

Appendix E: Unit Tests

Below is a table of the unit tests included within this project.

The unit tests are associated with an appropriate requirement to allow for traceability, and the tests aim to check that the code works for any associated requirements. Not all requirements have associated tests, and vice versa - this is because some requirements cannot be explicitly unit tested, and some tests do not link up directly to a requirement, but are still needed to ensure the code functions as intended.

There is a criticality measure against each test, for both acceptance and unit tests - this is to represent how important the test is to the overall function of the code. Criticality is on a scale - high criticality means that if that test fails, the project will not function at all; low criticality means that if the test fails, the project will still mostly function as intended.

ID	Test Name	Purpose	Criticality	Class	Req ID	Result
1.1	constructor	Verifies GUI is constructed correctly	High	GUIController_Test	1.1.4	Passed
1.2	screenCanBeSet	Verifies that a screen from the game can be set as the current screen	Medium	GUIController_Test	1.1.4	Passed
1.3	screenCanBeChanged	Verifies screen can be changed to a different screen	Medium	GUIController_Test	1.1.4	Passed
1.4	getTime	Verifies the game returns the correct time value	Medium	GameSnapshot_Test	6.1.2	Passed
1.5	getRooms	Verifies the game returns the correct list of rooms	High	GameSnapshot_Test	4.1.1	Passed
1.6	getClues	Verifies the game returns the correct list of clues	High	GameSnapshot_Test	5	Passed
1.7	getClue	Verifies that the list held by the journal contains clues in the right format	Low	Journal_Test	5.2.1	Passed
1.8	addClue	Verifies that clues are added to the journal properly	Medium	Journal_Test	5.2.1	Passed
1.9	addConversation	Verifies that conversation snippets are added in the correct format	Low	Journal_Test	7	Passed
1.10	getConversation	Verifies that getConversation returns all the conversation snippets that it is currently holding	High	Journal_Test	7	Passed
1.11	SetRoomGetRoom	Verifies that a room, once set, is returned in the correct format.	Medium	MapEntity_Test	4	Passed
1.12	getName	Verifies that a map entity returns its name correctly	Low	MapEntity_Test	-	Passed

1.13	genDescription	Verifies that a map entity returns its description correctly	Low	MapEntity_Test	-	Passed
1.14	getTexture	Verifies that a map entity sets its texture correctly	Medium	MapEntity_Test	-	Passed
1.15	setSpeech	Verifies that the narrator screen text can be set correctly	Low	NarratorScreen_Test	7.1.7	Passed
1.16	updateSpeech	Verifies that the narrator screen text can be updated correctly	Low	NarratorScreen_Test	7.1.7	Passed
1.17	getTransition	Verifies that the player transitions between rooms	High	Room_Test	2.1.4	Passed
1.18	addTransition	Verifies that adding new transitions between rooms is correctly implemented	High	Room_Test	2.1.4	Passed
1.19	walkable	Verifies that walkable tiles are walkable and tiles that aren't walkable aren't walkable	High	Room_Test	-	Passed
1.20	trigger	Verifies that a trigger tile is triggerable, and one that isn't triggerable isn't triggerable	High	Room_Test	-	Passed
1.21	matRotation	Verifies that doormats are orientated the correct way	Low	Room_Test	2.1.4	Passed
1.22	distributeRoomsGiveClues	Verifies that the distributeClues() method distributes clues so that there is at minimum 1 clue per room	Medium	ScenarioBuilder_Test	5.1.1	Passed
1.23	distributeCluesDiffRooms	Verifies that the clues aren't being given to the same room	Medium	ScenarioBuilder_Test	5.1.1	Passed
1.24	generateMotives	Verifies that the motive clue is split into 3 equal parts	Low	ScenarioBuilder_Test	5.1.3	Passed
1.25	modifyScoreAddition	Verifies that the correct score is added to the current score	Low	Scoring_Test	6	Passed
1.26	modifyScoreSubtraction	Verifies that the correct score is subtracted from the current score	Low	Scoring_Test	6	Passed
1.27	updateScoreNoDecrease	Verifies that the score doesn't change if the time passing is less than 5 seconds	Low	Scoring_Test	6.1.2	Passed
1.28	updateScoreHasDecrease	Verifies that the the score decreases by 1 after 5 seconds	Low	Scoring_Test	6.1.2	Passed
1.29	getInfo	Verifies that all all parts of the clue are stored corretly	High	Clue_Test	5	Passed
1.30	isMotive	Verifies that motive clues return true as a motive clue and false if they are not motive clues	High	Clue_Test	5.1.3	Passed
1.31	isMeans	Verifies that means clues return	High	Clue_Test	5	Passed

		true as a motive clue and false if they are not means clues.				
1.32	constructorValidationFail	Verifies that given a false .JSON file, the game encounters “JSON not being verified” error.	Low	Dialogue_Test	7	Passed
1.33	constructor2ValidationPass	Verifies that given a valid .JSON file no error is thrown.	High	Dialogue_Test	7	Passed
1.34	getUsingClue	Verifies that when given a clue, the dialogue getter returns a valid response.	High	Dialogue_Test	7	Passed
1.35	getUsingString	Verifies that when given a string, the dialogue getter returns a valid response.	High	Dialogue_Test	7	Passed
1.36	aStar	Verifies that the A-Star algorithm produces the correct result.	Low	Player_Test	2	Passed
1.37	testPlayerName	Verifies the player name is returned correctly	Low	Player_Test	-	Passed
1.38	init	Verifies that the NPCs holds data about themselves correctly	High	Suspect_Test	3	Passed
1.39	hasBeenAccused	Verifies that once a NPC who isn't the murderer has been accused, the NPC records that they have been accused.	Low	Suspect_Test	7.1.9	Passed
1.40	setPosition	Verifies that the position of a NPC can be successfully set.	High	Suspect_Test	3	Passed
1.41	setKiller	Verifies that the setKiller method successfully sets the NPC as the killer.	High	Suspect_Test	3.1.5	Passed
1.42	setVictim	Verifies that the setKiller method successfully sets the NPC as the victim.	High	Suspect_Test	3.1.5	Passed