pass
G7	GetWinner_OnePlayerWithLandmarksWins		Winner found if only one player has landmarks	Pass
G8	GetWinner_OnePlayerWithUnitsWins		Winner found if only one player has units	Pass
G9	InitializeMap_OneLandmarkAllocatedWithUnitPerPlayer	N6	Each player has one landmark with a unit on it	Pass
G10	NeutralPlayerTurn_EnsureNeutralPlayerMovesCorrectly	F1	Neutral player moves into sectors with no units and not the vice chancellor	Pass
G11	NextPlayer_CurrentPlayerChangesToNextPlayerEachTime		Turn changes to next player	Pass
G12	NextPlayer_EliminatedPlayersAreSkipped	N3	Players that are not active do not become current player	Pass
G13	NextPlayer_PlayerNullifyResourcesCountersAreDecremented	F8, N14	Resource nullified counter decreases after each player	Pass

G14	NextPlayer_PlayersWithASkipFlagAreSkipped	F8, N14	Player who should be skipped (with skip flag) is skipped	Pass
G15	NextPlayer_UnitFrozenCountersAreDecremented	F8, N14	Frozen counter for a unit decreases each turn	Pass
G16	NoUnitSelected_ReturnsFalseWhenUnitIsSelected		Detects when unit is not selected correctly, false when unit selected	Pass

Player class tests (PlayerTest)

<u>**Test ID**</u>	<u>**Test name**</u>	<u>**Requirement**</u>	<u>**Expected outcome**</u>	<u>**Result**</u>
P1	CaptureLandmark_BothPlayersBeerAmountCorrect	N6, F5	New owners beer amount increases correctly, old owner beer amount decreases correctly (landmark amount)	Pass
P2	CaptureLandmark_BothPlayersKnowledgeAmountCorrect	N6, F5	New owners knowledge amount increases correctly, old owner knowledge amount decreases correctly (landmark amount)	Pass
P3	CaptureLandmak_NeutralLandmarkPlayerBeerAmountCorrect	N6, F5	New owner gains the correct amount of beer (landmark amount)	Pass
P4	CaptureLandmark_NeutralLandmarkPlayerKnowledgeAmountCorrect	N6, F5	New owner gains the correct amount of knowledge (landmark amount)	Pass
P5	CaptureSector_ChangesOwner	N9, F5	Sector changes its owner (from null or previous owner) and sector added to players sectors	Pass
P6	IsEliminated_PlayerWithNoUnitsAndNoLandmarksEliminated		Player is eliminated when they have no units and no landmarks	Pass
P7	SpawnUnits_SpawnedWhenLandmarkOwned	F6	Units only spawn when player has a landmark	Pass
P8	SpawnUnits_NotSpawnedWhenLandmarkOwnedAndOccupied	F6	A unit does not spawn on an owned landmark when it is occupied	Pass
P9	SpawnUnits_SpawnedWhenLandmarkOwnedAndUnoccupied	F6	A unit does spawn in an owned landmark that is not occupied	Pass

SaveGame class tests (SaveGameTest)

<u>**Test ID**</u>	<u>**Test name**</u>	<u>**Requirement**</u>	<u>**Expected outcome**</u>	<u>**Result**</u>
SG1	Load_NullFileName	F7	No game is loaded	Pass

SG2	SaveLoad_Game_NeutralPlayer	F7	Before and after load, the game has a neutral player	Pass
SG3	SaveLoad_Game_TurnState	F7	Before and after load, the current player and the number of remaining actions is are the same	Pass
SG4	SaveLoad_Player_NullifiedResources	F7	Before and after load, the Nullify Resources effect are the same	Pass
SG5	SaveLoad_Player_PunishmentCards	F7	Checks if the number and type of punishment cards owned by each player is saved and loaded correctly	Pass
SG6	SaveLoad_Player_ResourceBonus	F7	Checks if the minigame resource bonus is saved and loaded correctly	Pass
SG7	SaveLoad_Player_Skip	F7	Checks if the Skip Turn effect is saved and loaded correctly	Pass
SG8	SaveLoad_Sector_Owner	F7	Checks if sector ownership is saved and loaded correctly	Pass
SG9	SaveLoad_Sector_PunishmentCard	F7	Checks if the placement and type of punishment cards are saved and loaded correctly	Pass
SG10	SaveLoad_Sector_VC	F7	Checks if the location of the VC is saved and loaded correctly	Pass
SG11	SaveLoad_Unit_Frozen	F7	Checks if the Freeze Unit effect is saved and loaded correctly	Pass
SG12	SaveLoad_Unit_Level	F7	Checks if unit level is saved and loaded correctly	Pass
SG13	SaveLoad_Unit_Name	F7	Checks if unit names are saved and loaded correctly	Pass
SG14	SaveLoad_Unit_Sector	F7	Checks if unit location is saved and loaded correctly	Pass

Sector class tests (SectorTest)

Test ID	**Test name**	**Requirement**	**Expected outcome**	**Result**
S1	AdjacentSelectedUnit_SectorsAreAdjacent	F5	Adjacent sectors in the game are adjacent	Pass
S2	ClearUnit_UnitRemovedFromSector		Unit is removed from sector and kept null if already empty	Pass

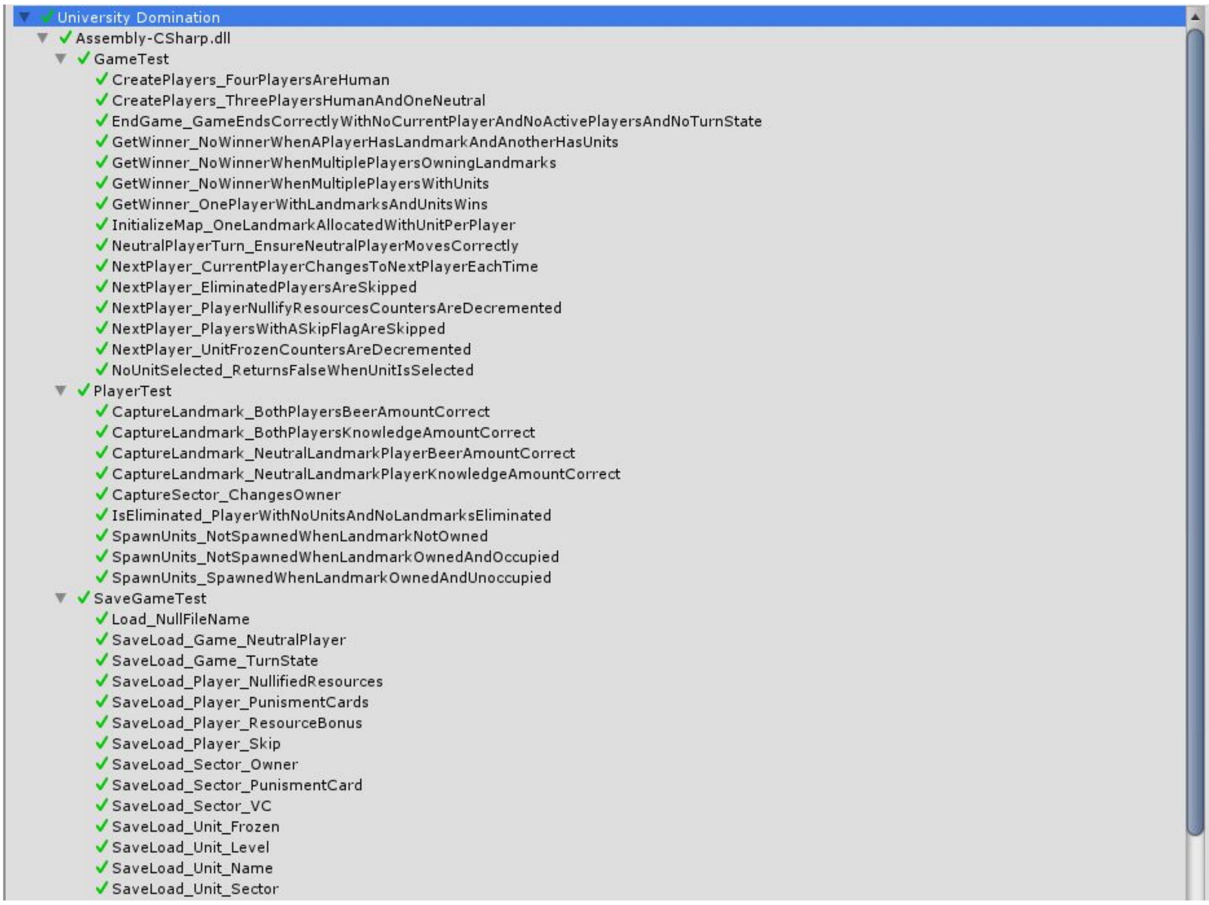
S3	Initialize_OwnedAndNotOwnedSectorsOwnedAndColor	N8	Landmarks and normal sector are initialized with the colour grey	Pass
S4	MoveIntoFriendlyUnit_UnitsSwapSectorsAndTurnStateProgressed	F5	Two units swap places and turn state progresses to move 2	Pass
S5	MoveIntoHostileUnit_AttackingUnitTakesSectorAndLevelUpAndTurnEnd	N9, F5	Attacking unit moves from its sector to the new sector, gains a level and the turn state progresses to end of turn	Pass
S6	MoveIntoHostileUnit_DefendingUnitDefendsSectorAndTurnEnd	N9, F5	Attacking unit is destroyed, defending unit does not level up and turn state progresses to end of turn	Pass
S7	MoveIntoHostileUnit_TieConflict_DefendingUnitDefendsSectorAndTurnEnd	N9, F5	Attacking unit is destroyed, defending unit does not level up and turn state progresses to end of turn	Pass
S8	MoveIntoUnoccupiedSector_NewSectorHasUnitAndOldDoesNotAndTurnStateProgressed	F5	Unit moves out of one sector into the other sector and turn state progresses to move 2	Pass
S9	OnMouseAsButton_CorrectUnitIsSelected	N8	Unit clicked on is selected only when the clicking the current players units and there is not already a unit selected	Pass
S10	SetOwner_SectorOwnerAndColorCorrect	N8	Colour of the owned sector matches the colour of the player	Pass

Unit class tests (UnitTest)

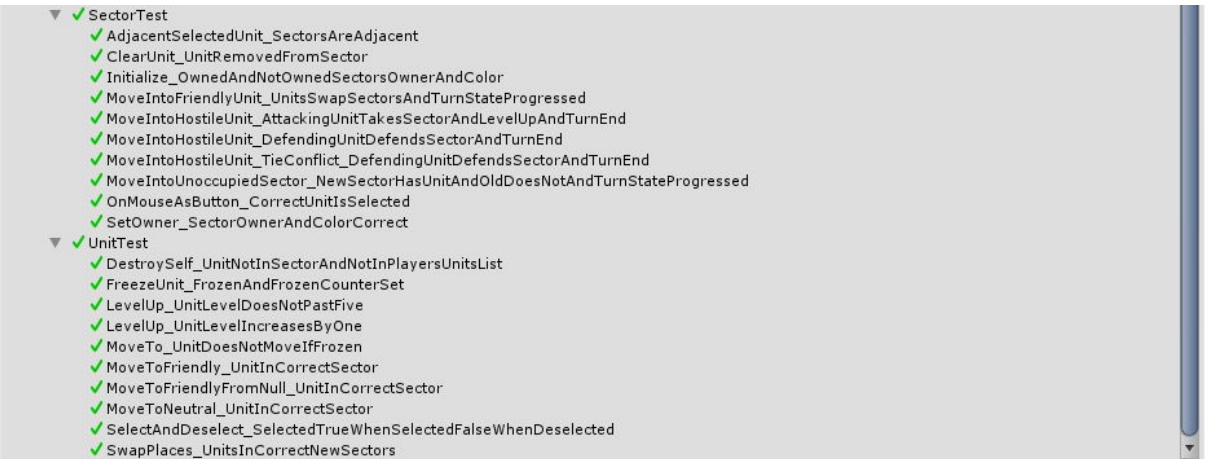
Test ID	**Test name**	**Requirement**	**Expected outcome**	**Result**
U1	DestroySelf_UnitNotInSectorAndNotInPlayersUnitsList		Unit no longer in sector and not in unit list of player	Pass
U2	FreezeUnit_FrozenAndFrozenCounterSet	F8, N14	Unit if frozen when set and counter is correct	Pass
U3	LevelUp_UnitLevelDoesNotPastFive	N13	Level does not increase past 5 after trying to level up again	Pass
U4	LevelUp_UnitLevelIncreasesByOne	N13	Level increases by one	Pass
U5	MoveTo_UnitDoesNotMoveIfFrozen	F8, N14	Unit does not move into sector when frozen	Pass
U6	MoveToFriendly_UnitInCorrectSector	F5	Unit moves to sector, unit is in new sector and no longer in old	Pass

			sector	
U7	MoveToFriendlyFromNull_UnitInCorrectSector	F5	Unit moves to sector, unit is in sector	Pass
U8	MoveToNeutral_UnitInCorrectSector	F5	Unit moves to sector, unit is in new sector and no longer in old sector	Pass
U9	SelectAndDeselect_SelectedTrueWhenSelectedFalseWhenDeselected		Unit is selected when it is meant to be and deselected when deselected	Pass
U10	SwapPlaces_UnitsInCorrectNewSectors	F5	Units move from their old sector to the new one, swapping places	Pass

Evidence



```
▼ ✓ University Domination
  ▼ ✓ Assembly-CSharp.dll
    ▼ ✓ GameTest
        ✓ CreatePlayers_FourPlayersAreHuman
        ✓ CreatePlayers_ThreePlayersHumanAndOneNeutral
        ✓ EndGame_GameEndsCorrectlyWithNoCurrentPlayerAndNoActivePlayersAndNoTurnState
        ✓ GetWinner_NoWinnerWhenAPlayerHasLandmarkAndAnotherHasUnits
        ✓ GetWinner_NoWinnerWhenMultiplePlayersOwningLandmarks
        ✓ GetWinner_NoWinnerWhenMultiplePlayersWithUnits
        ✓ GetWinner_OnePlayerWithLandmarksAndUnitsWins
        ✓ InitializeMap_OneLandmarkAllocatedWithUnitPerPlayer
        ✓ NeutralPlayerTurn_EnsureNeutralPlayerMovesCorrectly
        ✓ NextPlayer_CurrentPlayerChangesToNextPlayerEachTime
        ✓ NextPlayer_EliminatedPlayersAreSkipped
        ✓ NextPlayer_PlayerNullifyResourcesCountersAreDecremented
        ✓ NextPlayer_PlayersWithASkipFlagAreSkipped
        ✓ NextPlayer_UnitFrozenCountersAreDecremented
        ✓ NoUnitSelected_ReturnsFalseWhenUnitIsSelected
    ▼ ✓ PlayerTest
        ✓ CaptureLandmark_BothPlayersBeerAmountCorrect
        ✓ CaptureLandmark_BothPlayersKnowledgeAmountCorrect
        ✓ CaptureLandmark_NeutralLandmarkPlayerBeerAmountCorrect
        ✓ CaptureLandmark_NeutralLandmarkPlayerKnowledgeAmountCorrect
        ✓ CaptureSector_ChangesOwner
        ✓ IsEliminated_PlayerWithNoUnitsAndNoLandmarksEliminated
        ✓ SpawnUnits_NotSpawnedWhenLandmarkNotOwned
        ✓ SpawnUnits_NotSpawnedWhenLandmarkOwnedAndOccupied
        ✓ SpawnUnits_SpawnedWhenLandmarkOwnedAndUnoccupied
    ▼ ✓ SaveGameTest
        ✓ Load_NullFileName
        ✓ SaveLoad_Game_NeutralPlayer
        ✓ SaveLoad_Game_TurnState
        ✓ SaveLoad_Player_NullifiedResources
        ✓ SaveLoad_Player_PunismentCards
        ✓ SaveLoad_Player_ResourceBonus
        ✓ SaveLoad_Player_Skip
        ✓ SaveLoad_Sector_Owner
        ✓ SaveLoad_Sector_PunismentCard
        ✓ SaveLoad_Sector_VC
        ✓ SaveLoad_Unit_Frozen
        ✓ SaveLoad_Unit_Level
        ✓ SaveLoad_Unit_Name
        ✓ SaveLoad_Unit_Sector
```



```
    ▼ ✓ SectorTest
        ✓ AdjacentSelectedUnit_SectorsAreAdjacent
        ✓ ClearUnit_UnitRemovedFromSector
        ✓ Initialize_OwnedAndNotOwnedSectorsOwnerAndColor
        ✓ MoveIntoFriendlyUnit_UnitsSwapSectorsAndTurnStateProgressed
        ✓ MoveIntoHostileUnit_AttackingUnitTakesSectorAndLevelUpAndTurnEnd
        ✓ MoveIntoHostileUnit_DefendingUnitDefendsSectorAndTurnEnd
        ✓ MoveIntoHostileUnit_TieConflict_DefendingUnitDefendsSectorAndTurnEnd
        ✓ MoveIntoUnoccupiedSector_NewSectorHasUnitAndOldDoesNotAndTurnStateProgressed
        ✓ OnMouseAsButton_CorrectUnitIsSelected
        ✓ SetOwner_SectorOwnerAndColorCorrect
    ▼ ✓ UnitTest
        ✓ DestroySelf_UnitNotInSectorAndNotInPlayersUnitsList
        ✓ FreezeUnit_FrozenAndFrozenCounterSet
        ✓ LevelUp_UnitLevelDoesNotPastFive
        ✓ LevelUp_UnitLevelIncreasesByOne
        ✓ MoveTo_UnitDoesNotMoveIfFrozen
        ✓ MoveToFriendly_UnitInCorrectSector
        ✓ MoveToFriendlyFromNull_UnitInCorrectSector
        ✓ MoveToNeutral_UnitInCorrectSector
        ✓ SelectAndDeselect_SelectedTrueWhenSelectedFalseWhenDeselected
        ✓ SwapPlaces_UnitsInCorrectNewSectors
```

To view and run the actual tests, please find them in the Unity files under “Assets, Unit Tests”, using Unity’s test runner in ‘PlayMode’ to run them in the Unity editor.