ENG1 Software Testing

Cohort 3 - Group 28

“Team 28”

Muhammed Salahudheen
Joel McBride
Jamie Rogers
Maciek Zaweracz
Rhys Yeaxlee
Alex Spencer
Alex Firth

Summary

Software testing is an essential part of any software engineering project, but the circumstances of creating a game force a certain approach. All tests need to be based on the requirements. At minimum every requirement listed “shall” must be tested for, ideally the tests cover every single requirement. Unit testing is a highly effective method of testing code functionality due to its speed and objectivity. In games, the high degree of interactivity limits how much can be tested with Unit Tests. A manual test plan must fill in the gaps, which must have specific inputs and expected outputs to ensure reliability. If a unit or manual test fails, we must alter the tested code to ensure adherence to the requirements the tests are based on.

The correspondence between tests and requirements can be found on a table on the website.

Test

Unit Tests

AssetsTests exists to check for image and sound files that are essential for users to understand the game environment. The assets act as affordances that help user navigate the User Interface, so their existence is essential. These tests are primarily for future proofing, in case assets are replaced these tests make sure that the files that use the assets actually have the correct path.

PersistenceTest simply checks for the persistent data used to store high scores.

PlayerTests has one test to ensure the position can be changed. The other tests ensure that interactable objects have the expected bounds.

LeaderboardsTest tests the functionality of the leaderboard, testing whether it can successfully be reset and written to, and whether it can ensure the scores are presented in order. The last test makes sure that scores that would rank 11th are not saved to the leaderboard.

All the above tests completely succeeded. In total 21 tests were ran and 21 tests passed. Screenshots showing the units tests being passed can be found in a zip file on our website.

Manual Tests

Test ID	What the Test Data is	Where the Test Data will be used	What the expected result is	What the purpose of the test is
1	Clicking the interact button after approaching the dock	Main Game	Duck feeding shall engage, fun will increment, time shall increment	UR_INTERACT UR_RECREATE UR_TIME_SKIP
2	Clicking the interact button after approaching the sport building	Main Game	Sport recreation shall engage, fun will go up, time shall increment	UR_INTERACT UR_RECREATE UR_TIME_SKIP
3	Clicking the eat button after approaching the piazza building	Main Game	Feeding shall engage, food will increment, time shall increment	UR_INTERACT UR_EAT
4	Clicking the study button after approaching the piazza building	Main Game	Studying shall engage, study shall increment, time shall increment	UR_INTERACT UR_STUDY UR_STUDY_GAME UR_TIME_SKIP
5	Clicking the study button after approaching the compsci building	Main Game	Studying shall engage, study shall incremenmt, time shall increment	UR_INTERACT UR_STUDY UR_STUDY_GAME UR_TIME_SKIP
6	Holding up	Main Game	Character moves up	UR_MOVEMENT
7	Holding left	Main Game	Character moves left	UR_MOVEMENT
8	Holding down	Main Game	Character moves down	UR_MOVEMENT
9	Holding right	Main Game	Character moves right	UR_MOVEMENT
10	Clicking the sleep early button after approaching the Goodricke building between 20:00 and 23:00	Main Game	Sleep engaged, next day progressed	UR_INTERACT UR_SLEEP UR_TIME_SKIP

11	Walk around	Main Game	Map is made visible	UR_MOVEMENT UR_MAP_DESIGN UR_CAMPUS_BUILDINGS
12	Wait for name entry screen	Main Game	Name entry screen appears	UR_USERNAME UR_GAME_LENGTH
13	Enter name into leaderboard	Name entry screen	Leaderboard accepts the name	UR_LEADERBOARD UR_USERNAME
14	Every day walk to nisa and perform recreation, then view results screen after entering name	Leaderboard screen	Bonus points are scored	UR_STREAKS
15	Every day walk to piazza and perform eating, then view results screen after entering name	Leaderboard screen	Bonus points are scored	UR_STREAKS
16	Every day walk to compsci and perform studying, then view results screen after entering name	Leaderboard screen	Bonus points are scored	UR_STREAKS
17	Input the correct number codes during the study game	Study game	Success message displayed	UR_STUDY_GAME
18	Input the incorrect number code during the study game	Study game	Failure message displayed	UR_STUDY_GAME
19	Play for 5-6 minutes	Main Game	The game will end	UR_GAME_LENGTH
20	Clicking the interact button after approaching Nisa	Main Game	Time will pass and the fun meter will go increment	UR_INTERACT UR_RECREATE UR_TIME_SKIP
21	Achieve a high score, then go to the leaderboard screen to see your name at the top	Leaderboard Screen Name screen	The name and score will appear	UR_LEADERBOARD UR_HI_SCORE UR_USERNAME
22	Click on controls	Controls screen	The scenario and controls will be presented	UR_CONTROLS UR_OBJECTIVE
23	Enter nothing into the name entry screen	Name entry screen	Leaderboard rejects and requests re-entry	UR_LEADERBOARD UR_USERNAME
24	Get a score of 0	Main Game End screen	The end screen says I did not pass.	UR_GAME_OVER
25	Get a score below 40	Main Game End screen	The end screen says I did not pass.	UR_GAME_OVER
26	Get a score above 40	Main Game End screen	The end screen says I did pass.	UR_PLAYER_SCORE
27	Do an interaction that skips time (interacting with NISA)	Main game	An animation will play	UR_TIME_SKIP_ANIMATION
28	Start game by pressing the play button	Main menu	A backstory will be described before transition to main game	UR_BACKSTORY
29	Wait for one in-game day	Main Game	It will take around a minute	UR_TIME_SCALE
30	Play for one in-game day (Go to NISA for two in-game hours, study at CompSci for two in-game hours)	Main Game	It will take around a minute	UR_TIME_SCALE

The above table describes every manual test, including the inputs, where the inputs would be performed, the expected results and what requirements they address. It should be noted that all but test 19 had unusual game speeds applied to reduce the time spent on each test. We believe there is little room for that alteration to corrupt the results.

Tests 27 and 28 fail due to the feature they test being unimplemented due to their low priority and time restrictions. Tests 29 and 30 represent how UR_TIME_SCALE conflicts with UR_TIME_SKIP and UR_GAME_LENGTH. A game with time skipping mechanics cannot have consistent day lengths.
Out of all the tests of implemented mechanics, only 15 and 19 had failures. These failures were patched and the subsequent tests succeeded. In total out of 30 manual tests, 25 succeeded on first run, 2 succeeded on second run and 3 failed.
The testing table and relevant videos can be found on our website.

Links

All aforementioned resources can be found at https://eng1team28.github.io/#Test2

- https://eng1team28.github.io/test/ (report output)
- https://eng1team28.github.io/extra/ManualTestPlan.pdf
- https://eng1team28.github.io/extra/Requirements-To-Tests%20Table.pdf
- https://www.youtube.com/watch?v=8wfPOAMYI-s (manual test video)