Criterion E: Evaluation

Evaluation of the Product:

The product has 5 separate exercises that all function properly, all five will be able to test the user's hand-eye coordination, memory, and reaction time. The product can calculate the user's score for each exercise and display it in the dashboard scene. The product can save/ load/ delete the user's score data by clicking a button. The reaction time test can measure the amount of time it takes for the user to react to the button changing color measures it in milliseconds, and will take the average score of 5 attempts. The sequence memory test lights up a random pattern of buttons, and the length of the pattern/ sequence increases as the user progresses. The aim trainer test randomly places a target on the screen for the user to click on and when clicked on the target will move to another random position on the screen, this will repeat 30 times, then an average of the amount of time taken for each target to be pressed will be calculated, the average is measured in milliseconds. The language memory test uses an array with 2466 different words, with each word being chosen as either seen or new, and each new word shown will become a seen word. There's approximately a 50/50 chance for either a seen word or a new word to be shown to the user, and the user is given 3 lives before the test ends. The number memory test shows the user a random number whose amount of digits increases as the user progresses, the user can input only numbers, enter their answer, and delete the most recently inputted number.

Evaluation Against Success Criteria (refer to **Appendix 4** for client review):

1. The application should have 5 functioning exercises to test hand-eye coordination, memory, and reaction time.

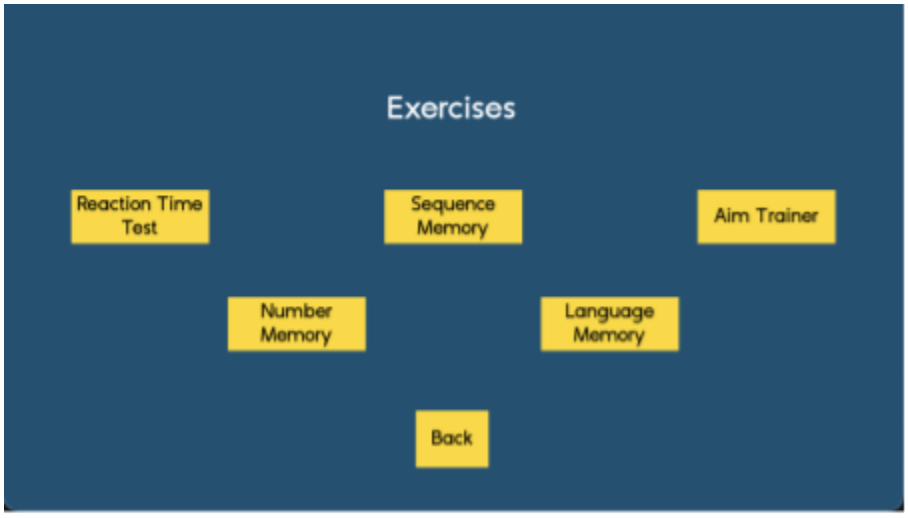

Figure 1: *Exercises Menu*

Success criterion was met: As shown in Figure 1 are the 5 exercises my app offers, reaction time test, sequence memory, aim trainer, number memory, and language memory.

2. Calculate the scores for each of the tests/ exercises and display them.

Success criterion was met: Figure 2 shows the Dashboard scene with example scores for each exercise, and the necessary scores have the appropriate unit (ms = milliseconds).

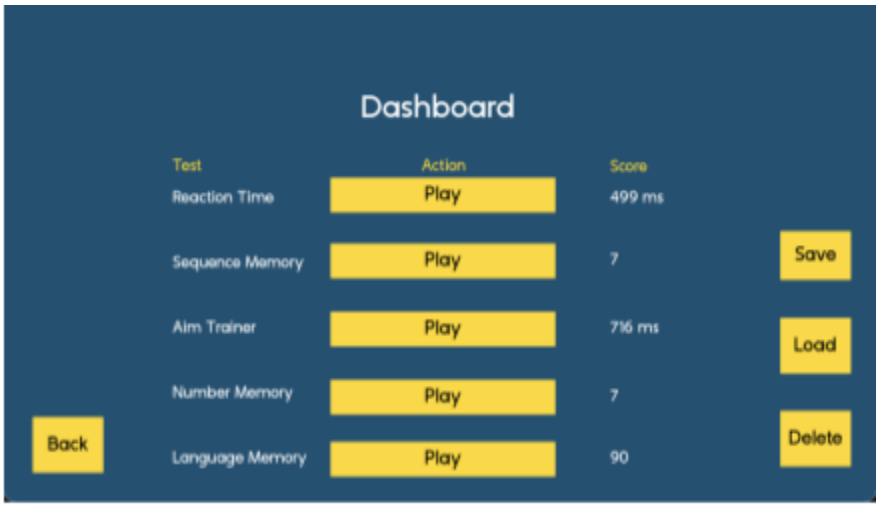

Figure 2: *Dashboard with Scores*

3. User must be able to save/ load/ delete the data from their exercises.

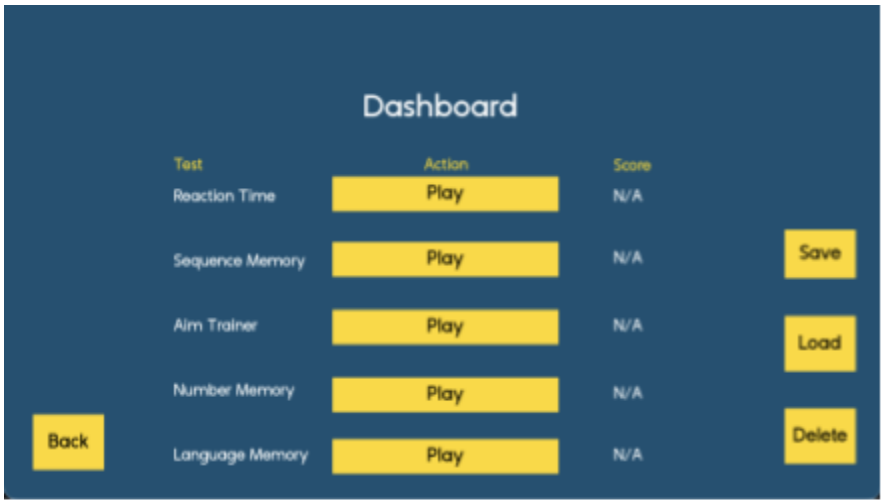

Figure 3: *Regular Dashboard Scene*

Success criterion was met: Figure 3 shows the plain Dashboard scene without any scores, but also as shown in Figure 2, the buttons, save, load, and delete, all perform their intended function, the code for each button is shown in **Appendix 5**.

4. The reaction time test must accurately measure the amount of time it takes for a user to react and find the average of five tests in milliseconds.

Success criterion was met: As shown in Figure 2 the app can measure the reaction time of the user in milliseconds.

5. The sequence memory test must use a random pattern of buttons every time and with every level of the test the length of the sequence must increase.

Success criterion was met: The sequence memory test is able to generate a sequence of numbers that correspond to a button that will then light up in the order of the sequence, which will increase as the user progresses.

6. The aim trainer must provide a certain amount of targets appearing one after another when the user clicks on it, provide an accurate measurement of the time taken for each target, and determine the average time taken, in milliseconds.

Success criterion was met: The aim trainer test provides the user with a total of 30 targets, appearing one after the other when clicked, and the amount of time taken for each target to be pressed is calculated then the average is calculated in milliseconds.

7. The language memory test must use an array of at least 500 words and provide a random word from that array approximately ½ of the time and create a new array of all the words the user has already been shown and show a random word from that list approximately ½ of the time. The user should be given 3 lives, and the test will end when the user runs out of lives.

Success criterion was met: The array used in the language memory test has over 2000 words (figured out by printing new.size() in String form). A random integer is generated, either 1 or 2, so there's a 50/50 chance of a work being chosen from either array (new/ seen). The user has 3 lives and the test ends when there aren't any more lives.

8. The number memory must be able to show the user a random number whose amount of digits increases as the level increases. It must allow the user to input the number they

think is the answer, use the return key to enter their answer and use the delete key to delete the most recently inputted number. The user must only be able to input numbers.

Success criterion was met: The number memory test displays a random number with a length that increases as the user progresses. The user can only begin inputting numbers when the time they're given to memorize is over. The user can input any number, in any order, can enter their answer, and use the delete key to delete the most recently inputted number. And the user can only input numbers.

Feedback from Client (Feedback is found in **Appendix 4**):

Criterion for Success	Client Feedback	Status
The application should have 5 functioning exercises to test hand-eye coordination, memory, and reaction time.	I like that there are 5 options for exercises I can do that challenge me a lot, but I wish it was wider since most of the exercises were memory-based.	Met ✅
Calculate the scores for each of the tests/ exercises and display them.	I thought it was cool that when I finish an exercise, my score appears in the dashboard menu, and it's the same one I got.	Met ✅
User must be able to save/ load/ delete the data from their exercises.	If I save before exiting the app and then reopen it and press load in the dashboard menu, my saved scores reappear.	Met ✅
The reaction time test must accurately measure the amount of time it takes for a user to react and find the average of five tests in milliseconds.	I like how I'm given 5 chances so that my result can be more accurate than if it was just 3 chances. I noticed that if I press the button too early it tells me that I pressed too early and lets me try again. And it gives me results in milliseconds.	Met ✅
The sequence memory test	I like how challenging this	Met ✅

must use a random pattern of buttons every time and with every level of the test the length of the sequence must increase.	one was, I probably enjoyed this exercise the most. The exercise was able to create a sequence of buttons I needed to press in the same sequence, and got challenging much faster than I anticipated.	
The aim trainer must provide a certain amount of targets appearing one after another when the user clicks on it, provide an accurate measurement of the time taken for each target, and determine the average time taken, in milliseconds.	The exercise shows me how many more targets I need to press by showing the "Remaining: (some number, 1-30 depending on how far the user has progressed)." Afterward, it shows me my average time taken in milliseconds.	Met ✅
The language memory test must use an array of at least 500 words and provide a random word from that array approximately ½ of the time and create a new array of all the words the user has already been shown and show a random word from that list approximately ½ of the time. The user should be given 3 lives, and the test will end when the user runs out of lives.	I couldn't tell how many words there were but I definitely never ran out, so there were enough. I also can't tell how often a new or seen word was selected but it seemed very random. When I ran out of my 3 lives the exercise ended, then showed me my final score.	Met ✅
The number memory must be able to show the user a random number whose amount of digits increases as the level increases. It must allow the user to input the number they think is the answer, use the return key to enter their answer and use the delete key to delete the most recently inputted number. The user must only be able to input numbers.	The number changed every time but the amount of digits increased every time. All the numbers I pressed were inputted correctly, and the enter and delete keys worked as intended. I wasn't able to input anything other than numbers.	Met ✅

Improvements:

After talking to my client, getting a lot of very helpful feedback, watching them engage with my app, and debugging the app myself many times. I've found that even though there are already many advantages to my app, there are also many things I can add to my app that can further improve my app and the user experience.

1. Adding a percentile

Adding a percentile of the score compared to the average score can help show the user how good they are in comparison to the population and could help show the user what should be improved.

2. Adding more exercises

The app could be more extensive by adding more exercises that cover more about hand-eye coordination and reaction time as 3 exercises cover memory while only one covers hand-eye coordination and reaction time. This can be done by adding more forms of the aim trainer test or finding other exercises that cover something other than memory, hand-eye coordination, and reaction time that are relevant to my client's problem.

3. Competing online

To make the app more engaging I could add an online component where users can compete in exercises against each other in real time, while still being able to practice by themselves. This could encourage doing these exercises even more and make the app more enjoyable.

4. Leaderboard

Creating another scene or making a website that can compile the high scores of all users for each exercise. By doing this, there will be more competition to do the exercises and to do the best therefore making the app encourage doing the exercises more.

Word Count: 745