

Unit Tests

The tests highlighted in **Blue** are new tests that have been added during Assessment 4.

Game class tests (GameTest):

| Test ID | Test Name | Related Requirements | Expected Outcome | Result |
|---------|--|----------------------|--|--------|
| G1 | CreatePlayers_AtLeastTwo PlayersAreHuman | N3, F1 | 2 players are human, 2 are not. | Pass |
| G2 | CreatePlayers_AtMostFour PlayersAreHuman | N3, F1 | 4 players are human. | Pass |
| G3 | CreatePlayers_FourPlayers AreHuman | N3, F1 | 4 players are human. | Pass |
| G4 | CreatePlayers_ThreePlayersAreHumanAndOneNot | N3, F1 | 3 players human, one not. | Pass |
| G5 | CreatePlayers_TwoPlayersAreHumanAndTwoNot | N3, F1 | 2 players human 2 are not. | Pass |
| G6 | EndGame_GameEndsCorrectlyWithNoCurrentPlayerAndNoActivePlayersAndNoTurnState | | Game ends with end game variables in place. | Pass |
| G7 | GetWinner_NoWinnerWhenAPlayerHasLandmarkAndAnotherHasUnits | | No winner found when only one player has landmarks but another still has units | Pass |
| G8 | GetWinner_NoWinnerWhenMultiplePlayersOwnLandmarks | | No winner found if multiple players have landmarks | Pass |
| G9 | GetWinner_NoWinnerWhenMultiplePlayersWithUnits | | No winner found if multiple players have units | Pass |
| G10 | GetWinner_OnePlayerWithLandmarksAndUnitWins | | Winner found if only one player owns a landmark and a unit. | Pass |
| G11 | InitializeMap_OneLandmarkAllocatedWithUnitPerPlayer | N6 | Each player has one landmark with a unit on it. | Pass |
| G12 | NextPlayer_CurrentPlayerChangesToNextPlayerEachTime | | Turn changes to next player | Pass |
| G13 | NextPlayer_EliminatedPlayersAreSkipped | N3 | Players that are not active do not become current player. | Pass |
| G14 | NextTurnState_TurnStateProgressesCorrectly | N3 | Turn state progresses through Move1, Move2 and EndOfTurn. | Pass |

| | | | | |
|-----|--|----|--|------|
| G15 | NoUnitSelected_ReturnsFalseWhenUnitIsSelected | F5 | Detects when unit is not selected correctly, false when unit selected. | Pass |
| G16 | SpawnVice_AddsAViceChancellorLandmarkToARandomSector | F2 | When called, the number of sectors containing a PVC landmark becomes equal to 1. | Pass |
| G17 | SpawnVice_AddsAViceChancellorLandmarkToASpecificSector | F2 | The sector corresponding to the ID provided contains a PVC landmark. | Pass |

Player class tests (PlayerTest):

| Test ID | Test Name | Related Requirements | Expected Outcome | Result |
|---------|---|----------------------|--|--------|
| 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_NotSpawnedWhenLandmarkOwnedAndOccupied | F6 | If a landmark sector owned by a player contains a unit, spawnUnits() does not place another unit in the landmark sector. | Pass |
| P8 | SpawnUnits_NotSpawnedWhenLandmarkNotOwned | F6 | If a landmark sector is not owned by playerA, playerA.spawnUnits() does not place a unit in the landmark sector. (landmarkedSector.GetUnit is null.) | Pass |

| | | | | |
|-----|--|------------|--|------|
| P9 | SpawnUnits_SpawnedWhenLandmarkOwnedAndUnoccupied | F6 | If a landmark is owned by playerA and doesn't contain a unit, playerA.SpawnUnits() will spawn a unit into the landmark sector. (landmarkSector.GetUnit() is in playerA.units). | Pass |
| P10 | ComputerTurn_MakesAValidMove | N3, F1, F5 | When ComputerTurn is called, the player captures another sector, it's unit has moved position and the sector moved to is adjacent to the starting sector. | Pass |
| P11 | ComputerTurn_InvalidTurnStateNoMovelsMade | N3, F1 | The number of owned sectors does not change and the unit owned by the player does not move sectors. | Pass |
| P12 | ComputerPlayPunishmentCard_CardPlayed | N3, F1 | The players card is added to cardDeck's activeCards list, the card is removed from the player's ownership and the turn ends. | Pass |
| P13 | ComputerPlayPunishmentCard_MakesNormalMoveWhenNoCardsOwned | N3, F1 | The player captures another sector, it's unit has moved position and the sector moved to is adjacent to the starting sector. | Pass |
| P14 | ComputerPlayPunishmentCard_MakesNormalMoveWhenInvalidTurnState | N3, F1 | The player captures another sector, it's unit moves sectors, the sector moved to is adjacent to the starting sector and the test card isn't activated. | Pass |
| P15 | ComputerPlayPunishmentCard_MakesNormalMoveWhenAllCardsOwnedAreActive | N3, F1 | None of the computer player's cards are activated, the player captures a new sector, it's unit moves sectors and the sector moved to is adjacent to the starting sector. | Pass |
| P16 | GetPunishmentCards_NewPlayerReturnsEmptyCardList | F12 | The player's punishment card list is empty when initialized. | Pass |
| P17 | AddPunishmentCard_CorrectlyAddsCardToList | F13 | When AddPunishmentCard(testCard) is called for a new player, it's punishmentCards list should be a list only | Pass |

| | | | | |
|-----|--|-----|---|------|
| | | | containing testCard. | |
| P18 | AddPunishmentCard_DoesNotAddCardIfListIs Full | F13 | If the player already owns 5 cards, addPunishment should not alter the players punishmentCards list. | Pass |
| P19 | AddPunishmentCard_DoesNotAddCardIfCardIsNotOwnedByPlayer | F13 | When addPunishmentCard(testCard) is called, if the testCard is not owned by the player, their punishmentCards list should not change. | Pass |

Sector class tests (SectorTest)

| Test ID | Test Name | Related Requirements | Expected Outcome | Result |
|---------|--|----------------------|---|--------|
| S1 | AdjacentSelectedUnit_SectorsAreAdjacent | F5 | If sectorA contains a unit which is unselected, sectorB.AdjacentSelectedUnit returns null. If sectorA contains a unit which is selected, sectorB.AdjacentSelectedUnit is not null. | 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 returns to its original colour once the highlight is reverted. | Pass |
| S4 | Initialize_OwnedAndNotOwnedSectorsOwnedAndColor | N8 | Landmarks and normal sectors 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 TurnState progresses to Move2. | Pass |
| S10 | OnMouseAsButton_CorrectUnitsIsSelected | 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):

| Test ID | Test Name | Related Requirements | Expected Outcome | Result |
|---------|---|----------------------|---|--------|
| U1 | DestroySelf_UnitNotInSectorAndNotInPlayersUnitsList | | Unit no longer in sector and not in unit list of player. | Pass |
| U2 | LevelUp_UnitLevelDoesNotPastFive | N9 | The Unit's level does not increase past 5 after trying to level up again. | Pass |
| U3 | LevelUp_UnitLevelIncreasesByOne | N9 | The Unit's level increases by one. | Pass |
| U4 | MoveToFriendly_UnitInCorrectSector | F5 | The Unit moves to sector; it is now in the new sector and no longer in the old one. | Pass |
| U5 | MoveToFriendlyFromNull_UnitInCorrectSector | F5 | The 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, | Pass |

| | | | | |
|-----|--|-----|--|------|
| | | | swapping places. | |
| U10 | OnMouseUpAsButton_PostgradMovesToAdjacentSector | F10 | When a postgrad is selected and an adjacent sector is clicked, the adjacent sector's unit attribute will be the postgrad unit, the owner of the sector will be equal to the owner of the unit and the unit's original sector will not contain any units. | Pass |
| U11 | OnMouseUpAsButton_PostgradMovesToPlayerOwnedSector | F10 | When a postgrad is selected and a player-owned sector is clicked, the clicked sector's unit attribute will be the postgrad unit, the owner of the sector will stay the same and the unit's original sector will not contain any units. | Pass |
| U12 | OnMouseUpAsButton_PostgradCannotMoveToNonAdjacentEnemySector | F10 | When a postgrad is selected and a non-adjacent enemy sector is clicked, the postgrad will not move to that sector. The enemy sector's unit attribute is null and it's owner stays the same. | Pass |

viceScript class tests (ViceScriptTest):

| Test ID | Test Name | Related Requirements | Expected Outcome | Result |
|---------|--|----------------------|--|--------|
| V1 | LegsLeftPressed_DecrementsLegChoiceOrLoopsToEnd | F3 | When LegsLeftPressed is called, if the legChoice is zero, then the legChoice becomes equal to the size of the legs list -1. Otherwise, legChoice decreases by 1. | Pass |
| V2 | LegsRightPressed_IncrementsLegChoiceOrLoopsToStart | F3 | When LegsRightPressed is called, if the legChoice is equal to the length of the legs list -1, then legChoice becomes 0. Otherwise, legChoice increases its value by 1. | Pass |
| V3 | MiniGameScene_LoadedWhenCurrentPlayerOwnsAndClicksThePVCSector | F3 | In the main game, if a sector containing a PVC landmark is clicked by the player who owns it, the minigameScene will become active. | Pass |

| | | | | |
|----|--|--------|--|------|
| V4 | Start_MinigameLoadsInDefaultState | F3 | When the minigame is loaded, the number of guesses, torsoChoice and legChoice equal 0. Also, the correct torso/legs values are within the range of the lists they point to. | Pass |
| V5 | SubmitGuess_AllGuessesIncorrectReturnToGame | F3, F4 | When the player makes 3 incorrect guesses, the minigame scene should be unloaded and the main game made active. | Pass |
| V6 | SubmitGuess_CorrectGuessReturnToGameAndIncreaseBeerKnowledgeValues | F3, F4 | When the player correctly guesses torso and legs, the minigame scene should be unloaded and the main game made active. Also, the player should gain +4 knowledge and +4 beer. | Pass |
| V7 | SubmitGuess_OneGuessFullyIncorrect | F3 | If the player guesses both the torso and legs incorrectly, then SubmitGuess should make the number of guesses equal to 1. It should also make the speech bubble text update, reporting that the player correctly guessed 0/2 items of clothing and has 2 guesses left. | Pass |
| V8 | SubmitGuess_OneGuessHalfIncorrect | F3 | If the player only guesses legs correctly, then SubmitGuess should make the number of guesses equal to 1. It should also make the speech bubble text update, reporting that the player correctly guessed 1/2 items of clothing and has 2 guesses remaining. | Pass |
| V9 | SubmitGuess_TwoGuessesFullyIncorrect | F3 | If the player guesses both the torso and legs incorrectly twice, then SubmitGuess should make the number of guesses equal to 2. It should also make the speech bubble text update, reporting that the player correctly guessed 0/2 items of clothing and has 1 guesses left. | Pass |

| | | | | |
|-----|---|----|--|------|
| V10 | TorsoLeftPressed_DecrementsTorsoChoiceOrLoopsToEnd | F3 | When TorsoLeftPressed is called, if the torsoChoice is zero, then the torsoChoice becomes equal to the size of the torso list -1. Otherwise, torsoChoice decreases by 1. | Pass |
| V11 | TorsoRightPressed_IncrementsTorsoChoiceOrLoopsToStart | F3 | When TorsoRightPressed is called, if the torsoChoice is equal to the length of the torso list -1, then torsoChoice becomes 0. Otherwise, torsoChoice increases its value by 1. | Pass |

CardDeck class tests (CardDeckTest):

| Test ID | Test Name | Related Requirements | Expected Outcome | Result |
|---------|--|----------------------|--|--------|
| D1 | ActivateCard_CardIsTakenFromPlayerAndActivated | F11, F12, F13 | When ActivateCard is called with a cardSlot corresponding to a test-card: The test-card should be added to the cardDeck's activeCards list. The test-card should be removed from its owner's punishmentCard list. The turnState should be set to EndOfTurn. | Pass |
| D2 | AssignPunishmentCard_GivesPlayerANewPunishmentCard | F11, F12 | When AssignPunishmentCard is called, the player's punishment card list should increase in size by 1 and the element added should not be null. | Pass |
| D3 | DeactivatePunishmentCards_NoActiveCards | F11 | If a player owns a card, but no active ones and DeactivatePunishmentCards is called, the cards should not be activated, the turncount of the cards should not change and the player should still own the cards. | Pass |
| D4 | DeactivatePunishmentCards_OneActiveCardToDeactivate | F11 | The active card owned by the player is removed from the activeCards list, the card's turnCount is 0 and it's owner becomes null. | Pass |
| D5 | DeactivatePunishmentCards_OtherPlayerHasActiveCardNoneToDeactivate | F11 | The there is an active card and | Pass |

| | | | | |
|-----|---|-----|---|------|
| | | | deactivatePunishmentCards is called for a player who doesn't own the card, the activeCards list contains the active card, the turncount and owner do not change. | |
| D6 | DeactivatePunishmentCards_TwoActiveCardsOneToDeactivate | F11 | Tests DeactivatePunishmentCards when there are two active cards owned by the player, one with a turnCount of 1, and the other with a turncount of 2. DeactivatePunishmentCards should remove the card with turncount = 1 and only decrease the turncount of the other by 1. | Pass |
| D7 | DeactivatePunishmentCards_TwoActiveCardsToDeactivate | F11 | Tests DeactivatePunishmentCards when there are two active cards owned by the player that need to be deactivated. Both cards should be removed from the activeCard list, their turnCounts = 0 and owner = null. | Pass |
| D8 | HasActiveCardOfType_FindsMatchingActiveCardReturnsTrue | | Test that HasActiveCardOfType returns true if a card matching the type specified is active. | Pass |
| D9 | HasActiveCardOfType_InvalidCardTypeReturnsFalse | | Tests that HasActiveCardOfType returns false if the type 'string' is used for the card type. | Pass |
| D10 | HasActiveCardOfType_NoMatchingActiveCardReturnsFalse | | Tests that HasActiveCardOfType returns false if the active card list contains a NothingCard and a FreshersFluCard is given as the card type. | Pass |
| D11 | HideMenu_MenuHiddenAndMapShown | | When HideMenu is called, the card deck menu is deactivated and the map gameobject is activated. | Pass |
| D12 | RemoveActiveCard_RemovesCardFromActiveCards | F11 | When RemoveActiveCard is called for the first card, the first card is removed | Pass |

| | | | | |
|-----|--|----------|---|------|
| | | | from the activeCards list and the other card remains active. When RemoveActiveCard is called for the second active card, the card is removed from the activeCards list, making it empty. | |
| D13 | SetActiveCard_AddsCardToActiveCardList | F12, F13 | When SetActiveCard is called, the card given is added to the activeCards list. When SetActiveCard is called with a different card, the activeCards list contains both cards. | Pass |
| D14 | ShowMenu_DoesNotShowMenu | F13 | If the TurnState is not Move1 or the currentPlayer is not human, cardDeck.menu will remain inactive and map will remain active. | Pass |
| D15 | ShowMenu_OneCardPlayableOneCardNot | F13 | If the current player owns a card that has not been activated, the card slot image will contain the card's image and the card slot will be interactable. If the current player owns a card that has been activated by another player, the card slot will contain the card's image, but the button's IsInteractable() is false. | Pass |
| D16 | ShowMenu_PlayerHasNoCards | F13 | When ShowMenu is called and the current player has no cards, all the cards slots will be non-interactable and will contain a null image. | Pass |

Card, NothingCard, FreshersFluCard, LecturerStrikeCard & KillerHangoverCard class tests (CardTest)

| Test ID | Test Name | Related Requirements | Expected Outcome | Result |
|---------|--------------------------------|----------------------|--|--------|
| C1 | Card_CorrectlyInstantiatesCard | F11 | When a NothingCard is instantiated, it's owner attribute is the player given in the constructor, the turnCount is 1 and the image is equal to Nothing image. | Pass |

| | | | | |
|----|--|---------------|--|------|
| C2 | Card_DeactivatePunishment_RemovesCardsOwner | | For each of the different types of card, when deactivatePunishment is called for that card, it's owner attribute is set to null. | Pass |
| C3 | Card_SetTurnCount_CorrectlyUpdatesValueOfTurnCount | | When SetTurnCount(10) is called for a specific card, GetTurnCount() returns 10. When SetTurnCount(-1) is called for a specific card, GetTurnCount() returns the same value as it previously did. | Pass |
| C4 | FreshersFluCard_ActivatePunishment_RemovesBonuses | N11, F11, F13 | When FreshersFluCard.activatePunishment is called, all of the players, except the owner of the card, have their Beer/Knowledge values set to 0. | Pass |
| C5 | FreshersFluCard_ActivatePunishment_StoresPvcBonuses | F13 | If player2 has +4 knowledge points on top of the landmark bonuses and player1 calls FreshersFluCard.activatePunishment, then the card's playerPvcBonuses[player2] = [0,4]. | Pass |
| C6 | FreshersFluCard_DeactivatePunishment_RecalculatesBonusesIfLandmarkCaptured | F13 | If player2 captures a knowledge landmark when player1's freshers flu card is active, when deactivatePunishment is called, player2 will receive a +2 knowledge bonus. | Pass |
| C7 | FreshersFluCard_DeactivatePunishment_ReturnsBonuses | F13 | When the FreshersFluCard is activated and DeactivatePunishment is called, all players bonuses will be returned to what they were before the card was activated. | Pass |
| C8 | KillerHangoverCard_EnemyPlayersMissTurn | N11, F11, F13 | When a killer hangover card's activatePunishment method is called, the current player remains equal to the owner of the card and the turnstate is Move1. Then, if the turn is ended, the turnstate = Move1 and the currentPlayer does not | Pass |

| | | | | |
|----|---|---------------|--|------|
| | | | change. | |
| C9 | LectureStrikeCard_EnemyPlayersOnlyHaveOneMove | N11, F11, F13 | When a lecturer strike card's activatePunishment method is called, the next player's turn turnState is Move1 and if game.NextTurnState is called, the turnState becomes EndOfTurn. | Pass |

Evidence:

The following two images are screenshots of the Unity test runner and show that all 95 unit tests pass.



