

Unit Testing Updated Table

Game class tests (GameTest)

| <u>Test ID</u> | <u>Test name</u> | <u>Requirement</u> | <u>Expected outcome</u> | <u>Result</u> |
|----------------|--|--------------------|---|---------------|
| G1 | CreatePlayers_FourPlayers AreHuman | N3, F1 | 4 players are human | Pass |
| G2 | CreatePlayers_ThreePlayer sHumanAndOneNeutral() | N3, F1 | 3 players are human, 1 is neutral | Pass |
| G3 | NeutralPlayerTurn_EnsureN eutralPlayerMovesCorrectly() | F1 | Neutral player only moves to correct sectors | Pass |
| G4 | EndGame_GameEndsCorr ectlyWithNoCurrentPlayerA ndNoActivePlayersAndNoT urnState | | Game ends with end game variables in place | Pass |
| G5 | GetWinner_NoWinnerWhen APlayerHasLandmarkAndA notherHasUnits | | No winner found when only one player has landmarks but another still has units | Pass |
| G6 | GetWinner_NoWinnerWhen MultiplePlayersOwningLand mark | | No winner found if multiple players have landmarks | Pass |
| G7 | GetWinner_NoWinnerWhen MultiplePlayersWithUnits | | No winner found if multiple players have units | Pass |
| G8 | GetWinner_OnePlayerWith LandmarksAndUnitsWins | | Winner found if only one player has landmarks | Pass |
| G9 | InitializeMap_OneLandmark AllocatedWithUnitPerPlayer | N6 | Each player has one landmark with a unit on it | Pass |
| G10 | NextPlayer_CurrentPlayerC hangesToNextPlayerEachTi me | | Turn changes to next player | Pass |
| G11 | NextPlayer_EliminatedPlaye rsAreSkipped | N3 | Players that are not active do not become current player | Pass |

| | | | | |
|-----|---|----|---|------|
| G12 | NextTurnState_TurnStateProgressesCorrectly | N3 | Turn state progresses through move 1, move 2 and end of turn. | Pass |
| G13 | 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 | CaptureLandmark_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_NotSpawnedWhenLandmarkNotOwned | F6 | Unit will not spawn on a landmark that is not owned | Pass |

| | | | | |
|----|---|----|--|------|
| P8 | SpawnUnits_NotSpawnedWhenLandmarkOwnedAndOccupied | F6 | Unit will not spawn on an owned and occupied landmark | Pass |
| P9 | SpawnUnits_SpawnedWhenLandmarkOwnedAndUnoccupied | F6 | Units only spawn when player has a landmark and landmark is unoccupied | Pass |

Sector class tests (SectorTest)

| <u>Test ID</u> | <u>Test name</u> | <u>Requirement</u> | <u>Expected outcome</u> | <u>Result</u> |
|----------------|--|--------------------|--|---------------|
| S1 | AdjacentSelectedUnit_SectorsAreAdjacent | F5 | | Pass |
| S2 | ClearUnit_UnitRemovedFromSector | | Unit is removed from sector and kept null if already empty | Pass |
| S3 | Highlight_SectorColourCorrect | N8 | Sector colour becomes brighter when highlighted and return to its original colour once the highlight is reverted. | Pass |
| S4 | Initialize_OwnedAndNotOwnedSectorsOwnedAndColor | N8 | Landmarks and normal sector are initialized with the colour grey | Pass |
| S5 | MoveIntoFriendlyUnit_UnitsSwapSectorsAndTurnStateProgressed | F5 | Two units swap places and turn state progresses to move 2 | Pass |
| S6 | 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 |

| | | | | |
|-----|--|--------|---|------|
| S7 | MoveIntoHostileUnit_DefendingUnitDefendsSectorAndTurnEnd | N9, F5 | Attacking unit is destroyed, defending unit does not level up and turn state progresses to end of turn | Pass |
| S8 | MoveIntoHostileUnit_TieConflict_DefendingUnitDefendsSectorAndTurnEnd | N9, F5 | Attacking unit is destroyed, defending unit does not level up and turn state progresses to end of turn | Pass |
| S9 | MoveIntoUnoccupiedSector_NewSectorHasUnitAndOldDoesNotAndTurnStateProgressed | F5 | Unit moves out of one sector into the other sector and turn state progresses to move 2 | Pass |
| S10 | OnMouseAsButton_CorrectUnitsSelected | N8 | Unit clicked on is selected only when the clicking the current players units and there is not already a unit selected | Pass |
| S11 | SetOwner_SectorOwnerAndColorCorrect | N8 | Colour of the owned sector matches the colour of the player | Pass |

Unit class tests (UnitTest)

| <u>Test ID</u> | <u>Test name</u> | <u>Requirement</u> | <u>Expected outcome</u> | <u>Result</u> |
|----------------|---|--------------------|---|---------------|
| U1 | DestroySelf_UnitNotInSectorAndNotInPlayersUnitsList | | Unit no longer in sector and not in unit list of player | Pass |
| U2 | LevelUp_UnitLevelDoesNotPastFive | | Level does not increase past 5 after trying to level up again | Pass |
| U3 | LevelUp_UnitLevelIncreasesByOne | | Level increases by one | Pass |

| | | | | |
|----|---|----|---|------|
| U4 | MoveToFriendly_UnitInCorrectSector | F5 | Unit moves to sector, unit is in new sector and no longer in old sector | Pass |
| U5 | MoveToFriendlyFromNull_UnitInCorrectSector | F5 | Unit moves to sector, unit is in sector | Pass |
| U6 | MoveToNeutral_UnitInCorrectSector | F5 | Unit moves to sector, unit is in new sector and no longer in old sector | Pass |
| U7 | MoveToHostile_UnitInCorrectSectorAndLevelUp | F5 | Unit moves to sector, levels up and is no longer in old sector | Pass |
| U8 | SelectAndDeselect_SelectedTrueWhenSelectedFalseWhenDeselected | | Unit is selected when it is meant to be and deselected when deselected | Pass |
| U9 | SwapPlaces_UnitsInCorrectNewSectors | F5 | Units move from their old sector to the new one, swapping places | Pass |

Save Game Tests (SaveGameTest)

| <u>Test ID</u> | <u>Test name</u> | <u>Requirement</u> | <u>Expected outcome</u> | <u>Result</u> |
|----------------|------------------|--------------------|--|---------------|
| SG1 | SaveLoadGame() | F7 | The game can save and load a game successfully | Pass |
| SG2 | Load() | | The game should return null when attempting to load a file which doesn't exist | Pass |