

Test ID	Test Description	Expected Result	Result	Evidence ID (Appendix)
	MapClass Tests			
U1	deselectAll_deselects_sector: <i>Tests if deselectAll deselects a sector and returns its colour back to the owner's colour.</i> 1. Load main game. 2. Find a sector in the map. 3. Set Selected = true & the colour to (0,0,0). 4. Run MapClass.deselectAll	The chosen sector's Selected attribute should be set to false. The chosen sector's colour should be equal to the sector owner's colour.	Pass	UE 1
U2	deselectAll_does_not_change_deselected_sectors: <i>Tests if deselectAll doesn't change non-selected sectors.</i> 1. Load main game. 2. Find a sector in the map. 3. Set Selected = true & the colour to the sector owner's colour. 4. Run MapClass.deselectAll	The chosen sector's Selected attribute should be set to false. The chosen sector's colour should be equal to the sector owner's colour.	Pass	UE 1
U3	colourSectors_colours_all_sectors_to_owner_colour: <i>Tests if colourSectors works correctly.</i> 1. Load main game. 2. Colour all sectors (0,0,0). 3. Run MapClass.colourSectors	The colour of each sector in the map should be equal to the sector's owner's colour.	Pass	UE 1
U4	getSelectedSector_returns_correct_sector: <i>Tests if getSelectedSector returns a selected sector.</i> 1. Load main game. 2. Find a sector in the map. 3. Set Selected = true. 4. Run MapClass.getSelectedSector().	The sector returned by getSelectedSector() should be the same as the chosen sector.	Pass	UE 1
U5	getSelectedSector_no_selected_sectors: <i>If there are no selected sectors, getSelectedSector returns null.</i> 1. Load main game. 2. Run getSelectedSector	The output of getSelectedSector() should be null.	Pass	UE 1
	GameClass Tests			
U6	changeTurn_moves_to_next_player_basic_case: <i>Tests if changeTurn moves to the next player in the Player list, where the next player is in the next position in the list.</i> 1. Load main game. 2. Run changeTurn().	The new CurrentPlayer should be the same as GameClass.Players[1].	Pass	UE 1
U7	changeTurn_moves_to_next_player_loop_case	The new	Pass	UE 1

	e: <i>Tests if changeTurn loops back to the first player.</i> 1. Load main game. 2. Run changeTurn() GameClass.Players.Count times.	CurrentPlayer should be the same as GameClass.Players[0].		
U8	currentPlayer_on_start: <i>Tests if CurrentPlayer returns Players[0] after the game starts.</i> 1. Load main game. 2. Assert expectedPlayer is GameClass.CurrentPlayer.	GameClass.CurrentP layer should be the same as GameClass.Players[0].	Pass	UE 1
	PlayerClass			
U9	playerClass_instantiation: <i>Tests if the PlayerClass constructor correctly stores name & colour.</i> 1. Create a new instance of PlayerClass with name = "TestPlayer" and color = (0.5,0.5,0.5).	The new player's name = "TestPlayer". The new player's colour = (0.5,0.5,0.5)	Pass	UE 1
U10	playerClass_allocated_initialised_as_false: <i>Tests if allocated is set to false on initialisation.</i> 1. Create a new instance of PlayerClass. 2. Test Allocated == false.	The new player's Allocated attribute should equal false.	Pass	UE 1
	Sector Tests			
U11	sector_creation_2_sectors: <i>Tests if sectors are initialised with the correct adjacent sectors for a map with 2 sectors.</i> 1. Create a new scene with a GameObject named "Map". 2. Create 2 inactive sectors adjacent to each other. 3. Activate the sectors.	The first sector's adjacent sector list should only contain sector 2. The second sector's adjacent sector list should only contain sector 1.	Pass	UE 1
U12	sector_creation_3_sectors_complete_graph: <i>Tests if sectors are initialised with the correct adjacent sectors for a map with 3 sectors, where each sector is adjacent to all other sectors.</i> 1. Create a new scene with a GameObject names "Map". 2. Create Sector1, with adjacent sector IDs 2 & 3. 3. Create Sector2, with adjacent sector IDs 1 & 3. 4. Create Sector3, with adjacent sector IDs 1 & 2. 5. Activate sectors & test lists.	Sector1.AdjacentSectors == List [Sector2, Sector3] Sector2.AdjacentSectors == List [Sector1, Sector3] Sector3.AdjacentSectors == List [Sector1, Sector2].	Pass	UE 1
U13	sector_creation_3_sectors_uncomplete_graph: <i>Tests if sectors are initialised with the correct adjacent sectors, where some sectors are</i>	Sector1.AdjacentSectors == List [Sector2] Sector2.AdjacentSectors	Pass	UE 1

	<i>adjacent to more than others.</i> 1. Create a new scene with a GameObject names "Map". 2. Create Sector1, with adjacent sector IDs 2. 3. Create Sector2, with adjacent sector IDs 1 & 3. 4. Create Sector3, with adjacent sector IDs 2. 5. Activate sectors & test lists.	ors == List [Sector1, Sector3] Sector3.AdjacentSectors == List [Sector2].		
U14	clickSector_highlight_adjacent_sectors_attack_phase: <i>Tests that the correct adjacent sectors are highlighted when a current player owned sector is clicked during the attack phase.</i> 1. Load main game. 2. Find a sector owned by current player. 3. Set current player allocated to true, & set GameState to ATTACK. 4. Run sector.clickSector().	Foreach adjacent sector: If the sector is not owned by the player, it's colour should equal (1,0,0), otherwise should be equal to the owner's colour. The sector clicked should become selected if there are valid moves.	Pass	UE 1
U15	clickSector_highlight_adjacent_sectors_movement_phase: <i>Tests that the correct adjacent sectors are highlighted when a current player owned sector is clicked during the movement phase.</i> 1. Load main game. 2. Find a sector owned by current player. 3. Set current player allocated to true, & set GameState to MOVEMENT. 4. Run sector.clickSector().	Foreach adjacent sector: If the sector is owned by the player, it's colour should equal (1,1,1), otherwise should be equal to the owner's colour. The sector clicked should become selected if there are valid moves.	Pass	UE 1
U16	clickSector_unowned_sector: <i>Tests that if a sector not owned by the current player is clicked, it should not become selected.</i> 1. Load main game. 2. Find a sector not owned by the current player. 3. Run sector.clickSector().	The clicked sector's Selected attribute == false. The clicked sector's colour == it's owner's colour.	Pass	UE 1
U17	clickSector_invalid_second_sector_clicked_attack_phase: <i>Tests if no changes occur when a player clicks one of their own adjacent sectors after selecting a sector during an attack phase.</i> 1. Load main game. 2. Find a sector owned by the current player. 3. Set current player's allocated to true & set GameState to ATTACK. 4. Run clickSector() for chosen sector. 5. Find a current-player-owned sector adjacent to the first. 6. Run clickSector() for the second sector.	In the first sector clicked: Selected = true. In the second sector clicked: Selected = false. All sectors adjacent to the first and not owned by the current player should be highlighted in (1,0,0), other adjacent sectors should have their owner's colour	Pass	UE 1

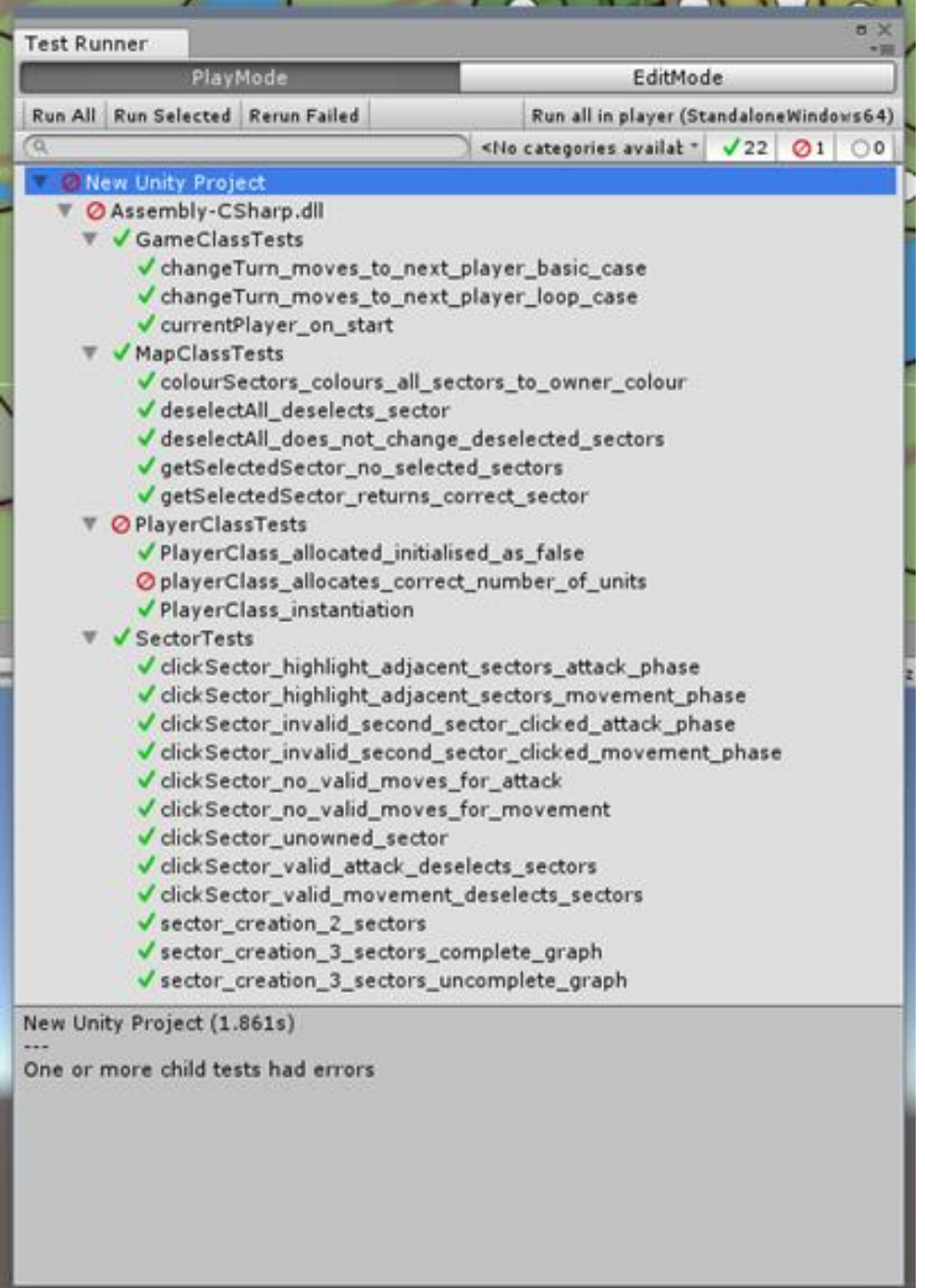
U18	<p>clickSector_invalid_second_sector_clicked_movement_phase: <i>Tests if no changes occur when a player clicks an enemy's adjacent sectors after selecting a sector during a movement phase</i></p> <ol style="list-style-type: none"> 1. Load main game. 2. Find a sector owned by the current player. 3. Set current player's allocated to true & set GameState to MOVEMENT. 4. Run clickSector() for chosen sector. 5. Find an enemy-owned sector adjacent to the first. 6. Run clickSector() for the second sector. 	<p>In the first sector clicked: Selected = true. In the second sector clicked: Selected = false. All current player owned sectors adjacent to the first should be highlighted in (1,1,1), other adjacent sectors should have their owner's colour.</p>	Pass	UE 1
U19	<p>clickSector_valid_attack_deselects_sectors: <i>Tests if clicking an adjacent enemy sector, after clicking an owned sector, deselects the first sector & removes highlighting.</i></p> <ol style="list-style-type: none"> 1. Load main game. 2. Find a sector owned by the current player. 3. Set current player's allocated to true & set GameState to ATTACK. 4. Run clickSector() for chosen sector. 5. Find an enemy-owned sector adjacent to the first. 6. Run clickSector() for the second sector. 	<p>There should be no selected sectors within the map (getSelectedsector() = null). The sectors adjacent to the first clicked sector should be coloured in their owner's colour.</p>	Pass	UE 1
U20	<p>clickSector_valid_movement_deselects_sectors: <i>Tests if clicking an adjacent owned sector, after clicking an owned sector, deselects the first sector & removes highlighting.</i></p> <ol style="list-style-type: none"> 1. Load main game. 2. Find a sector owned by the current player. 3. Set current player's allocated to true & set GameState to MOVEMENT. 4. Run clickSector() for chosen sector. 5. Find an owned sector adjacent to the first. 6. Run clickSector() for the second sector. 	<p>There should be no selected sectors within the map (getSelectedsector() = null). The sectors adjacent to the first clicked sector should be coloured in their owner's colour.</p>	Pass	UE 1
U21	<p>clickSector_no_valid_moves_for_attack: <i>Tests that nothing is selected or highlighted when a sector that has no valid attack moves from it is clicked.</i></p> <ol style="list-style-type: none"> 1. Load main game. 2. Find a sector owned by the current player. 3. Change owner of all adjacent sectors to current player. 4. Set current player's allocated to true & set GameState to ATTACK. 5. Run clickSector() for chosen sector. 	<p>There should be no selected sectors within the map (getSelectedsector() = null). The sectors adjacent to the clicked sector should be coloured in their owner's colour.</p>	Pass	UE 1
U22	<p>clickSector_no_valid_moves_for_movement: <i>Tests that nothing is selected or highlighted when a sector is clicked that has no valid movement moves from it.</i></p>	<p>There should be no selected sectors within the map (getSelectedsector())</p>	Pass	UE 1

	1. Load main game. 2. Set otherPlayer = CurrentPlayer. 3. Change the turn using GameClass.changeTurn(). 4. Find a sector owned by the current player. 5. Change owner of all adjacent sectors to otherPlayer. 6. Set current player's allocated to true & set GameState to MOVEMENT. 7. Run clickSector() for chosen sector.	= null). The sectors adjacent to the clicked sector should be coloured in their owner's colour.		
--	--	--	--	--

Failed Tests:

Test ID	Test Description	Expected Result	Result	If fail: Additional notes	Rerun result	Evidence ID (Appendix)
U23	playerClass_allocate_s_correct_number_of_units: <i>Tests that the players are assigned the correct number of gang members.</i> 1. Load main game. 2. Get the current player. 3. Count number of sectors owned by player. 4. Multiply number of sectors by 10. (Ten units allocated per sector)	The current player's GangMembersLeft attribute should be equal to the number of sectors owned by the current player multiplied by 10.	Fail	Player owned 37 sectors so expected a value of 370 for GangMembersLeft. Actual Value = 50. GangMembersLeft value is fixed at 50 and does not currently depend on the number of sectors owned by the player. To fix problem, added giveNewGangMembers method to PlayerClass and allocateGangMembersToPlayers method to GameClass.	Pass	UE 2

Appendix:

Evidence ID	Evidence Of Testing
UE 1	 <p>The screenshot shows the Unity Test Runner interface. At the top, there are tabs for 'PlayMode' and 'EditMode', with 'EditMode' currently selected. Below the tabs are buttons for 'Run All', 'Run Selected', 'Rerun Failed', and 'Run all in player (StandaloneWindows64)'. A search bar is present with the text '<No categories available>'. To the right of the search bar, there are three circular icons: a green one with '22', a red one with '1', and a grey one with '0'. The main area displays a tree view of test results for 'New Unity Project' (indicated by a red circle with an 'X'). The tree is expanded to show the following structure:</p> <ul style="list-style-type: none"> Assembly-CSharp.dll <ul style="list-style-type: none"> GameClassTests <ul style="list-style-type: none"> changeTurn_moves_to_next_player_basic_case (green checkmark) changeTurn_moves_to_next_player_loop_case (green checkmark) currentPlayer_on_start (green checkmark) MapClassTests <ul style="list-style-type: none"> colourSectors_colours_all_sectors_to_owner_colour (green checkmark) deselectAll_deselects_sector (green checkmark) deselectAll_does_not_change_deselected_sectors (green checkmark) getSelectedSector_no_selected_sectors (green checkmark) getSelectedSector_returns_correct_sector (green checkmark) PlayerClassTests <ul style="list-style-type: none"> PlayerClass_allocated_initialised_as_false (green checkmark) playerClass_allocates_correct_number_of_units (red circle with 'X') PlayerClass_instantiation (green checkmark) SectorTests <ul style="list-style-type: none"> clickSector_highlight_adjacent_sectors_attack_phase (green checkmark) clickSector_highlight_adjacent_sectors_movement_phase (green checkmark) clickSector_invalid_second_sector_clicked_attack_phase (green checkmark) clickSector_invalid_second_sector_clicked_movement_phase (green checkmark) clickSector_no_valid_moves_for_attack (green checkmark) clickSector_no_valid_moves_for_movement (green checkmark) clickSector_unowned_sector (green checkmark) clickSector_valid_attack_deselects_sectors (green checkmark) clickSector_valid_movement_deselects_sectors (green checkmark) sector_creation_2_sectors (green checkmark) sector_creation_3_sectors_complete_graph (green checkmark) sector_creation_3_sectors_uncomplete_graph (green checkmark) <p>At the bottom of the window, a console log shows the following text:</p> <pre>New Unity Project (1.861s) --- One or more child tests had errors</pre>

