

UNIVERSITY OF YORK  
DEPARTMENT OF COMPUTER SCIENCE

# Software Testing Report Cohort 2 - Group 16 (Skloch)

## Group Members:

Charlotte MacDonald  
Hollie Shackley  
Luis Benito  
Kaustav Das  
Sam Hartley  
Owen Gilmore

## A) Testing Method and Approaches

Our team adopted a structured approach to ensure that every component of our project was thoroughly evaluated against its user and functional requirements. This strategy is aligned with the traditional Testing Pyramid, involving a combination of automated and manual testing techniques which goes as follows:

**Unit Testing:** We focused primarily on unit tests ensuring each module's functionality was independently verified, using automated tests to rapidly assess the smallest units of code. This both helped in detecting issues at an early stage and also enhanced our ability to perform regression testing efficiently. Unit tests were used as they are fundamental for a robust testing strategy, allowing issues to be identified and resolved without the overhead of more complex setups. This layer was heavily automated, enhancing our test execution speed and frequency.

**Integration Testing:** These tests evaluated the interactions between integrated units to expose faults in their interfaces. It was crucial for ensuring that independently tested modules work together as expected. Integration tests were used as these are critical for ensuring that combined parts of the application function together, which cannot be captured by unit tests alone.

**End-to-End Testing:** Conducted as part of our system testing phase, these tests validated the entire application's flow from end to end. Although limited due to their expensive nature in terms of time and resources, they were vital for confirming the system met our development goals and user requirements. While end-to-end tests covered a smaller portion of our testing effort, they provided a crucial check on the real-world use of the system, ensuring that the entire application functions correctly from the user's point of view.

Wherever feasible, tests were automated to speed up the testing cycles and enable continuous integration practices. Manual testing was used selectively for areas that are less easy or impossible to automate, such as UI/UX aspects and scenarios with complex interaction. Both automated and manual tests included negative testing to ensure the system can handle invalid input or unexpected user behaviour in a controlled manner as well as boundary testing and exception handling in order to focus on edge cases and ensure system stability under unusual conditions.

We aimed for comprehensive line and branch coverage to ensure a wide range of code paths are tested.

All tests are linked to specific requirements via a traceability matrix, ensuring each requirement is covered by one or more tests. This matrix also helps with impact analysis when requirements change.

Given the limited scope for end-to-end tests, focus was placed on areas that were critical to the development process or had undergone significant changes since assessment 1. This approach ensured that our testing resources were allocated efficiently, and addressed the most important areas of the application.

As well as unit testing being implemented as the code was written, informal manual regression testing was carried out after all major changes. These mostly consisted of team members, especially those not working on implementation themselves, playing the game and testing the new features in the process. This enabled us to identify issues as they came up rather than finding anything at the last minute which we might not have had time to fix.

B)

## Test Report

The tests conducted for this project include automated unit tests designed to verify the functionality of various components within the game. This testing suite was run to ensure the correctness, completeness and reliability of the system. Automated testing was preferred over manual testing to ensure repeatability and efficiency.

### Statistics of Test Runs and Results Achieved

The tests were organised into several categories, each focusing on different aspects of the game. The following table provides an overview of the tests and their results:

Test Name	Total Tests	Pass	Fail	Comment	Actionable Step	Requirement fulfilled
Achievement Tests	2	2	0	The achievement system is functioning correctly. The tests confirm that achievements are awarded appropriately on the specified criterias.	N/A	UR_STREAKS
Activity Tests	6	6	0	Failed Test: <i>testActivityCompletion</i>  Reason for failure: Incorrect assumption about the activity duration calculation. (The calculation of activity durations have inconsistencies that need addressing)	Review and refactor the logic for calculating durations to ensure accurate tracking.	UR_ENERGY, FR_ACTIVITY_ENERGY
Asset Tests	18	18	0	All required assets are loaded correctly, and the integrity checks confirm that assets are not missing or corrupted.	N/A	UR_SOUND, UR_MAP, UR_ADDITIONAL_MAP, UR_TUTORIAL, UR_CREDITS, FR_CREDITS
Clock Tests	11	11	0	The game clock is functioning as expected, with correct handling of time related events and transitions.	N/A	UR_TIME
DialogueManager Tests	9	9	0	All Dialogues and NPC interactions are working as intended	N/A	UR_INTERACT, FR_INTERACT_CONTROLS, FR_INTERACT_DISTANCE
Keyboard Input Handler Tests	5	5	0	The logic for keyboard inputs of a wide variety of necessary movements and interactions are working as intended	N/A	UR_MOVEMENT, UR_INTERACT, FR_MENU_NAVIGATION, FR_PAUSE_CONTROLS
Leaderboard Manager Tests	2	2	0	The leaderboard management system correctly tracks and updates the player scores, which leads to accurate rankings	N/A	UR_LEADERBOARD
Player Tests	15	15	0	Player tests confirmed that the movements in all 8 directions are working properly alongside collision effect of the player and the animations	N/A	UR_MOVEMENT, UR_ANIMATION, FR_PAUSE_FREEZE
State Tests	9	9	0	Confirmed that all the state tests are running as intended, which includes how the trigger mechanism for interacting with NPCs and objects, time passed and the variety of activities are	N/A	UR_TIME, UR_INTERACT, GAME_END, FR_INTERACT_CONTROLS,

				running smoothly.		FR_INTERACT_DISTANCE, FR_ACTIVITY_ENERGY_REQ, FR_TIME
--	--	--	--	-------------------	--	----------------------------------------------------------

## Completeness and Correctness of Tests,

The automated tests provide a reliable and repeatable way to verify system behaviour. While most tests passed, the identified issues highlight areas for improvement:

- **Coverage:** The tests cover all critical functionalities, ensuring all the major components are verified.
- **Traceability:** Each test is mapped to specific requirements, ensuring comprehensive coverage of the system functionality.
- **Correctness:** The high pass rate indicates that the system meets its design specifications. The identified (controlled) failures are isolated cases that can be addressed with targeted fixes.

All in all, our testing framework is robust and provides confidence in our system's stability. The few failures identified does not undermine the overall functionality of the system, but highlights the specific areas for improvement, which will be addressed to ensure the system's reliability.

c)

### Coverage Report:

<https://samh366.github.io/16-2-website/assets/deliverables/group2/Coverage%20Report/index.html>

### Test Results:

<https://samh366.github.io/16-2-website/assets/deliverables/group2/Test%20Results.html>

## Manual Tests

Requirement	Instructions	Pass/Fail	Actionable Step	Reason for being manual
UR_MAP	Load the game. Select the "Start Game" button on the menu screen. Walk around the map and consider if it resembles Heslington.	Pass	N/A	There is no simple way to automate testing if the map resembles Heslington so this will be up for the tester to determine.
UR_COMPATABILITY	With the Jar (Java ARchive) file, go onto your desktop and try to run the game.	Pass	N/A	
UR_GRAPHICS	Load the game. Play while considering if the graphics have positive/happy vibes.	Pass	N/A	"Positive/happy vibes" is a subjective requirement which will require the tester's own judgement.
UR_ACCESSIBLE, FR_ACCESSIBLE	Load the game. Are there any objects that are only distinguishable with colour? If so, this manual test fails.	Pass	N/A	There is no easy way to test how accessible the game is without in-person testing.
UR_MENU, FR_MENU_NAVIGATION, CONTROLS.	Load the game. Click one of the buttons and consider if it takes you to the correct page. Test for all buttons.	Pass	N/A	MenuScreen uses listeners on the buttons which are difficult to automate in the time constraints.

FR_MENU_SELECT_CONTROLS				
FR_START_GAME	Load the game. Click the Play game button. You should be taken to a tutorial screen with a Continue button. Click this button and you should be taken to a screen to select avatars. When selected, you should be taken to the Play screen.	Pass	N/A	These screens use listeners on the buttons which are difficult to automate in the time constraints.
UR_OBJECTS	Move the character around and see if there are objects such as trees, buildings, cars etc that the player's character can not walk over.	Pass	N/A	Only a human/object recognition AI could check if there are objects within the game.
UR_GAME_END	Play through all 7 days of the week and check to see if there is a game over screen with a score breakdown	Pass	N/A	Only a human can check to see if there is a score breakdown displayed once completing the game.
UR_SCORE, FR_SCORING, FR_SCORING_GAME_OVER, FR_SCORE_CALC	Play the game all the way through, check the game over screen for a final score. You shouldn't be able to see your score until the game is over.	Pass	N/A	Too complicated to automate testing given time constraints.
UR_TUTORIAL	Start a new game check to see if there is a tutorial giving instructions on how to play the game including controls and objectives.	Pass	N/A	Only a human can check to see if there is a comprehensive tutorial, this test is subjective.
UR_ADDITIONAL_MAP, FR_ADDITIONAL_MAP	When playing the game, interact with the bus stop to see if the map switches	Pass	N/A	Only a human can see if the map visibly changes to the correct map.
UR_LEADERBOARD, FR_LEADERBOARD	Once completed the game, add your score to the leaderboard, then exit to main menu to check to see if score was correctly added to the leaderboard (given that the score was high enough to make top 10)	Pass	N/A	Only a human can play the game and navigate the menu to access the leaderboard.
UR_STREAKS, FR_STREAKS	When playing the game, feed the ducks 6 times, go to the gym 3 times and walk at least 200 steps	Pass	N/A	Extremely difficult to automate this task, as an AI agent would be needed.
UR_AVATARS, FR_AVATARS	Upon starting the game check to see if you can select from a masculine or feminine avatar. Then check that the avatar chosen is the one playable in the game.	Pass	N/A	Extremely difficult to automate this task, as an AI agent would be needed.
UR_CREDITS, FR_CREDITS	Go to the main menu and click on the credits button, a credits screen should appear, providing the creators, assets etc.	Pass	N/A	Extremely difficult to automate this task, as an AI agent would be needed.
UR_MAP_OVERVIEW	When playing the game click the M key to toggle between the regular view and the map overview, the views should change as appropriate.	Pass	N/A	Needs a human to click the key and visually check if the map view actually changes
UR_SOUND	With the sound turned on within the game and the system sound on and a working audio output device listen for music or appropriate sound effects	Pass	N/A	Needs a human to listen and positive music is subjective.

FR_MAP_B US_STOP	When playing the game, move to the north-west area of the map where the bus-stand is and press E to interact. Use the arrow keys to select yes to head to town. If the map changes to town, that indicates the bus-stop is operating properly	Pass	N/A	Extremely difficult to automate this task, as an AI agent would be needed
FR_INTERA CT_PROMPT	When playing the game, position your player close to all of the interactable items, and if the option to interact pops up then it passed this manual test	Pass	N/A	Extremely difficult to automate this task, as an AI agent would be needed
FR_PLAY_S PORT	When playing the game, move to the north-east area of the map where the gym is situated and press E to interact. Use the arrow keys to select yes to use the gym. If the energy level goes down and 2 hours pass then it passed this manual test	Pass	N/A	Extremely difficult to automate this task, as an AI agent would be needed
FR_FEED_ DUCKS	When playing the game, move to the south-west area of the map where the ducks are and press E to interact. Use the arrow keys to select yes to feed the ducks. If the energy level goes down and 1 hour passes then it passed this manual test.	Pass	N/A	Extremely difficult to automate this task, as an AI agent would be needed
FR_STUDY	When playing the game, move to the Ron Cooke Hub or the Comp-Sci building and press E to interact. Use the arrow keys to select yes to study. If the energy goes down and 2 hours pass, then the game passed this manual test.	Pass	N/A	Extremely difficult to automate this task, as an AI agent would be needed
FR_DOUBL E_STUDY	Using the manual test for the requirement FR_STUDY and study twice a day, then sleep and attempt studying twice on separate day. If studying twice on more than one day is possible, then manual test has failed, else pass.	Pass	N/A	Extremely difficult to automate this task, as an AI agent would be needed
FR_EAT	Move to the Piazza building and press E to interact. Use the arrow keys to select yes to eat. If the energy goes down and 1 hour passes, then the game passed this manual test.	Pass	N/A	Extremely difficult to automate this task, as an AI agent would be needed
FR_SLEEP	Move to the student accommodation buildings (Constantine and Anne Lister) and press E to interact. Use the arrow keys to select yes to sleep. If the energy refills to 100%, time resets to the start of day (8am) and day increments by 1, then the game passed this manual test.	Pass	N/A	Extremely difficult to automate this task, as an AI agent would be needed
FR_TIME_L IMIT	Interact with the Activities at night, and if it is possible to do any of those activities at night then manual test has failed, else pass.	Pass	N/A	Extremely difficult to automate this task, as an AI agent would be needed
FR_STREA K_GYM_BR O	Finish the game up until the 7th day and make sure to use the gym at least 3 times to get this achievement. If Achievement "Gym Bro" is shown on the right side of the game over screen, then the game passes this manual test.	Pass	N/A	Extremely difficult to automate this task, as an AI agent would be needed

FR_STREAK_LOTD	Finish the game up until the 7th day and make sure to feed the ducks at least 6 times to get this achievement. If Achievement "Duck duck go" is shown on the right side of the game over screen, then the game passed this manual test.	Pass	N/A	Extremely difficult to automate this task, as an AI agent would be needed
FR_STREAK_JOGGER	Finish the game up until the 7th day and make sure to walk at least 200 steps each day to get this achievement. If Achievement "Jogger" is shown on the right side of the game over screen, then the game passes this manual test.	Pass	N/A	Extremely difficult to automate this task, as an AI agent would be needed
FR_MAP_OVERVIEW_CONTROLS	When playing the game, press the M key to toggle between a regular perspective and a zoomed out view of the map, with labels for the activities. If this toggle does indeed occur and the two views are both correctly displayed, the test passes.	Pass	N/A	This test is based on the visuals displayed on-screen and thus requires human verification.
FR_MAP_OVERVIEW_MOVEMENT	When playing the game, press the M key to toggle between a regular perspective and a zoomed out view of the map, with labels for the activities. If you are unable to move your character or interact with objects while the map view is displayed, this test passes.	Pass	N/A	This test is based on the visuals displayed on-screen and thus requires human verification.
UR_ANIMATION	When playing the game, verify that there are some simple animations being played correctly, e.g. the player's walking animation. If these animations are working, the test passes.	Pass	N/A	This test is based on the visuals displayed on-screen and thus requires human verification.
UR_OBJECTS	Load the game. Press the Start Game button in the Menu Screen. Continue at the tutorial and select your avatar. Wander through the map, NPCs, buildings and other objects should be visible. If they are, this test is passed	Pass	N/A	Can't test objects in the map as the map has been made in Tiled including the objects and buildings.
UR_ENERGY, FR_ENERGY	Load the game. Press the Start Game button in the Menu screen. Continue at the tutorial and select your avatar. You should be able to see an energy bar on the screen. Go to complete an activity by pressing e when close to an interactable object. When completing an activity that uses energy, you should see the energy bar decrease. If so, this test passes.	Pass	N/A	Difficult to test rendering on the screen and involves a lot of different methods to test.
FR_TIME	Load the game and press the start game button on the menu screen. Continue at the tutorial and select your avatar. The time and day should be displayed towards the top of the screen. Play the game for a few minutes. If the time has not increased, this test shall fail. Do a sleep activity by navigating to accommodation and interacting. The day should now say 2. If not, this test has failed.	Pass	N/A	Requires on-screen rendering so better to test manually to ensure it appears.
FR_TUTORIAL	Load the game and press the start game button on the menu screen. This should take you through to a tutorial screen. Read the text, if it informs you about in-game controls and the objective, this test passes.	Pass	N/A	Text in the tutorial cannot be automated. The tester must read it and evaluate whether it is legible and descriptive enough.

UR_PAUSE, FR_PAUSE	Start the game and press the escape key to open the pause menu. If the game state pauses and the menu shows, then this manual test is passed.	Pass	N/A	Only a human can verify that the game state hasn't continued to advance while the game is paused.
-----------------------	-----------------------------------------------------------------------------------------------------------------------------------------------	------	-----	---------------------------------------------------------------------------------------------------