Change Table

Change Description	Date & Time Pushed dd/mm/yy hh:mm	Changed by	Reason for change	Implemented in

PVC spawning:
The position of the PVC can be preset for testing purposes in the Inspector window. The number of PVCs (at most 1) is checked at the beginning of each game (Game.InitializeMap() and Game.CountPVC()).
The method Game.SpawnPVC() is invoked at the end of each NextPlayer() call, and it spawns the PVC in a random unoccupied sector when numberOfTurns == delayPVCBy.
	27/01/18	Andrius	A required feature of assessment 3	*Added:*
				Game.numberOfTurns
				Game.PVCExists
				Game.delayPVCBy
				Game.SpawnPVC()
				Game.CountPVC()
				Sector.PVC
				Sector.GetPVC()
				Sector.SpawnPVC()
				Changed:
				Game.InitializeMap()
				Game.NextPlayer()

Spawning of Players: Changed how players are created in the game. The game used to create 'human' players by manually adding them as child GameObjects using the Unity Inspector window. Now, players are spawned when the game is initialized by creating the player GameObjects in code.	03/02/18 15:21	Tom	The previous implementation of the player spawning system was very limiting and this meant that all the players were of type 'Player'. However, when the NonHumanPlayer type was implemented this had to change because now players are a mix of 'Player' and 'NonHumanPlayer' types, and the number of each is not fixed.	*Added:* Player.SetUnitPrefab *Changed:* Game.CreatePlayers
Fixing Unit Tests: Due to the changes made to the game initialization, some of the tests failed because the SetUp() method was no longer correctly setting up a new game. Other changes were also made to the GameTest tests due to the changes made to GameTest.SetUp()	03/02/18 15:21	Tom	Changes to the game initialization caused some of the unit tests, provided for the preliminary implementation, to fail. Therefore, the setup for the tests was altered to accommodate the new, more flexible, game initialization.	*Changed:* GameTest.SetUp GameTest.NextPlayer_EliminatedPlayersAreSkipped GameTest.NoUnitSelected_ReturnsFalseWhenUnitIsSelected
NonHumanPlayer:	03/02/18 15:21	Tom	The NonHumanPlayer class has been	*Added:*

A NonHumanPlayer class has been added to the game and contains two methods, makeMove and findBestMove. findBestMove calculates the “best” move to make out of an array of possible moves (sectors), and makeMove executes the move. Game.nonHumanPlayerTurn is a new method which checks if the current player is a NonHumanPlayer, and if so, calls it’s makeMove method. Calls to nonHumanPlayerTurn have been added to Sector.MoveIntoUnoccupiedSector & Sector.MoveIntoFriendlyUnit.			added to the game because it represents the computer-controlled players outlined in requirement F1. makeMove is used to simulate a human player by calling findBestMove then using the OnMouseAsButtonAccessible method in Sector to make the move. Also, Game.nonHumanPlayerTurn & Game.UpdateAccessible work together to detect a NonHumanPlayer’s turn.	NonHumanPlayer.makeMove NonHumanPlauer.findBestMove Game.nonHumanPlayerTurn *Changed:* Game.UpdateAccessible Sector.OnMouseAsButton Sector.MoveIntoUnoccupiedSector Sector.MoveIntoFriendlyUnit
Menu UI: At first, a menu with two buttons “New Game” and “Load” is shown. The load button is shown only when there exists a saved game. When the new game button is pressed, the number of players can be chosen. When the start button is clicked, *Game.Initialize()* is called. By pressing the escape key on the keyboard while in game, the menu can be opened again, but this time “Continue” and “Save” buttons are shown. The continue button simply hides the menu, while the save button saves the current state of a game to *Application.persistentDataPath* in a file named “savedGame.gd” (saving & loading functionality implemented in *SavingNLoading.cs*)	13/02/2018	James, Andrius, Tom	A required part of assessment 3	*Added:* Menu UI elements Menu class SavingNLoading class (added this to the GameManager object too) Game.InitializeSectors() *Changed* Game.numberOfTurns to public Game.PVCExists to public Game.InitializeMap()

NonHumanPlayerTests: The playmode test script NonHumanPlayerTest.cs was added to the project. It contains four tests, a setup method, a createSector method and a teardown method.	11/02/18 13:51	Tom	This script tests that the class NonHumanPlayer functions correctly. MakeMove_executesValidMove tests if the computer player correctly executes a turn and the other three tests check if FindBestMove finds the best move in different situations.	*Added:* In NonHumanPlayerTest.cs: MakeMove_executesValidMove FindBestMove_OneBestMoveThreeBadMoves FindBestMove_AllMovesSameScore FindBestMove_NoPossibleMoves createSector Setup ClearSceneAfterTest
Passing The Turn: At the beginning of each game the 'Pass Turn' button is shown along with the player UI. When the button is pressed, Game.PassTurn method is invoked. Game.PassTurn method deselects currently selected unit and its adjacent units as well. If it didn't, the next player couldn't select their own sector, they would have to wait for the previous player to deselect it.	11/02/18 14:38	Andrius	A required part of assessment 3	*Added:* Game.PassTurn A new game object (UI) in player UI
SpawnPVC & PassTurn Test: Unit tests for the SpawnPVC and PassTurn have been added to GameTest.cs.	12/02/18 10:45	Tom	These tests have been added to ensure that the spawning of the PVC and the pass turn button work correctly.	*Added:* In GameTest.cs: SpawnPVC_spawnsAfter10Turns SpawnPVC_doesNotSpawnAfterFirstTurn SpawnPVC_doesNotSpawnMultiplePVCs SpawnPVC_PVCSpawnsInUnownedSector PassTurn_CorrectlyPassesTurn numberOfPVCSectors
Number of Players slider menu & gameover screen: A menu was added to the start of the game	12/02/18 20:37	Tom	This will allow the users to pick the number of human players that want to play the game. It means that the users	*Added:* Game.SetNumberOfPlayers

which lets the user pick the number of human players (2 to 4) that want to play a new game. A gameover screen has also been added to the menu, which is shown when the game ends. The necessary click events within the menu class were added to accommodate for these new features.			have a choice between playing against the AI or not. The gameover screen has been added to give the users an indication that the game has been won.	Menu.NewGameButtonClicked Menu.ExitButtonClicked Menu.ShowGameOverMenu *Changed:* Game.EndGame Game.Initialize Menu.StartButtonClicked

Added Dropper Game
Added the Dropper Game to the project to later be added to the capturing the PVC.
This consisted of adding two scripts one for spawning falling objects and one for moving the catcher.
	12/02/18 12:35	Haydn	This is was added to be played as a mini game to captcher the PVC and add a bonus for own him.	*Added:*
				MovementLR.SpawnItem()
				MovementLR.GetBeer()
				MovementLR.GetBook()
				MovementLR.Update()
				Dropper.Update()

Improved NonHumanPlayer and changed the position of landmarks: Some of the if statements within the NonHumanPlayer class have been changed. Now if the best move is equal to the new best, then it will choose the new best 50% of the time. Also, the position of landmarks have been changed so that they are on top of university buildings.	13/02/18 11:15	Tom	The changes to the NonHumanPlayer have been done because before the AI was getting stuck in loops. Now the chance of this is significantly reduced. The positions of landmarks have been altered in order to meet requirement N7, which states that the Landmarks must correspond with real-life landmarks at the University of York.	*Changed:* NonHumanPlayer.makeMove NonHumanPlayer.findBestMove Game.EndGame

Linked Dropper Game to PVC sector
The Dropper Game was linked to the PVC sector so when it was captured the Dropper game would run.
	13/02/18 20:45	Haydn	This change was to add to access to the dropper game in common play so that it started when a player moved into the sector with the PVC.	*Added:*
				MovementLR.StartDropperGame()
				MovementLR.Start()
				MovementLR.StopDropperGame()
				Updated:
				Unit.MoveTo
				MovementLR.SpawnItem()

Apply score bonus to PVC
Linked the outcome to of Dropper Game to the owners Beer score and Knowledge Score and on change of ownership.
	14/02/18 18:48	Haydn	This change adds the effect of owning the sector with the PVC by increasing the level of Beer and Knowledge.	*Added:*
				MovementLR.GetLastBeer()
				MovementLR.GetLastBoo()
				addScoreFromDropper()

				Updated: MovementLR.SpawnItem()
<u>Refactoring and dropper game bug fixes</u> Comments were added to MovementLR, Dropper, Game and Unit. Duplication in the MovementLR.SpawnItem was eliminated and some unnecessary attributes were removed. Added: If the AI captures the PVC, they do not play the minigame, but are given a fixed bonus of 2 books & 2 beers. Added a check to Game.Update to pause the game if the PVC minigame is running. Added a call to UpdateDisplay in Unit.addScoreFromDropper.	14/02/18 21:37	Tom	Refactoring was carried out to clean up unnecessary code and make it more readable. The game.Update method only calls UpdateAccessible when movementLR.stopped is true. This prevents the AI from making moves while the minigame is running. The call to UpdateDisplay was added so that beer/knowledge values in the player GUI cards are updated after the minigame bonuses are given.	*Changed:* Game.Update MovementLR.Start MovementLR.StartDropperGame MovementLR.StopDropperGame MovementLR.SpawnItem Unit.MoveTo Unit.addScoreFromDropper *Removed:* MovementLR.CheckDropperFinished