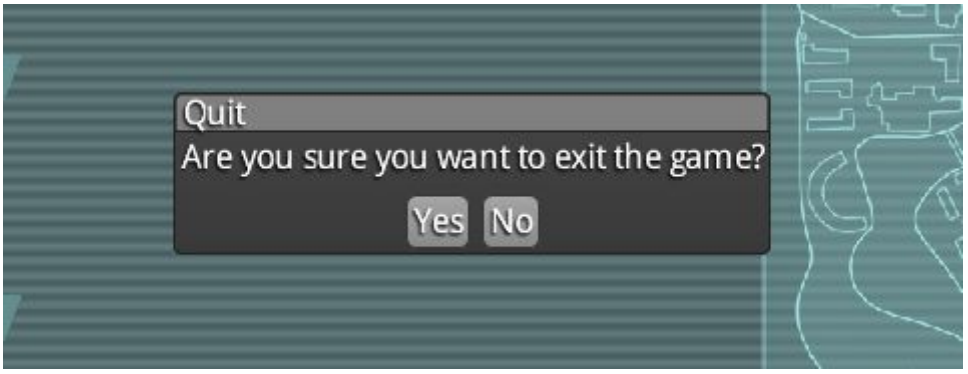
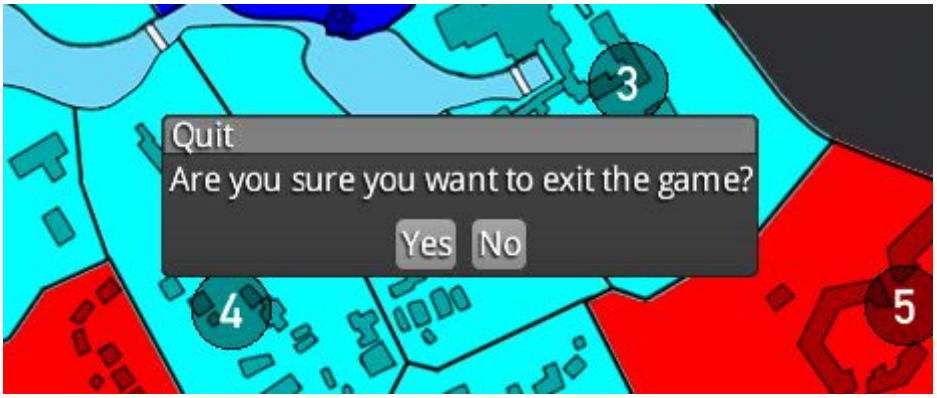
DialogFactory class

ID	Method	Expected Outcome	Actual Outcome	Pass/Fail
1	basicDialogBox	Test Basic Dialog Box works mid-game	Dialog box with the correct message and title is displayed	Pass
Result of basicDialogBox when called from phaseReinforcement.touchUp				
2	nextTurnDialogBox	The dialog box should show the next player and number of troops to allocate	Dialog box shows the next player and the number of troops to allocate	Pass
Result of nextTurnDialogBox when called from PhaseReinforce.enterPhase				
3	Exit Program Dialog Box	The dialog box should give a yes no option to quit the game, and upon clicking yes the application should close	Dialog box quits the application when yes is clicked, and has no action if no is clicked	Pass



Result of exitProgramDialogBox when “Quit” on the main menu screen is clicked, closing the program when yes is clicked, and closing the dialog when no is clicked.

4	leaveGameDialog Box	Dialog should give yes no for leaving the current game	Dialog box calls the openMenu function resetting the game and loading the menu	Pass

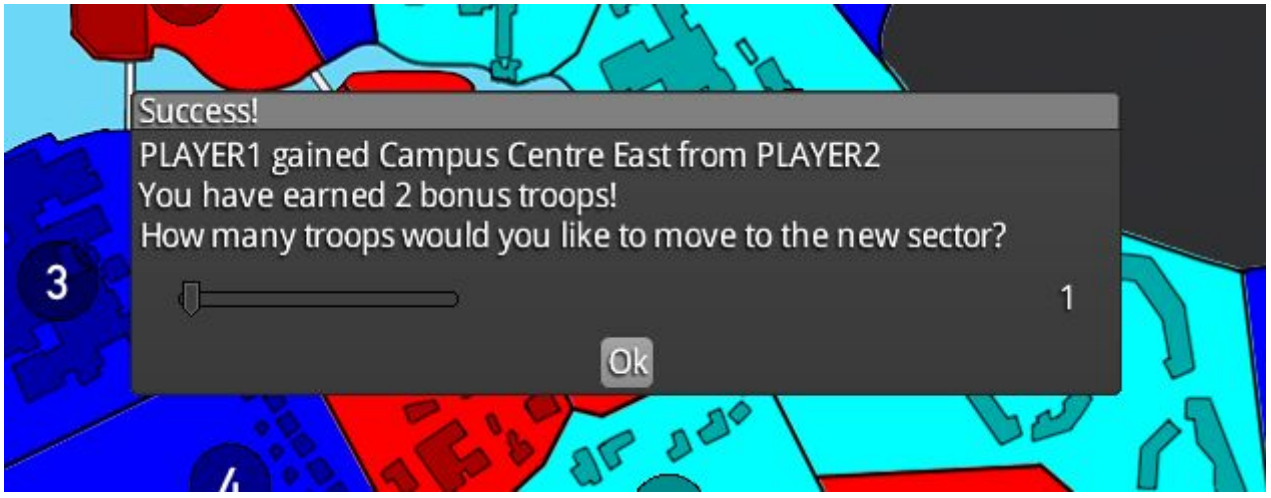


Result of leaveGameDialogBox when clicking “Quit” in the game. Main menu is loaded when yes is clicked, dialog box is closed when no is clicked.

5	sectorOwnerChange Dialog	Dialog should show the new owner for the sector	Dialog is never drawn as the conditions in Map.attackSector never reaches the sectorOwnerChangeDialog call.	Fail
6	attackDialog	Slider should be shown allowing a range of troops to be selected depending on the number of attacking troops available	Slider ranging between 1 and the number of troops attacking works, and when clicking ok the game moves onto the next stage.	Pass

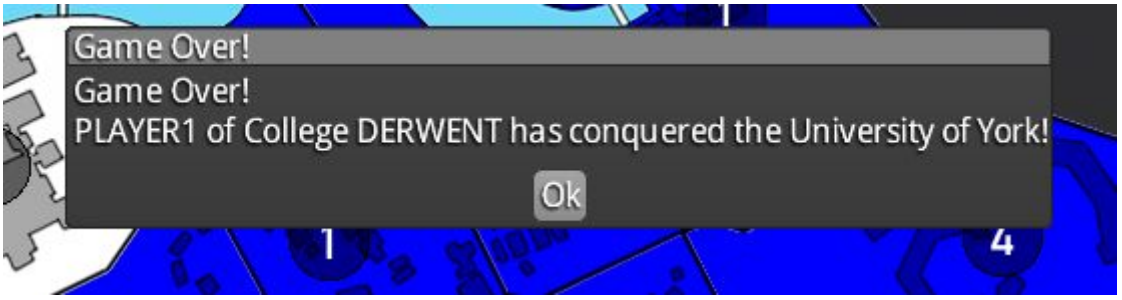
Result showing attackDialog and a slider with possible values to attack with, called in PhaseAttack.getNumberOfAttackers.

7	attackSuccessDialog Box	Slider should appear showing the possible number of troops to move to the defeated tile.	Slider ranging between 1 and the number of troops available to move should appear.	Pass



Result of attackDialogBox showing the range of troops to transfer, max here is 1 as only 2 troops survived the attack. Called in

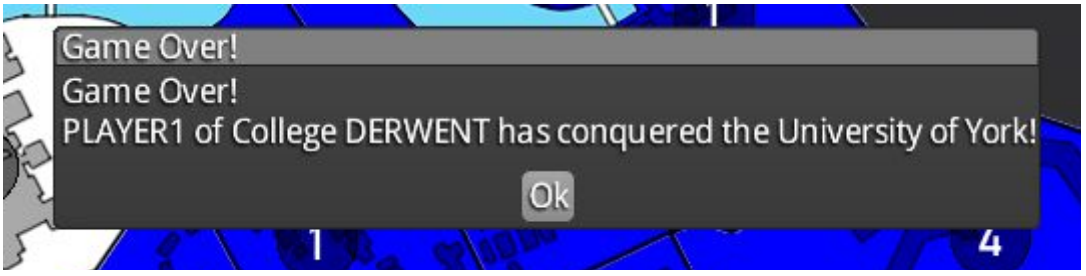
8	gameOverDialog	Dialog should be displayed when the game has concluded, displaying the winner and moving to the main screen.	Dialog shows as expected didplaying the relevant information	Pass



When playing the game to completion the winner was shown this message and on pressing ok the game

proceeded to the main menu.

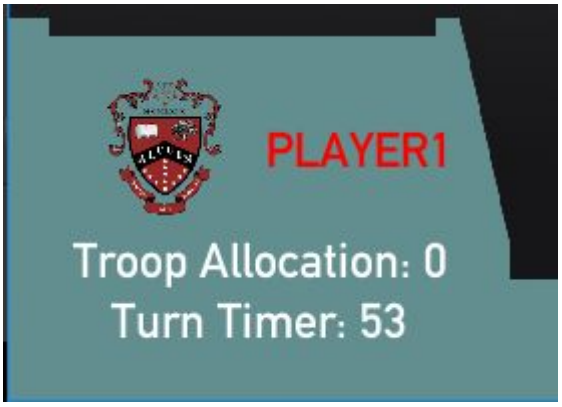
GameScreen class

ID	Method	Expected Outcome	Actual Outcome	Pass/Fail
1	getTurnTimeRemaining	Return an integer of seconds remaining in the game	The correct number of seconds is returned	Pass
 The method is called in GameScreen.render and draws the current remaining time to the UI correctly.				
2	isGameOver	Return True if the game is concluded, false if not	The method returned true when one player defeated the other	Pass
 When the function is called in removeEliminatedPlayers and returns true, gameOver is called ending the game and drawing the end game dialog box.				
3	getCurrentPlayer	Return the Player whose turn it currently is	The current player is returned	Pass
 nextPhase calls this method and uses the return to enter the next phase for the current player.				

4	nextPhase	Depending on the current phase, once the method is called the current phase should progress by one, and if it reaches Movement for player A, player B's turn should begin.	Once end phase is clicked the current phase progresses to the next.	Pass

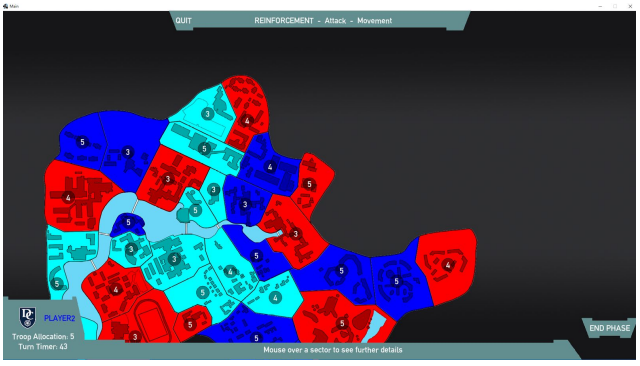
Once end phase is clicked the current phase progresses to the next.

5	nextPlayer	When called the currentPlayerPointer should increment by one, unless it reaches the maximum number of players, in which case it should restart to 0.	When end phase was clicked while in the movement phase, the current player changed to the next in the queue.	Pass

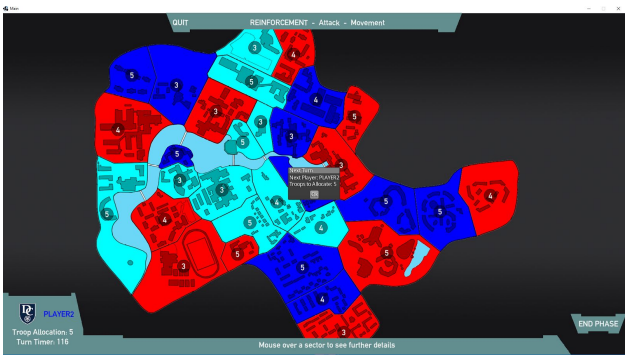


When End Phase clicked the currentPlayerPointer is incremented so other methods such as the render for the player name can access the new property.

6	controlCamera	When up, down, left and right are pressed the camera should move in the respective direction	When the up arrow was pressed the camera moved up until the key was released	Pass



7	resetCameraPostition	When the method is called the camera should reset to level 1 zoom and the map should be centered	When the method is called the map view resets	Pass



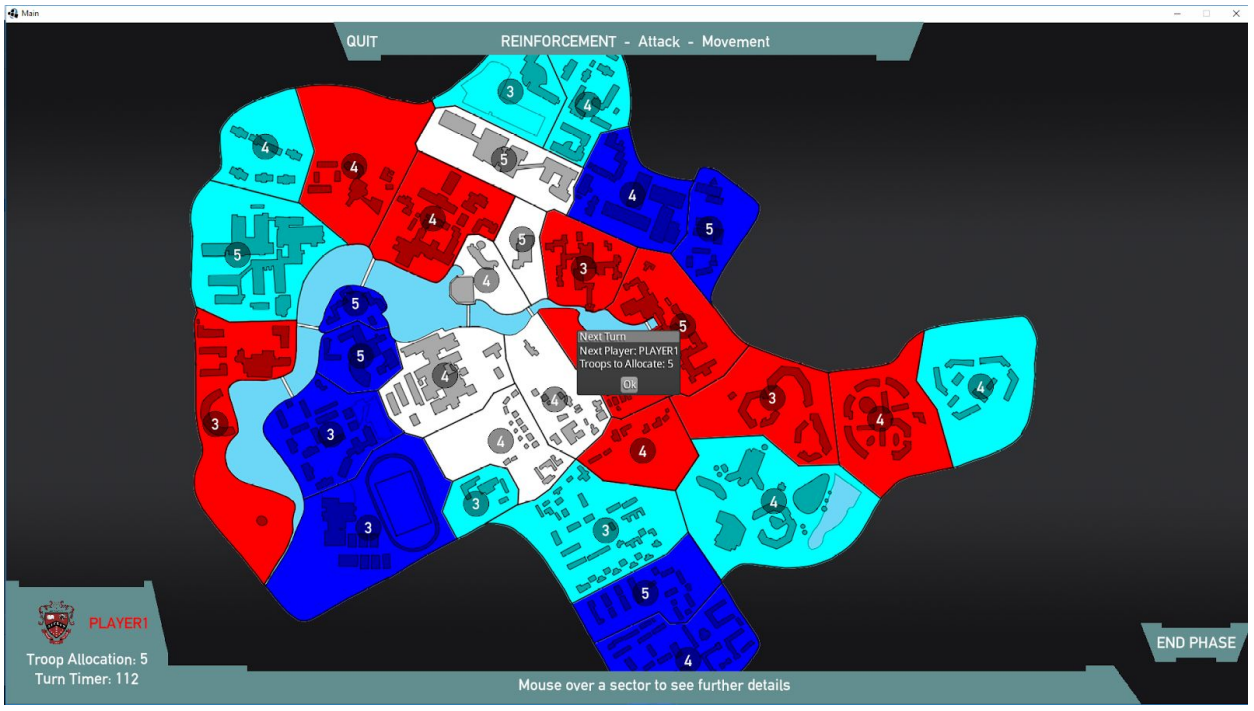
When nextPlayer is called due to end phase being clicked and the current player updates, the method is called and the camera view resets.

8	render	This method should update the game and render it to the screen, every delta time	The screen updates smoothly and seamlessly, to give the illusion of moving images and to update the UI	Pass

The turn timer is as it is called from the render method.

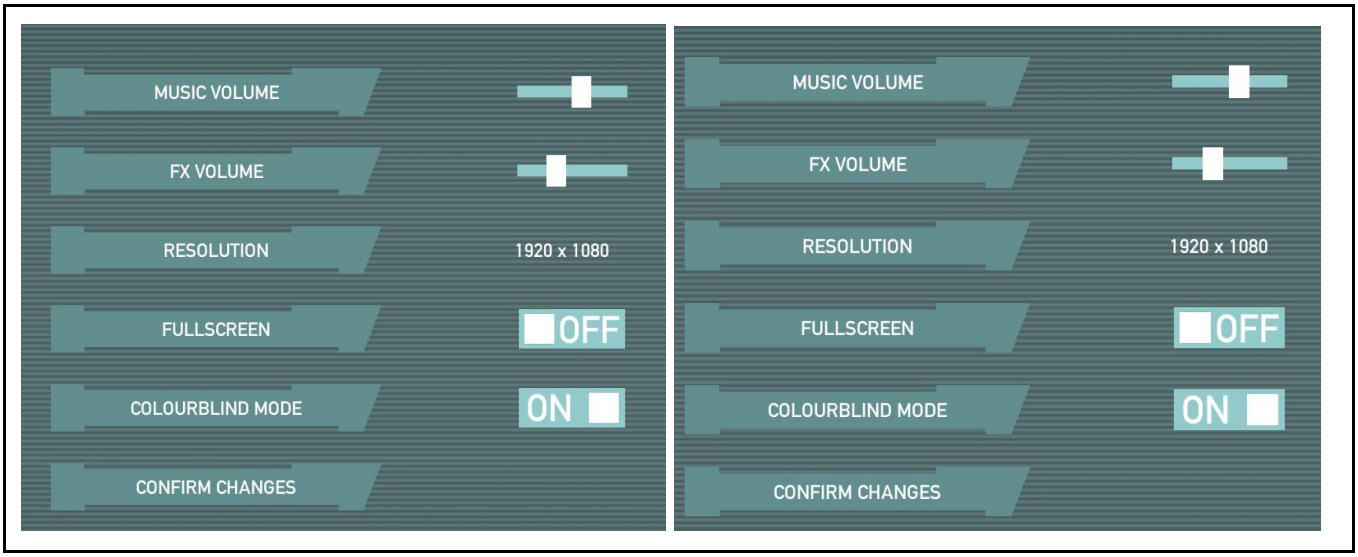
Main Class

1	create	Should create the stages and initialise DialogFactory and WidgetFactory	The game works after this point and therefore the method is functioning as it establishes all aspects of the game	Pass
2	setGameSetupScreen	The players entered in the start screen should be loaded into the main game, with turn timer and neutral player parameters established	The game successfully sets up	Pass



You can see the settings defined in the setup screen are also seen in the actual game

3	applyPreferences	The set preferences should be used and upon reloading the game the last state should be kept.	When setting the preferences and clicking Confirm Changes the options take place and are still in place once the game is reloaded.	

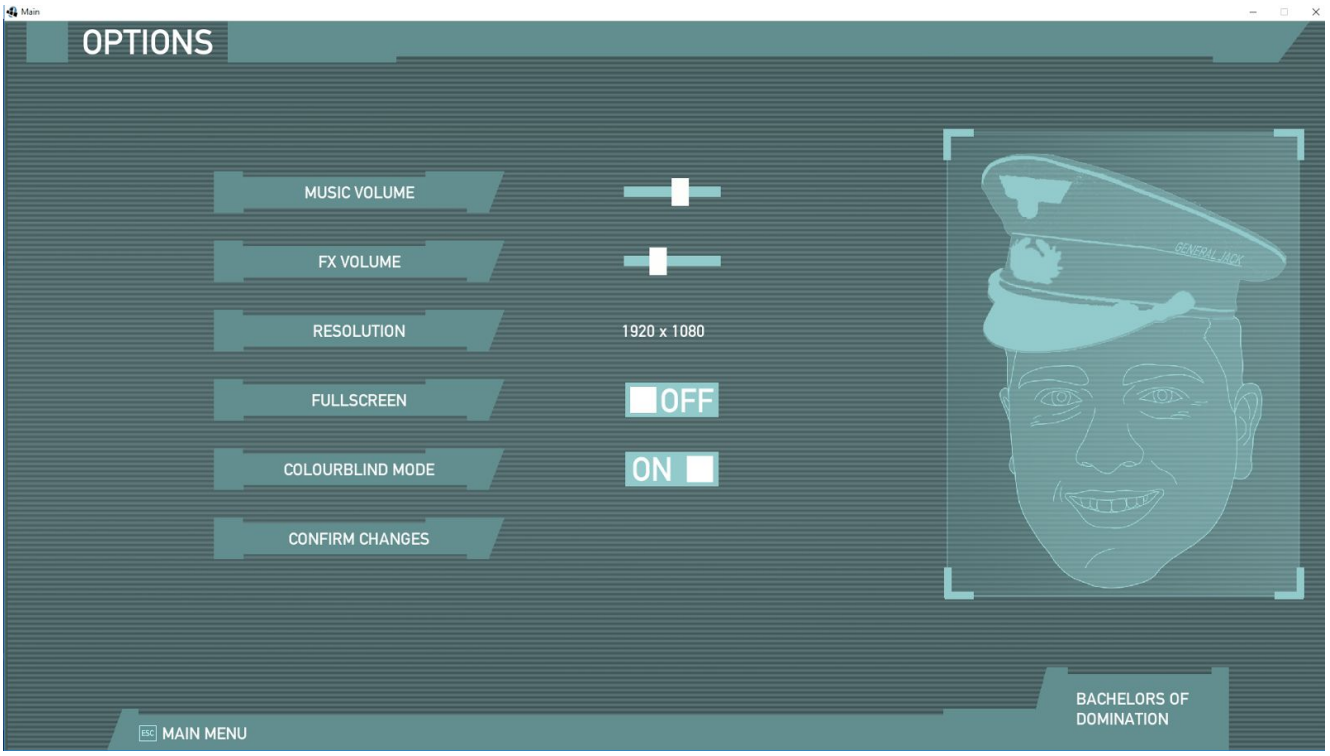


OptionsScreen class

1a	getPossibleResolutions	Should generate an array of possible resolutions that the monitor supports.	The list generated created a list that allowed resolutions that are impractically small for the amount of UI we have, and therefore created scaling bugs	Fail

Photo shows that getPossibleResolutions created an array that when called caused the scaling of the game to be impractical and unuseable.

1b	getPossibleResolutions	Should generate an array of possible resolutions that the monitor supports.	The list created is now limited to a minimum of 1000x1000, and therefore has solved the error we had before, and allows for nicely	Pass

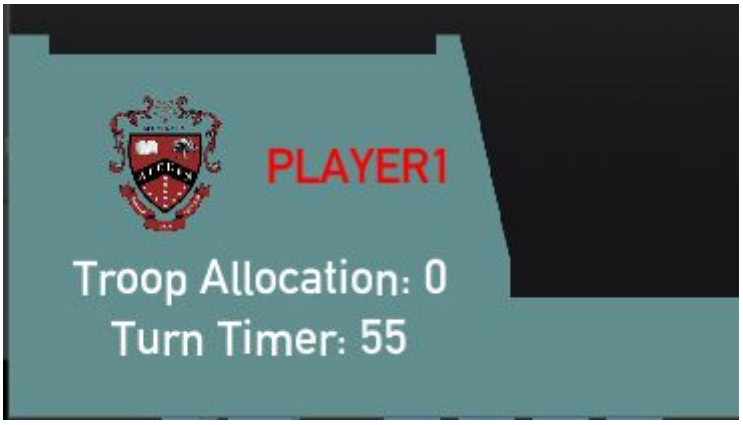
			scaled game windows.	
2	setupUi	This method should create an option screen that has sound volumes, resolution, full screen and colourblind mode	The method generates all the necessary components for the option menu	Pass
				

Phase

1	setBottomBarText	The bottom bar of the UI should show the details of the sector that is passed to it, a basic message if nothing is passed.	The UI bar is sufficiently drawn depending on the methods parameters when called	Pass

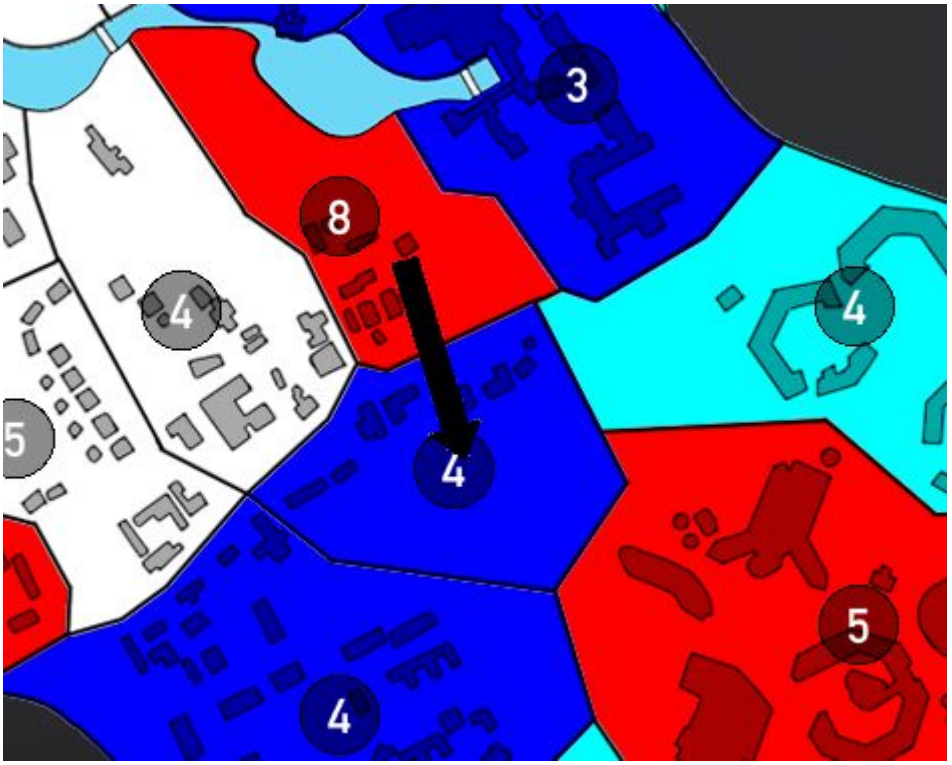
When called with the sector the mouse is hovered over you get the above result. The first message is of no sector is detected.

2	genGameHUDBottomBarLeftPart	Should create a bar with the logo, player name, troop allocation and turn timer.	All the data is present in the bar	Pass



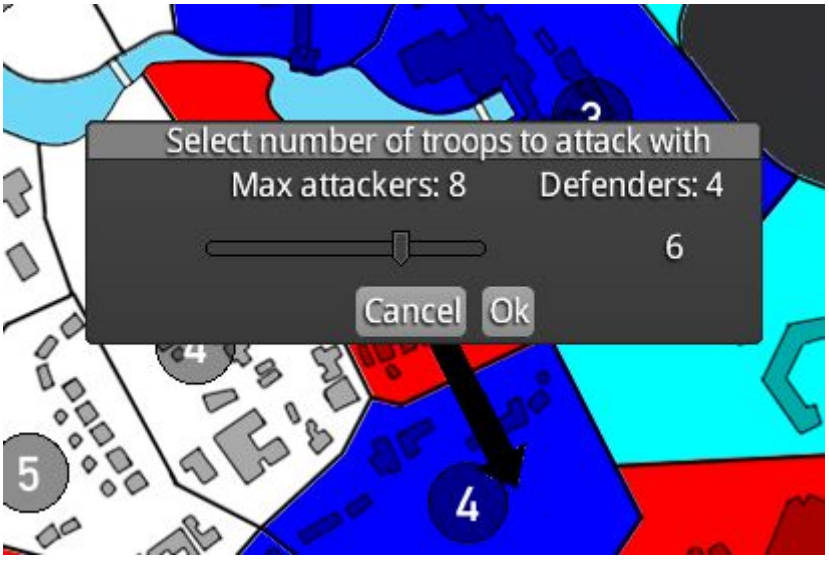
PhaseAttack

1	generateArrow	An arrow should be rendered starting from startX and startY, and ending on endX and endY.	An arrow is drawn.	Pass

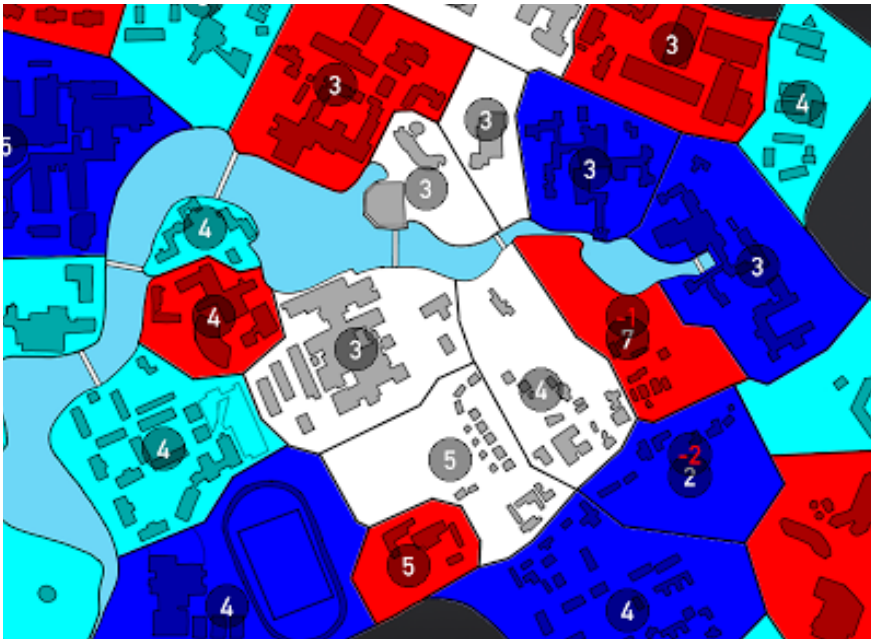


When called from PhaseAttack the arrow start is the attacking sector and the end is the defending sector, clearly shown to work here.

2	getNumberOfAttackers	Should generate a dialog with slider for the range of troops its possible to attack with.	The slider is accurately drawn and the range is as expected.	Pass



3	executeAttack	Should remove a calculated number of troops from the attacker and defender based on a formula set for fair balancing	The number of troops lost on each is a fair number and is well calculated	Pass



-1 and -2 animations are shown, where the attacking sector has more troops making a realistic success rate.

WidgetFactory

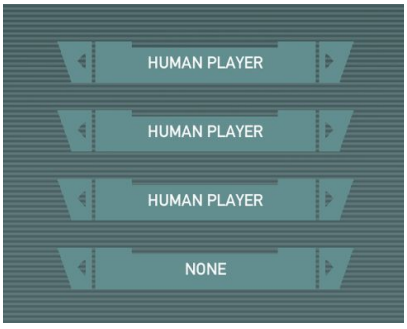
1	genStartGameButton	A button positioned according to GUI design should be displayed saying start game, and when pressed starts the game	The button is drawn and clickable	Pass
2	genEndPhaseButton	A button positioned according to GUI design should be displayed saying end phase, and when pressed ends the current phase	The slider is accurately drawn and the range is as expected.	Pass
3	genBottomBar	Should create a bar in the home	The bar is as expected	Pass

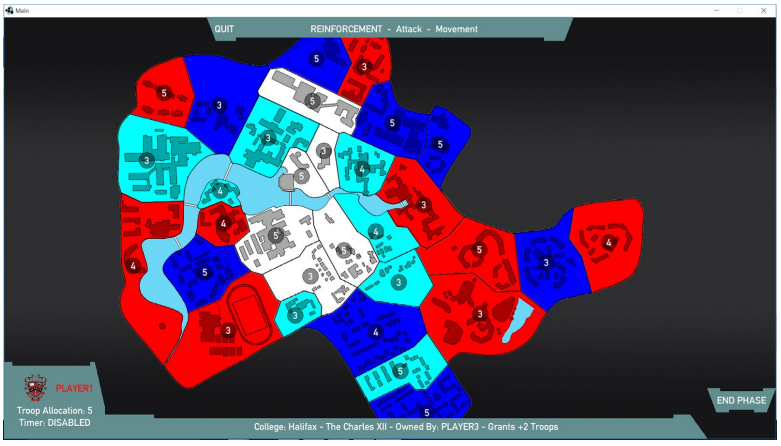
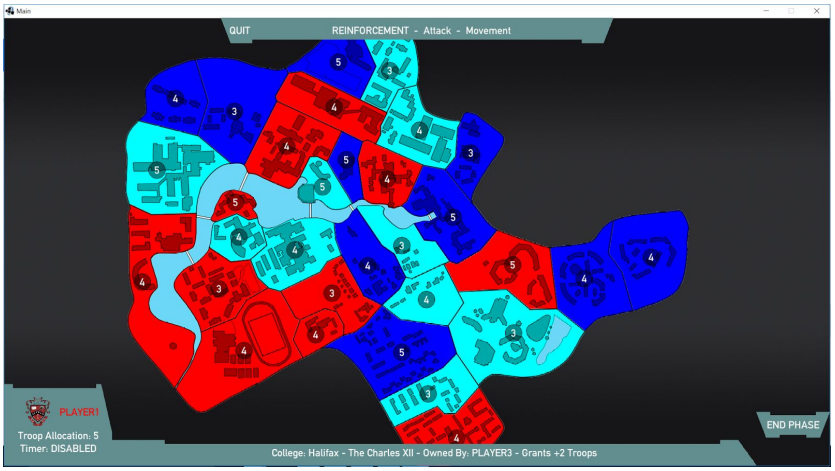
		screen with a call to generate an escape button		

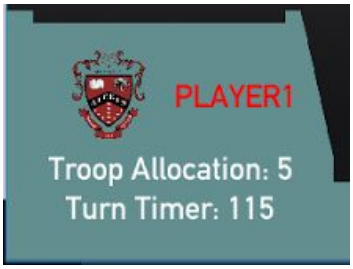
GameSetupScreen class

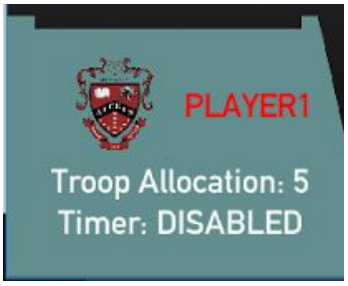
ID	Method	Expected Outcome	Actual Outcome	Pass/Fail
1	setupUI	Pressing Esc should take the user back to the menu screen	Pressing Esc returned the user to the menu screen	Pass

2	togglePlayerType	All 4 player type labels should cycle between “Human Player” and “None” when the corresponding arrow is pressed	All 4 player type labels cycled between “Human Player” and “None” when the corresponding arrow is pressed	Pass

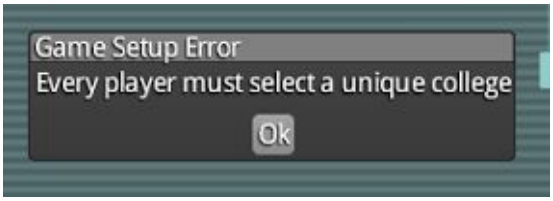


3	toggleCollegeType	All 4 college labels should cycle between the implemented colleges when the corresponding arrow is pressed	All 4 college labels cycled between the implemented colleges when the corresponding arrow is pressed	Pass
4	setupSwitchTable	The NeutralAI should be toggleable by clicking the appropriate switch	The NeutralAI was toggleable	Pass
 				
5	setupSwitchTable	The turn timer should be toggleable by clicking the appropriate switch	The turn timer was toggleable	Pass

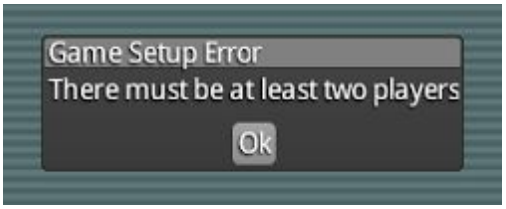




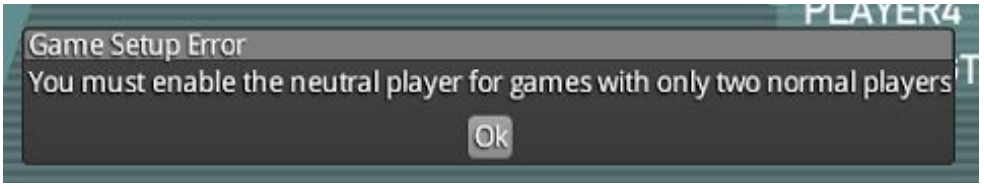
6	validateCollegeSelection	All players should have selected a unique college, otherwise a dialogBox stating this will be displayed and the game will not start	When duplicate colleges were selected, the correct dialogBox was shown and the game did not start	Pass

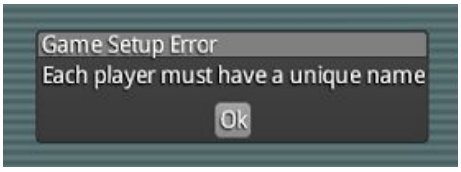
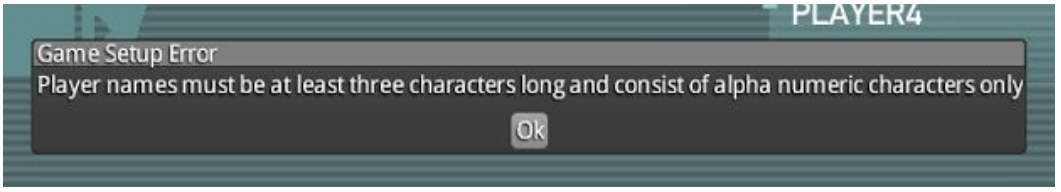


7	validatePlayerConfiguration	If there are less than 2 “Human Players” a dialogBox should be displayed stating there must be at least 2 players	When less than 2 “Human Players” were configured the correct dialogBox was shown	Pass



8	validatePlayerConfiguration	If there is only 2 players, the NeutralAI must be enabled, otherwise, a dialogBox will be displayed stating this	When only 2 players were configured and the NeutralAI was not enabled the correct dialogBox was displayed	Pass



9	validatePlayerNames	If any player names contain a special character, or any of the player names are not unique, then a dialogBox should be displayed stating that the player names are either invalid or there are duplicate names	The correct dialogBox was displayed as expected	Pass
 				
10	startGame	This should successfully start a game if all validations are passed	The game started when valid parameters were provided, however the map is sometimes drawn off centre when the game starts	Fail

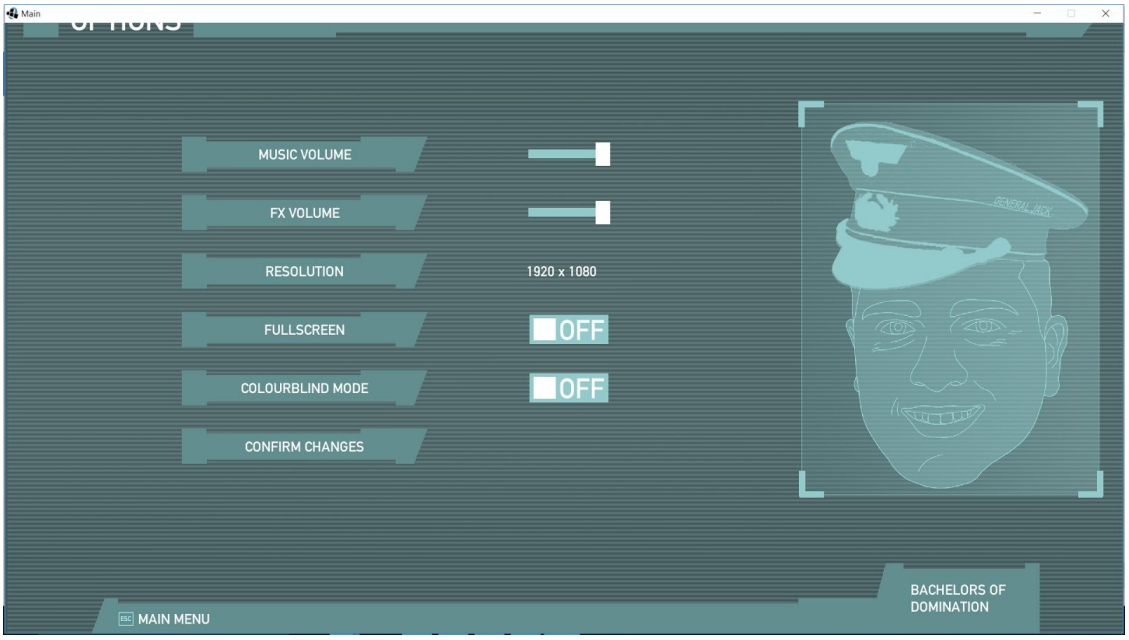
MenuScreen class

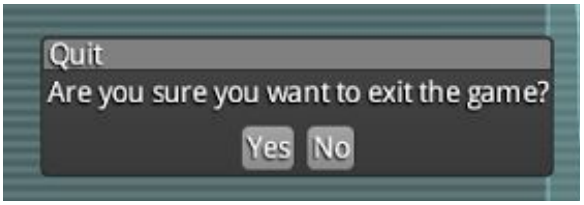
ID	Method	Expected Outcome	Actual Outcome	Pass/Fail
1	setupUI	On launch the menu screen should load containing all relevant buttons, header, footer	On launch the menu screen loaded containing all relevant buttons, header, footer and	Pass

		and images	images	
2	setupMenuTable	Clicking “Start New Game” should display the game setup screen	Clicking “Start New Game” displayed the game setup screen	Pass

3	setupMenuTable	Clicking “Load Game” should have no effect as this has not yet been implemented	Nothing happened	Pass

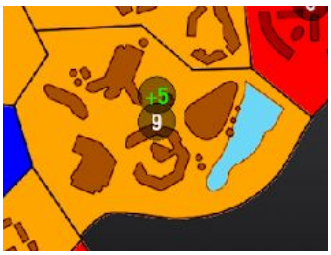
4	setupMenuTable	Clicking “Options” should display the options menu	Clicking “Options” displayed the options menu	Pass

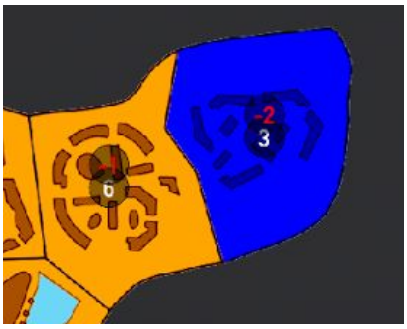


5	setupMenuTable	Clicking “QUIT” and pressing Esc should display a dialogBox asking the user to confirm they want to quit	Clicking “QUIT” and pressing Esc displayed a dialogBox asking the user to confirm they want to quit	Pass
				

UnitChangeParticle class

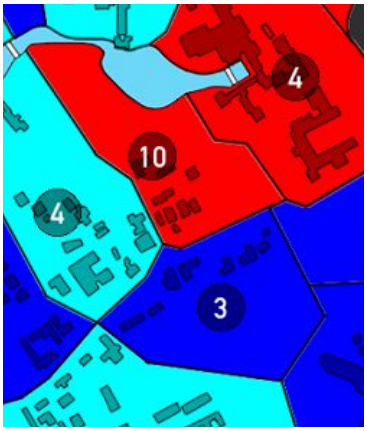
ID	Method	Expected Outcome	Actual Outcome	Pass/Fail
1	UnitChangeParticle	Initialises the UnitChangeParticle, setting the texture and base information required	Glyph is drawn correctly	Pass
2	toDelete	Deletes the glyph after it has been displayed for 1 second	The glyph was deleted after 1 second	Pass
3	draw	The glyph should be drawn and animated from the location at which it is draw and animated from. This should be the same position as the current troops glyph	The glyph was drawn and animated correctly on every instance tested	Pass





Sector class

ID	Method	Expected Outcome	Actual Outcome	Pass/Fail
1	changeSecorColor	When starting the game all sectors should be coloured from white to the appropriate colour based on which payer owns that tile. Then when taken over by another player the tile should again be coloured.	All sectors were coloured correctly	Pass



4
1
4
7