

## Changes Table

Change Description	Date & Time Pushed dd/mm/yy hh:mm	Changed by	Reason for change	Implemented in
<u>Fixed AI bugs and Unit Tests</u> Fixed a bug where the AI players would use the first move in the next player's turn. Added an if statement to Sector.OnMouseUpAsButton to prevent users from selecting AI units. Made various fixes to the Unit tests caused by them not being maintained during assessment 3.	28/02/18 10:31	Tom	Before starting to implement any of the new features, we needed to ensure that the current game worked correctly. Some errors were detected and fixes were made for them.	<i>Added:</i> GameTest.ClearSceneAfterTest PlayerTest.ClearSceneAfterTest SectorTest.ClearSceneAfterTest UnitTest.ClearSceneAfterTest  <i>Changed:</i> Game.CreatePlayers Player.playComputerMove Sector.OnMouseUpAsButton  GameTest.Setup PlayerTest.Setup SectorTest.Setup UnitTest.Setup
<u>Added ViceScriptTest file and fixed minigame bug</u> Added Unit tests for the PVC minigame. Added accessor methods for the number of guesses, correct torso id, correct legs id, torso choice and legs choice attributes.  Switched right and left increment/decrement for the torso arrow buttons to match the leg buttons. Fixed the left/right arrow buttons in the minigame.	28/02/18 15:51	Tom	Unit tests for the PVC minigame were added because they were not included by the previous team. This will allow us to quickly test that any new features do not break the minigame. The accessor methods were added to allow for more tests. The torso & leg arrow buttons were fixed so that the left/right buttons traverse the list of torsos/legs in opposite directions, not the same. Switched the torsoLeftPressed and torsoRightPressed so that they traverse the list of torsos in the same way the leg arrows do.	<i>Added:</i> ViceScriptTest viceScript.GetNumberOfGuesses viceScript.GetCorrectTorso viceScript.GetCorrectLegs viceScript.GetTorsoChoice viceScript.GetLegChoice  <i>Changed:</i> viceScript.torsoRightPressed viceScript.torsoLeftPressed viceScript.legsRightPressed viceScript.legsLeftPressed
<u>Added unit tests for SpawnVice and ComputerTurn</u> Unit tests were added to test the SpawnVice	04/03/18 10:53	Tom	These tests were added because no unit tests were included for the features added within the previous assessment.	<i>Added:</i> GameTest.SpawnVice_AddsAVice ChancellorLandmarkToARandom

method and the ComputerTurn functionality.				Sector GameTest.SpawnVice_AddsAVice ChancellorLandmarkToASpecificSector PlayerTest.ComputerTurn_MakesAValidMove PlayerTest.ComputerTurn_InvalidTurnStateNoMovelsMade
<u>Implemented a foundation for adding punishment cards</u> Firstly, an abstract class "Card" was added, which handles the basic card operations. It also contains the abstract method activatePunishment that forces subclasses of card to implement this method. A card deck was added along with a GUI card deck menu system. This system was then integrated into Player and Game.	06/03/18 20:34	Tom	Players now have the ability to store a list of cards. CardDeck has been added to handle the assignment of punishment cards, as well as the activation of cards via the GUI. The class Card was chosen because each of the punishment cards will have different abilities, but having them all inherit from a single class means that they can be seen as one type.	<i>Added:</i> Card NothingCard CardDeck Player.GetPunishmentCards Player.AddPunishmentCards Game.AssignPunishmentCard  <i>Changed:</i> Game.NextPlayer Game.Initialize
<u>Added card turn timeout and started FreshersFlu card</u> The AssignPunishmentCard method was moved from Game to CardDeck. The ability for cards to deactivate themselves after a certain number of turns was added to punishment card system.	10/03/18 21:37	Tom	Changes were made to Player.Capture to avoid a player's beer/knowledge values becoming negative, which could be caused by the addition of FreshersFluCard. To deactivate cards, there needed to be a way of tracking the active cards, hence CardDeck now has Get, Set and remove active card methods. The turn timeout was added so that the effects of the cards are not permanent, but are removed after the number of turns specified by Card.turnCount.	<i>Added:</i> FreshersFluCard Card.GetTurnCount Card.SetTurnCount Card.deactivatePunishment CardDeck.GetActiveCards CardDeck.SetActiveCard CardDeck.RemoveActiveCard CardDeck.AssignPunishmentCard CardDeck.DeactivatePunishmentCards  <i>Changed:</i> Card.Card NothingCard.NothingCard CardDeck.ShowMenu CardDeck.ActivateCard

				Game.EndTurn GameTest.Setup
<p><u>Implemented FreshersFluCard</u></p> <p>Now that the turn timeout system was implemented, the FreshersFluCard effects could be completed.</p> <p>Within activatePunishment, the card calculates the amount of bonus each player has, then removes the bonuses from the enemy players. Within deactivatePunishment, the card restores the bonuses, taking into account any changes in landmark ownership.</p>	14/03/18 20:00	Tom	<p>This has been added because it is a card that team decided to implement. The card takes into consideration changes in the players bonuses while the card is active by recalculating the bonuses gained from the landmarks owned.</p> <p>The card also colours the enemy units green, which gives a visual indication that the card is active and "infecting" units.</p>	<p><i>Added:</i> CardDeck.HasActiveCardOfType</p> <p><i>Changed:</i> FreshersFluCard.activatePunishment FreshersFluCard.deactivatePunishment Unit.Initialize</p>
<p><u>Added unit tests for the card system</u></p> <p>Unit tests for the CardDeck class and card-related Player methods were added to the project.</p> <p>Once created, a couple of errors were identified and so fixes were made to Player.AddPunishmentCard and CardDeck.DeactivatePunishmentCards.</p>	18/03/18 17:35	Tom	<p>In Player.AddPunishmentCard a check to see if the card being added belongs to the player was added.</p> <p>The other issue found was because DeactivatePunishmentCards was not deactivating all the punishment cards if more than 1 needed deactivating. This was because the code was removing cards from the activeCards list while iterating through it at the same time.</p>	<p><i>Added:</i> CardDeckTest CardDeck.GetCardSlots PlayerTest.GetPunishmentCards_ NewPlayerReturnsEmptyCardList PlayerTest.AddPunishmentCard_ CorrectlyAddsCardToList PlayerTest.AddPunishmentCard_ DoesNotAddCardIfListIsFull PlayerTest.AddPunishmentCard_ DoesNotAddCardIfCardIsNotOwnedByPlayer</p> <p><i>Changed:</i> CardDeck.DeactivatePunishmentCards Player.AddPunishmentCard</p>
<p><u>AI can now use punishment cards</u></p> <p>The computer players are now able to play punishment cards and was achieved by adding the ComputerPlayPunishmentCard method to Player.</p> <p>Additionally, unit tests for computer players using punishment cards were added.</p>	19/03/18 14:01	Tom	<p>The ability for the AI players to use punishment cards was added. The team felt it was necessary to do this so that human players do not have a large advantage over the AI.</p> <p>Also, unit tests were added to test this new feature because it makes it easier to ensure that the AI works correctly.</p>	<p><i>Added:</i> Player.ComputerPlayPunishmentCard PlayerTest.ComputerPlayPunishmentCard_CardPlayed PlayerTest.ComputerPlayPunishmentCard_MakesNormalMoveWhenNoCardsOwned</p>

				PlayerTest.ComputerPlayPunishmentCard_MakesNormalMoveWhenInvalidTurnState PlayerTest.ComputerPlayPunishmentCard_MakesNormalMoveWhenAllCardsOwnedAreActive  <i>Changed:</i> Player.Update PlayerTest.Setup
<u>Implemented card saving/loading</u> Saving and loading functionality was extended to allow saving and loading of punishment cards. In addition, the code of saving and loading functionality was improved since there were a number of repetitive statements that could have been written using loops.	22/03/18 21:27	Andrius	New requirement for the game - punishment cards. This functionality was implemented so players still have their punishment cards after saving and loading the game again.	<i>Changed:</i> GameControl.Save GameControl.Load
<u>Lecturer Striking Card Implemented</u> The new card was implemented with the functionality of missing a turn. The graphics for the new card were also added as well as the card generation for this card.	22/03/18 20:21	Kate	New requirement for the game - punishment cards. This card was implemented so the player missed their turn. As punishment cards are played at the start of the turn, the player would not be allowed to make any moves until their following turn.	<i>Added:</i> LecturerStrikeCard.activatePunishment
<u>Added Killer hangover card</u> The class <i>KillerHangoverCard</i> was added to the game. This represents the killer hangover card.	23/03/18 13:27	Steve	New requirement for the game - punishment cards. This card was implemented so other players miss their next turn.	<i>Added:</i> KillerHangoverCard.activatePunishment
<u>Added unit test for new cards</u> Nine unit tests were written for the Card, FreshersFluCard, LecturerStrikeCard and KillerHangoverCard classes. Also, changes were made to the LecturerStrikeCard functionality and the loading of the PunishmentCardGUI	23/03/18 18:50	Tom	The unit tests were added to ensure that the cards perform correctly under different scenarios. The changes to the SectorTest.Setup were made because some of the tests failed because cardDeck was not loaded.	<i>Added:</i> CardTest  <i>Changed:</i> Game.NextTurnState LecturerStrikeCard.activatePunishment

gameobjects were added to the SectorTest.Setup.				SectorTest.Setup
<u>Implemented postgrad bonus</u> Functionality that gives an edge to those players' sectors that are of level 5.	23/03/18 22:05	Andrius	New requirement for the game - postgraduates. When a sector reaches level 5, the player who owns it can move its units to other owned sectors, however far they may be. The bonus does not give the player the ability to attack non-adjacent sectors.	<i>Changed:</i> Sector.OnMouseUpAsButtonAccessible Unit.Select
<u>Removed unused code</u> Removed an unused variable	24/03/18 10:02	Andrius	The variable was never used in the game	<i>Changed:</i> Unit.SetLevel Unit.LevelUp
<u>Added missing code of postgrad bonus</u> Fixed the deselection of sectors. It did not work properly when the selected sector was of level 5 (postgrad bonus functionality)	24/03/18 10:32	Andrius	Forgot to adjust the code in Unit.Deselect() for the postgrad bonus functionality.	<i>Changed:</i> Unit.Deselect
<u>Added strike and hangover cards to saving/loading</u> Added two cards to the loading code.	24/03/18 11:03	Andrius	Not all cards were loaded properly. Killer hangover and lecturer strike cards had to be added to the loading code.	<i>Changed:</i> GameControl.Load KillerHangoverCard.KillerHangoverCard LecturerStrikeCard.LecturerStrikeCard
<u>Bug Fixes and strike visual effect added</u> An extra if statement was added to Sector.OnMouseUpAsButtonAccessible for the Postgrad movement. The LecturerStrikeCard functionality in Game.NextTurnState was changed in order to fix a bug. A model of people striking was added to the LecturerStrikeCard, which is loaded in activatePunishment and GameControl.Load, and is destroyed in deactivatePunishment.	24/03/18 17:59	Tom	The changes to Sector.OnMouseUpAsButtonAccessible were made because the Postgrad units were able to move to any sector in the map. They should have been restricted to owned or adjacent sectors. The changes to NextTurnState were made because before the LectureStrikeCard would only work if it was the most recent card played. The visual effects for LectureStrikeCard were added to give the users an indication that the card is active.	<i>Added:</i> LecturerStrikeCard.deactivatePunishment <i>Changed:</i> Game.NextTurnState Sector.OnMouseUpAsButtonAccessible LecturerStrikeCard.activatePunishment GameControl.Load

<u>Added tests for postgrad unit</u> Three unit tests were created to test the postgrad movement for the three main scenarios; the postgrad moves to an adjacent sector; the postgrad 'teleports' to a player-owned sector and the postgrad tries to move to a non-adjacent enemy sector. Some small updates to the code were also made, such as updating comments and removing duplication.	25/03/18 15:42	Tom	These tests ensure that the postgrad will be able to move as planned. Also, the duplication of Update in game was removed to make sure that the code remains maintainable and easy to understand.	<i>Added:</i> SectorTest.OnMouseUpAsButton_PostgradMovesToAdjacentSector SectorTest.OnMouseUpAsButton_PostgradMovesToPlayerOwnedSector SectorTest.OnMouseUpAsButton_PostgradCannotMoveToNonAdjacentEnemySector <i>Changed:</i> Game.Update Game.UpdateAccessible CardTest.Card_DeactivatePunishment_RemovesCardOwner
<u>Updated punishment card images</u> Added new images for all four punishment cards.	01/04/18 16:57	Tom	This was done because the previous card images were only placeholders and/or lacked descriptions.	<i>Changed:</i> Punishment card images