SE 3XA3: Test Report
Chrome Dino Runner

Team 1#, Team Rex
Anjola Adewale and adewaa1
Chelsea Maramot and maramotc
Sheridan Fong and fongs7

April 13, 2022

Contents

List of Tables

List of Figures

Table 1: **Revision History**

Date	**Version**	**Notes**
February 4, 2022	0	All team members
April 10, 2022	1.0	Sheridan Fong (updated functional tests)
April 10, 2022	1.1	Anjola Adewale (updated non-functional tests)
April 11, 2022	1.2	Chelsea Maramot (updated matrix and code coverage)

1 Functional Requirements Evaluation

These tests ensure that the user is able to use the software according to the Software Requirement Specification (SRS). The following tests include the checking the functionalities of the Instructions, Leaderboard, Game Settings, and Game Track.

1.1 E1: Viewing the Instructions Page

The Instructions Page is fully functional. The user is able to access it by pressing the “How to play” button in the Menu Page. The keyboard bindings are displayed in a comprehensive manner. The system then displays the Instructions page satisfying FR1. Finally, the user is able to transition back to the Menu Page from the Instructions Page satisfying FR2.

1.2 E2: Viewing the Leaderboard

The Leaderboard Page is fully functional. The user is able to access the Leaderboard Page from the Main Page by clicking the text. The system displays the top five scores of all users. This is different than FR3 as FR3 displays the top scores of the specific user. The leaderboard implemented does not filter the high scores by user and therefore FR5 is not met. The user can navigate back to the menu by pressing ’b’ and therefore this satisfies FR4.

1.3 E3: Change Game Settings

The Game Settings Page meets the functional requirements stated in the SRS. The user is able to transition from the Game Settings Page to the Menu Page and vice versa, as stated in FR6. Moreover, the game audio can be manipulated by the user within the Game Settings Page, as stated in FR7. Finally, FR8 states that themes must be displayed and the user is able to select a theme. This functional requirement has been met.

1.4 E4: Playing the Game

The game functions as proposed, the game displays the correct theme based off user input, the initial score is 0 and sound is outputted based on settings, this satisfies FR9, FR10, and FR11 respectively. The system starts gameplay when the user inputs any key not just WASD keys therefore FR12 fails. The game is controlled by WASD keys and arrow keys therefore satisfying FR13. The game awards points when the user moves and display's the current score to the screen, therefore FR14 and FR15 are satisfied. FR17 is satisfied as the system allows the user to replay the game and they can stop the game by pressing 'p' and return to the main menu. The user can only enter their username on the Main Page therefore FR18 fails. After game play the score is entered into the database.

2 Nonfunctional Requirements Evaluation

2.1 Look and Feel

LF1: The user interface consists of essential components relevant to the game such as the obstacles, and characters LF2,LF3: Although we implemented two new themes, we left the original version of the game as a theme option. Therefore the game maintains the 90's arcade appearance. LF4: The game is fully implemented in black and whit, therefore the game colours do not distract the user's game play

2.2 Usability

UH1: The game was implemented to be controlled using two methods. The arrow keys and "WASD" keys UH2: Through survey testing we verified that the game has a simple main menu page which leads to the different game settings UH3. Through survey testing we verified that the game was partially intuitive. Some users found that they had to perform "jumping" and "ducking" actions earlier than anticipated in order to do die in the game. UH4: The game instructions and settings are implemented in English UH5: The theme of the game is easily adjustable. All a user has to do is navigate to the game page and select the theme they want. UH6: User's that participated in the test group did not undergo any prior training before playing the game, thus this requirement is achieved UH7: The instructions in the main menu are easy to understand as verified by survey testing UH8:The game is delivered as an executable file, therefore there is a level of abstraction from the user U9,UH10: These requirement were fully implemented, the game used common controls like the space-bar, arrow keys and “WASD” keys to control the character. UH11,UH12: Since the game a was fully implemented in black and white with large fonts, all accessibility requirements are satisfied

2.3 Performance

PE1: The scores are uploaded to the leaderboard in less than 10 seconds after game play. This was verified by using the time library in our code. PE2: The interface has a response time of less than 2 seconds. This was verified manually and through using the time library. PE3: The game updates the new status parameters such as game theme and audio settings within 5 seconds. This was verified manually. PE4: This functional requirement was not specific. Integer whole numbers for the game score were updated within 900 ms of changing. This was verified manually. PE5: The leaderboard is updated according to the top integer score, this was verified using manual and unit testing. PE6: The maximum latency is less then 900 ms when navigating between pages. This was verified by looking at the flags and when they changed using the time library. PE7: The game is only available to a single player at a time on a local device. However, multiple people can play the game on different local devices. Since, this requirement was not clearly defined it is partially satisfied. PE8: The game has a maximum of five users stored in the leaderboard. This was satisfied with manual and unit testing. PE9: Developers are able to implement new features without compromising the functionality of the game. This was tested manually by the developers when implementing new features. The modular designed allowed for a relatively easy development process. PE10: The game maintains functionality with existing software until Spring 2022 (which is now), so this NFR is satisfied. PE11: The game plays with full functionality without internet connection, therefore this is satisfied. PE12: The game does not play on any operating system. This was proven through our usability tests when deploying it to other operating systems. PE13: This requirement is satisfied as the game is deployed using an executable file. PE14: This FR is not satisfied as additional instructions to download the executable is required. PE15: The game was released on April 5th 2022 and was ahead of the schedule release data by one week. Therefore, this requirement is satisfied.

2.4 Maintainability and Support

MA1: This requirement is partially satisfied as it was not specific. All source code was updated with the latest changes before or on the day of scheduled dates. MA2: All source code is documented using Doxygen. This requirement is satisfied. MA3: The full project source code is available to Git users but you must first log-in to your Gitlab account to raise issues. Therefore, this requirement is partially satisfied. MA4: The game only runs on Windows therefore this requirement is partially satisfied.

2.5 Security Requirements

SR1: In the current implementation of the game, once the game is launched the user is not able to edit the Leaderboard. SR2: Users are permitted to change game settings through the user interface therefore the requirement is satisfied. SR3: The product does not protect itself from unauthorized user modifications, this is due to the download method for users. SR4: The user is able to modify the scores due to the download method. SR5: This is satisfied as the user can view other usernames and their corresponding high scores. SR6: This functional requirement is partially satisfied as our game has not been rigorously tested against unauthorized or undesirable software programs. There has been no issues thus far with harmful software.

2.6 Cultural and Political Requirements

CP1: The game themes are not offensive to any cultures or religions therefore this requirement is satisfied. CP2: The game does not allow users to input offensive English user names. This was completed by having a dictionary of inappropriate usernames that the system checked for. The dictionary is currently only in English and therefore does not satisfy the requirement fully.

3 Automated Testing

3.0.1 Settings Page Tests

1. FS-SPT-1

   Initial State: audio variable = True

Input: Keyboard Input N or A

Output: The audio variable will change based on the keyboard input.

2. FS-SPT-2

   Initial State: theme = 'default'

   Input: Keyboard Input "1", "2", or "3"

   Output: Based on the keyboard input, the theme variable value should change:"1" for the default dino theme, "2" for the student theme and "3" for the corona virus theme.

3.0.2 Leaderboard Tests

1. FST-LT-1

   Initial State: User is at the Restart screen waiting for an area to be selected(mouse event)

   Input: User clicks on the area of the "Leaderboard" text

   Output: The Leaderboard page is displayed and contains up to five users and their scores.

2. FST-LT-2

   Initial State: None, this function is called on a unit basis. However the score.txt file must be populated

   Input: Leaders_text (Python list containing top users and their score)

   Output: Unit test passes if the users in the leader board (leaders_text) have the highest recorded scores

3. FST-LT-4

   Initial State: Restart flag is set to True and game point is higher than the current user's highest score.

   Input: Game points

   Output: If game points is higher than the current user high score, unit test passes if the user's high score is updated to the game points.

3.0.3 Instructions Page Tests

1. FST-IPT-1

   Initial State: Current page displayed within interface is the main menu page and instructions = False.

   Input: User selects the area within the “How to play” text

   Output: The current page displayed on the interface will be the instructions page and instructions = True.

2. FST-IPT-2

   Initial State: The Current page displayed within the interface is the instructions page.

   Input: The user enters the associated keyboard exit key (“e”).

   Output: Current page displayed within the interface will transition to the main menu page.

3. FST-IPT-3

   Initial State: The game is paused and instructions = False.

   Input: The user enters the associated keyboard (“i”) input.

   Output: Current page displayed within the interface will transition to the instructions page and instructions = True.

4 Trace to Requirements

Test	Requirements
Functional Requirements Testing	
E1	FR1, FR2
E2	FR3, FR4, FR5
E3	FR6, FR7, FR8
E4	FR9-FR18
Non-functional Requirements Testing	
NFR1	LF1-LF4
NFR2	UH1-UH12
NFR3	PE1-PE14
NFR4	MA1-MA4
NFR5	SR1-SR6
NFR6	CP1, CP2
Automated Testing	
FS-SPT-1	FR7
FS-SPT-2	FR8
FS-LT-1	FR4
FS-LT-2	FR3, FR5
FS-LT-4	FR3, FR5
FS-IPT-1	FR2
FS-IPT-2	FR2
FS-IPT-3	FR1

Table 2: Trace Between Tests and Requirements

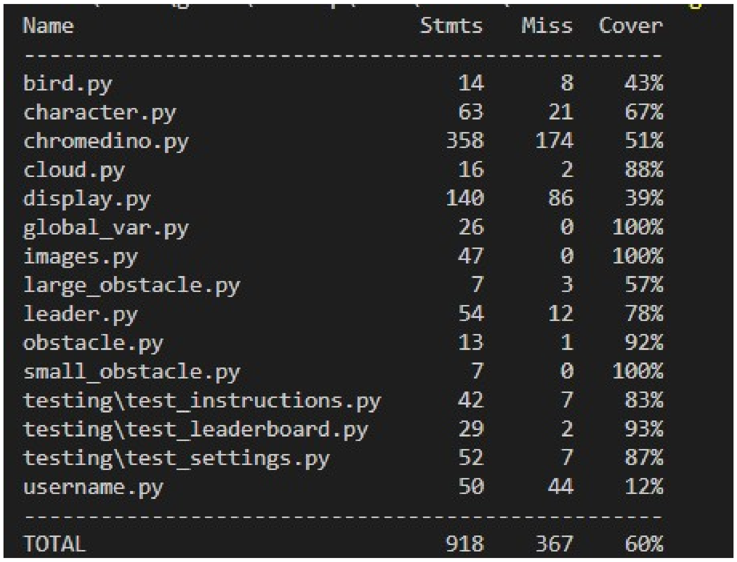
5 Trace to Modules

Test	Modules
Functional Requirements Testing	
E1	M9
E2	M11, M12
E3	M10, M13
E4	M10
Non-functional Requirements Testing	
NFR1	M9, M10, M12, M13
NFR2	M9, M10
NFR3	M10
NFR4	M10
NFR5	M10, M11
NFR6	M11
Automated Testing	
FS-SPT-1	M10, M13
FS-SPT-2	M10, M2, M13
FS-LT-1	M11, M12
FS-LT-2	M11
FS-LT-4	M11
FS-IPT-1	M9
FS-IPT-2	M9, M10
FS-IPT-3	M9, M10

Table 3: Trace Between Tests and Modules

6 Code Coverage Metrics

Our code coverage is roughly 60 percent (see Figure 1 below), which does not meet the 90 percent coverage stated in the Test Plan. This is the case as the team had difficulties passing inputs due to the nature of testing the game, which requires dynamic execution. This 60 percent does not include manual tests and non-functional tests.



```
Name                              Stmts   Miss  Cover
-----------------------------------------------------
bird.py                              14      8    43%
character.py                         63     21    67%
chromedino.py                       358    174    51%
cloud.py                             16      2    88%
display.py                          140     86    39%
global_var.py                        26      0   100%
images.py                            47      0   100%
large_obstacle.py                     7      3    57%
leader.py                            54     12    78%
obstacle.py                          13      1    92%
small_obstacle.py                     7      0   100%
testing\test_instructions.py         42      7    83%
testing\test_leaderboard.py          29      2    93%
testing\test_settings.py             52      7    87%
username.py                          50     44    12%
-----------------------------------------------------
TOTAL                               918    367    60%
```

Figure 1: Code coverage report