GNG2101

Design Project User and Product Manual

User Manual: Computer Mouse Software

Submitted by:

Mouse-E, Team C2.4

Ali Elmawazini, 300317827

Zeyu Shao: 300166803

Ryley MacWilliams: 300263961

Omar Muslat: 300066154

Orel Benkarmona: 300160407

Giovanni Abruzzo: 300233180

April 11th, 2023

University of Ottawa

Table of Contents

T	able o	of Contents	iii
L	ist of	Figures	vi
L	ist of	Tables	ix
L	ist of	Acronyms and Glossary	X
1	In	ntroduction	1
2	О	verview	2
	2.1	Conventions	5
	2.2	Cautions & Warnings	5
3	G	etting started	6
	3.1	Configuration Considerations	14
	3.2	User Access Considerations	16
	3.3	Accessing/setting up the System	16
	3.4	System Organization & Navigation	16
	3.5	Exiting the System	17
4	U	sing the System	18
	4.1	Left Click Exercise.	18
	4.2	Dragging Exercise	21
	4.3	Email Feature	23
5	T	roubleshooting & Support	27
	5.1	Error Messages or Behaviors	27
	5.2	Special Considerations	27

	5.3	Ma	intenance	. 27
	5.4	Sup	pport	. 27
6	Pro	oduct	Documentation	. 28
	6.1	Sub	osystem 1 User Interface	. 29
	6.1	.1	BOM (Bill of Materials)	. 29
	6.1	.2	Equipment list	. 29
	6.1	.3	Instructions	. 29
	6.1	.4	Testing & Validation of UI subsystem	. 31
	6.2	Sub	osystem 2: Login	. 32
	6.2	2.1	Bom (Bill of Materials)	. 32
	6.2	2.2	Equipment list	. 33
	6.2	2.3	Instructions	. 33
	6.2	2.4	Testing & Validation of Login subsystem	. 35
	6.3	Sub	osystem 3 Exercises	. 37
	6.3	3.1	BOM (Bill of Materials)	. 37
	6.3	3.2	Equipment List	. 37
	6.3	3.3	Instructions	. 37
	6.3	3.4	Testing & Validation of Exercises subsystem	. 45
	6.4	Sub	osystem 4: Email	. 46
	6.4	l.1	BOM (Bill of Materials	. 46
	6.4	1.2	Equipment List	. 47
	6.4	1.3	Instructions	. 47
	6.5	Sub	osystem 5: Tutorial	. 50

	6.5.1	BOM (Bill of Material)	50
	6.5.2	Equipment List	51
	6.5.3	Instructions	51
6	.6 Tes	sting and Validation of Final Product	53
7	Conclus	sions and Recommendations for Future Work	57
8	Bibliog	raphy	60
API	PENDICI	ES	61
9	APPEN	DIX I: Design Files	61
10	APPEN	DIX II: Other Appendices	62

List of Figures

Figure 1: UI Menu	2
Figure 2: Exercise Menu	3
Figure 3: Left Clicking Exercise	3
Figure 4: Drag and Drop Exercise	4
Figure 5: Email Exercise	4
Figure 6: Tutorial Page	5
Figure 7: Tutorial Page	6
Figure 8: Menu Button	7
Figure 9: UI Menu Page	7
Figure 10: UI Exercise Page	8
Figure 11: Left Click Exercise	8
Figure 12: Exercise Button	9
Figure 13: Exercise Menu	9
Figure 14: Drag Exercise	10
Figure 15: Exercise Button	10
Figure 16: Email Exercise	11
Figure 17: Email Exercise part 2	11
Figure 18: Email Exercise Send Button	12
Figure 19: Email Exercise Copy and Paste	12
Figure 20: Copy from Right Click	13
Figure 21: Paste from Right Click	13

Figure 22: Paste in Email Exercise	14
Figure 23: Computer	15
Figure 24: Computer Mouse	15
Figure 25: Computer Keyboard	15
Figure 26: Internet Connection	16
Figure 27: Icon sections	18
Figure 28: Exercise Main Menu	19
Figure 29: Left Click Exercise	19
Figure 30: Tutorial Video	20
Figure 31: Options in Left Clicking Exercise	21
Figure 32: Drag Exercise Icon	22
Figure 33: Drag Exercise	22
Figure 34: Objective of Drag Exercise	23
Figure 35: Email Exercise Part 1	25
Figure 36: Email Exercise Part 2	26
Figure 37: Concept of Final Product	28
Figure 38: UI Menu	30
Figure 39: UI Menu	31
Figure 40: Login Subsystem	35
Figure 41: Left Click Exercise	41
Figure 42: Drag and Drop Exercise	45
Figure 43: Figma Email Concept	48
Figure 44: Email Exercise	49

Figure 45: Email Exercise	50
Figure 46: BOM of Subsystems	50
Figure 47: First Email Exercise	57
Figure 48: Final Email Exercise	57
Figure 49: Detail Design	58
Figure 50: Concept 1	62
Figure 51: Concept 2	63
Figure 52: Concept 3	64

List of Tables

Table 1. Acronyms	x
Table 2: Bill of material Prototype 1	29
Table 3: Bill of material for Prototype 2	32
Table 4: List of equipment for prototype 2	33
Table 5: Testing prototype 2 with the established target specifications	36
Table 6: BOM for Exercises	37
Table 7: Bill of material for Prototype 4	46
Table 8: Equipment List for Prototype 4	47
Table 9. Referenced Documents	61

List of Acronyms and Glossary

Table 1. Acronyms

Acronym	Definition
PC	Personal Computer
UI	User Interface
Web-App	Website application

1 Introduction

This User and Product Manual (UPM) provides the information necessary for Computer-Wise to effectively use the Mouse-E product and for prototype documentation. This design was developed in order to teach the users at Computer-Wise to use a computer mouse. Since Computer-Wise is a non-profitable organization that empowered the disabled community, many of their clients are unable to use a computer mouse. As a result, Mouse-E is a software product developed for the purpose of teaching the basic mouse skills without diminishing the physical or mental capabilities of the users at Computer-Wise. Thus, this document informs the techniques used and the steps taken to prototype the design. With that, a detailed description of the prototype testing and a BOM of materials is documented. Overall, the product has been built by taking into account safety and ethical constraints.

Introduction 1

2 Overview

The problem is that individuals with disabilities such as Cerebral Palsy face difficulties in acquiring basic computer skills, particularly in using a mouse. This limits their ability to use computers, which are essential for communication, education, and job opportunities. Therefore, developing an application that teaches basic mouse skills is crucial to provide individuals with disabilities an opportunity to access and benefit from the digital world.

The fundamental needs of the user are to learn basic computer mouse skills that are easy to follow and simple to understand. The application should also be directed towards an adult audience and have an empathetic approach. Individuals with disabilities, such as Cerebral Palsy, require an application that takes into consideration their specific needs and limitations, and offers a user-friendly and accessible interface.

The key aspect that differentiates this product from others is its focus on individuals with disabilities, particularly those with Cerebral Palsy. The application is designed to be empathetic towards the users' needs and limitations, offering a user-friendly and accessible interface. Additionally, the application provides a variety of tasks that are easy to follow and understand, helping users to acquire basic mouse skills effectively. As shown in the following figures representing the final product





Figure 1: UI Menu



Exercises



Figure 2: Exercise Menu

Mouse Skills Home Exercise About

Click the button as much as you can before time runs out!

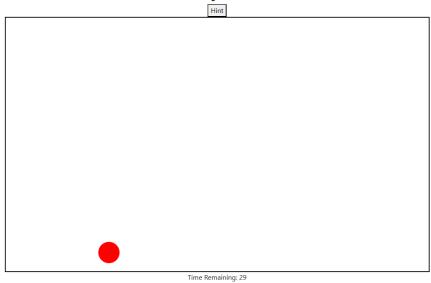
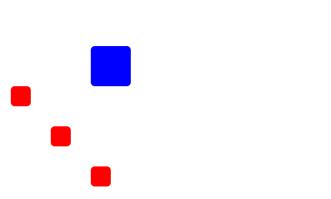


Figure 3: Left Clicking Exercise





Hint

Figure 4: Drag and Drop Exercise

Mouse Skills Home Exercise About

Mouse Skills Email Simulator

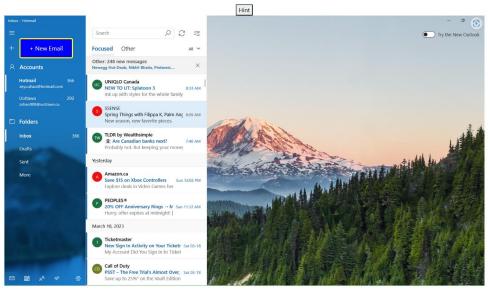
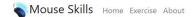


Figure 5: Email Exercise



Tutorial Video



Figure 6: Tutorial Page

The key features of the product include teaching basic computer mouse skills to individuals with disabilities such as Cerebral Palsy, through a variety of easy-to-follow and simple-to-understand tasks, including real world applications such as Outlook. The application is designed with an empathetic approach, specifically directed towards an adult audience, and offers a user-friendly and accessible interface.

The architecture of the system is software-based, and the product is a web application. It does not have any physical components like a metal frame or microcontroller. The user access mode is through a graphical user interface (GUI), which is designed to be easy to use and navigate, with simple buttons and menus. The system environment does not require any special conditions and can be used on any computer with an internet connection.

2.1 Conventions

There is the use of a step system within section 3, documenting the chronological order of the actions required to complete a task.

2.2 Cautions & Warnings

Due to the nature of the product and the plethora of accessibility features that are designed to appeal to the casual PC market, there are no cautions or warnings to be taken to account.

The Mouse-E product is a software that teaches the basic functions and maneuvers of a computer mouse towards people with disabilities. Before diving into the software, some steps are required to accomplish.

1- Have with you a computer or an electronic device that can support a web-browser and a computer mouse in order to properly use the software.

Now that this step is completed, the following steps demonstrate a walkthrough of the software from going to any exercise to watch tutorial videos.

- 2- Copy and paste the following link to open the software: https://zeyushao-uottawa.github.io/Mouse-Skills/
- 3- When the link has been copied, the first screen that will pop-up is a tutorial video where the user can watch or skip depending on their need. This tutorial video explains every basic function of a computer mouse.



Figure 7: Tutorial Page

4- Drag your cursor to the Mouse Skills button and click on it. This button sends the user to the menu.



Tutorial Video



Figure 8: Menu Button

Mouse Skills Home Exercise About



Figure 9: UI Menu Page

- 5- Two images are shown. One is a clickable button sending you to the exercise menu and one is an image to have a general idea of terms that will be used in the instructions.
- 6- Click on the exercise button and you will see a menu for exercises. 3 exercises will be shown: left click, drag and Email.

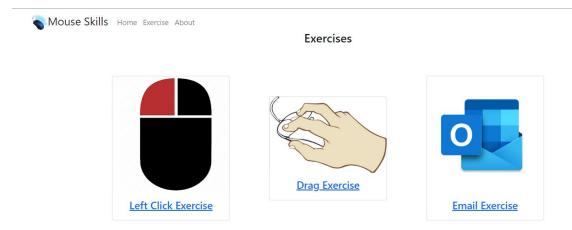


Figure 10: UI Exercise Page

- 7- Depending on the user, they have the freedom to click on any exercise to practise the different computer mouse skills.
- 8- Note that all exercises do not end. The user has full control on whether they want to keep practising or practise another exercise.
- 9- Click on the left click exercise. In this exercise, the user needs to click on the red dot with the left button of the computer mouse. A hint button is available to watch a tutorial video on the exercise.

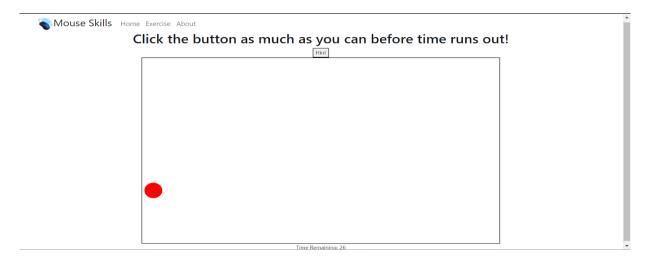


Figure 11: Left Click Exercise

10- Click on the exercise button found in the top left corner to go back to the exercise menu.

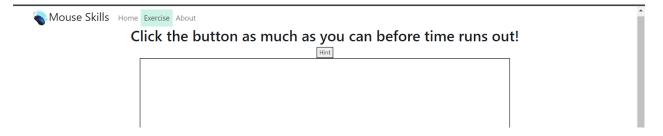


Figure 12: Exercise Button

11- click on the drag exercise.

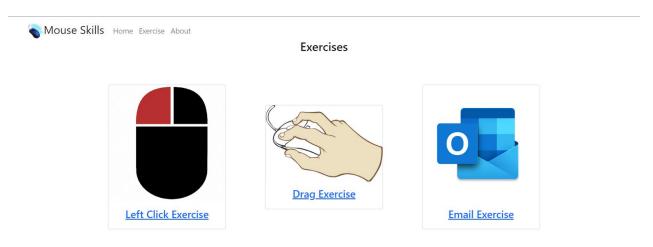


Figure 13: Exercise Menu

12- This exercise practises drag and drop manoeuvres while using a computer mouse. As mentioned in step 10 there is no ending aspect in the exercise. The goal is to drag the red box into the blue box. Similarly, to the left click exercise, there is a hint button to watch a tutorial video on the exercise.

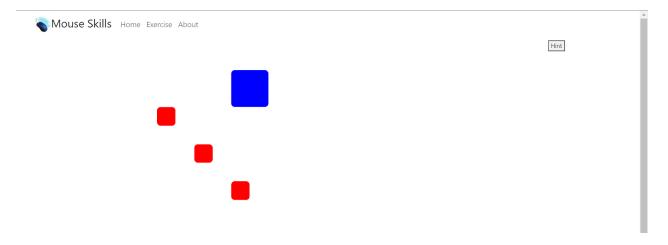


Figure 14: Drag Exercise

13- Similar to step 10, click on the exercise button found in the top left corner to send you back to the exercise menu.

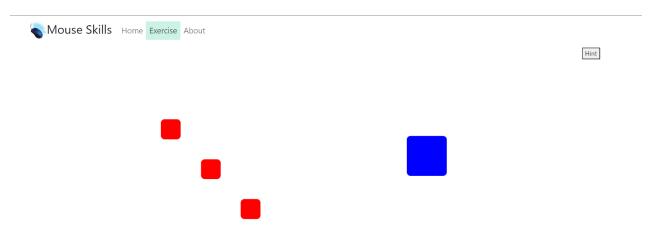


Figure 15: Exercise Button

14- Click on the final exercise: Email exercise. This exercise teaches the user how to use left clicking, right clicking and dragging. Similar to the previous 2 exercises, there is a hint button for the exercise.

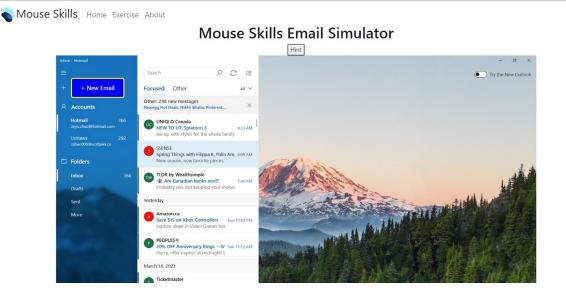


Figure 16: Email Exercise

- 14.1- Click on the New Email button found in the left corner of the email.
- 14.2- Now compose an email by entering the information you want to enter.

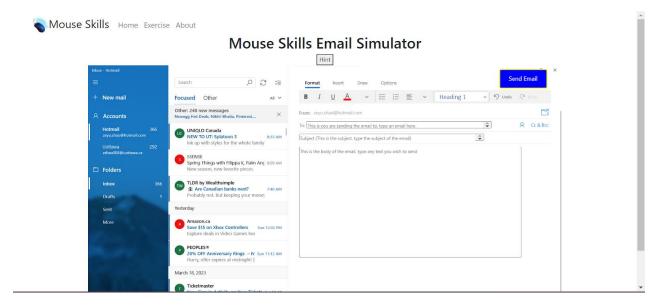


Figure 17: Email Exercise part 2

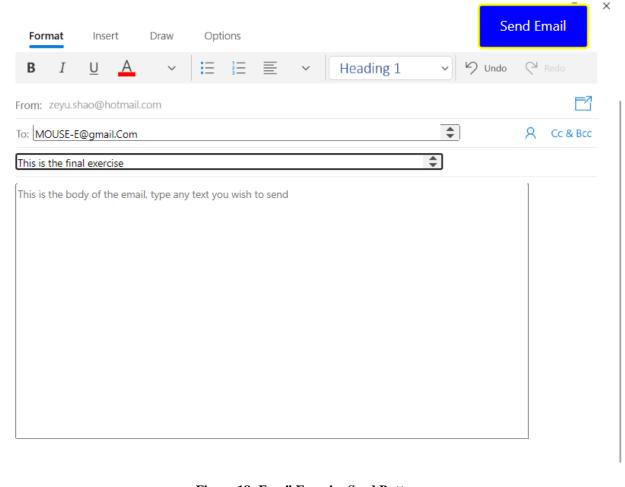


Figure 18: Email Exercise Send Button

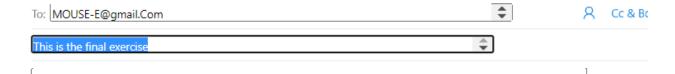


Figure 19: Email Exercise Copy and Paste

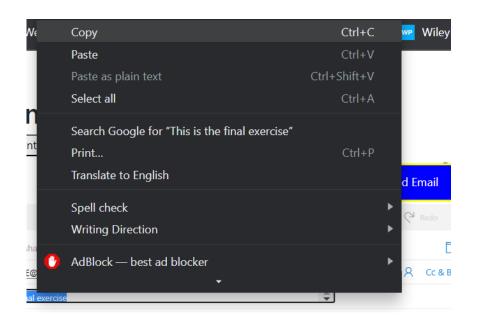


Figure 20: Copy from Right Click

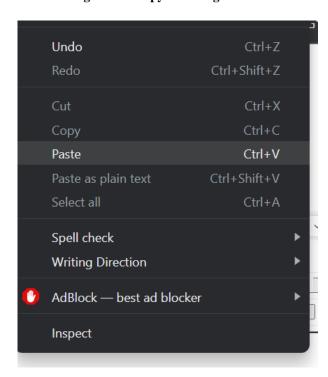


Figure 21: Paste from Right Click

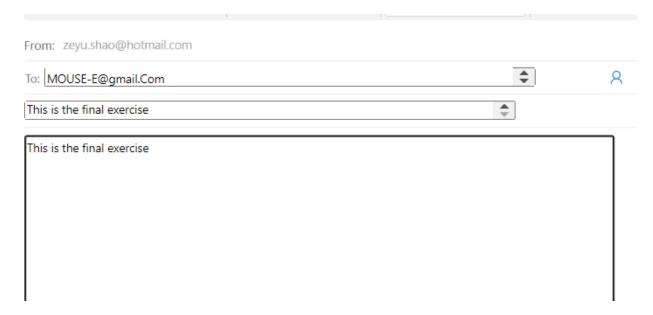


Figure 22: Paste in Email Exercise

- 14.3- Copy and paste using the right click button the information in the subject section into the body section
- 14.4- Click on Send Email.
- 15. Now that the exercise is done, the user has the freedom to repeat anything they feel they need more practice on whether it is a different exercise, the same exercise or just watch a tutorial video.

3.1 Configuration Considerations

To use the final product of Mouse-E which is a computer software that teaches individuals the basics of computer mouse operations, the team used equipment such as a computer, a computer mouse, a keyboard and a video camera. As for developing the software, the team downloaded multiple software's and libraries. The software's downloaded are GitHub, Vue.js and VS code. Those software's allowed us to develop a web-application. As for the libraries, the team downloaded Bootstrap, Vite.Js, Firebase and Prettier. Hence, the inputs are the software's, the codes and the libraries and the output is the screen that the user is able to see and communicate with by using the equipment listed.

As for the users, they only need the equipment such as a computer, a mouse, a keyboard and an internet provider to open the web-app. As for opening the website, they only need the website link which is the following: https://zeyushao-uottawa.github.io/Mouse-Skills/.

Below are images the equipment described:



Figure 23: Computer



Figure 24: Computer Mouse



Figure 25: Computer Keyboard



Figure 26: Internet Connection

3.2 User Access Considerations

While the purpose of the product was to introduce mouse operations to inexperienced PC mouse users, the initial user base are the people from Computer Wise. These individuals are predominantly elderly individuals as well as individuals with disabilities.

3.3 Accessing/setting up the System

In order for the user to access Mouse-E, the user must first open their computer and plug in their keyboard and computer mouse. Then, when everything is properly set up, select whichever web browser that feels familiar. Then, copy and paste the following URL link into the search tab to have access to the software https://zeyushao-uottawa.github.io/Mouse-Skills/. From there, the user has fully accessed the software and can begin practicing some basic mouse skills.

3.4 System Organization & Navigation

The system is organized into pages, each page is a different component that contains differing functions/features. There are a total of 6 pages, the tutorial page, home page, exercise view page, left click exercise page, drag exercise page, and email exercise page. Each page contains a navigation bar (or a menu) at the top that contains routes to the main menu page, exercise view page, and tutorial page. In the exercise view page, there are links to access the three exercises.

The user begins from the tutorial page, which contains a video that guides the users through the mouse and the website. The user can then move on to the main menu page, which contains a link to the exercise view and pictures. From there they can go to the exercise view page and see links to all three exercises. In each of the three exercises, there are the actual exercises they can

perform and then if they want to go back or to a different page they can click on the navigation bar at the top.

3.5 Exiting the System

In order for the user to exit our website, they simply need to close the tab or ctrl + W.

4 Using the System

The following sub-sections provide detailed, step-by-step instructions on how to use the various functions or features of the software Mouse-E.

4.1 Left Click Exercise

First, to access the left click exercise, navigate to the top menu bar and click on the "Exercise" tab. Once you click the exercise tab, you will be greeted with a page that has three images with text below them. One image will be of just a computer mouse and will have the title "Left Click Exercises". If you left click inside the box that contains the text and image you will be brought to a page that says "Click the button as much as you can before time runs out!". You will then be able to navigate your mouse over top of the red circle that appears at a random spot on the screen and when you left click the red dot, the dot will disappear and reappear at an entirely new and random location on the user's screen. Inside of this exercise's page, there is also a "hint" button that when clicked, will show a pop up of a YouTube tutorial video that explains and shows the user how to properly complete the exercise. Then you can click the "Exit" button in the bottom right of the pop up tutorial video to go back to the screen where you can complete the exercise. At the bottom of the exercise screen there is a 30 second countdown timer that begins as soon as you enter the tab. Once the timer runs out you will be greeted with two options, "Restart Game" and "Freeplay". If you left click "Restart Game" the timer will start again from 30 seconds and begin the countdown. The game can be restarted as many times as the user wants to try to beat their high score or to simply practice. If you click the "Freeplay" button, then the count down timer gets replaced with "Game over! Your score is X." with X being whatever score that the had just got. This score will then increase by increments of one for each time that another red dot is left clicked. This section of the game can go on until the user wants to switch exercises or is done with the website. If the user wants to navigate to a new exercise, they can navigate back up to the "Exercise" tab at the top of the website and click on that to then select a new Exercise. If the user is done with their practice, they can close the tab at any time how you normally would on your browser or you can click the "Control" key then the "W" key on their keyboard to close the tab.



Figure 27: Icon sections

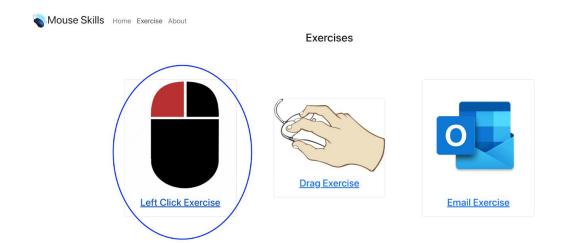


Figure 28: Exercise Main Menu

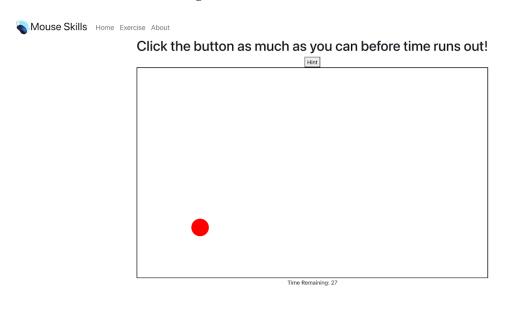
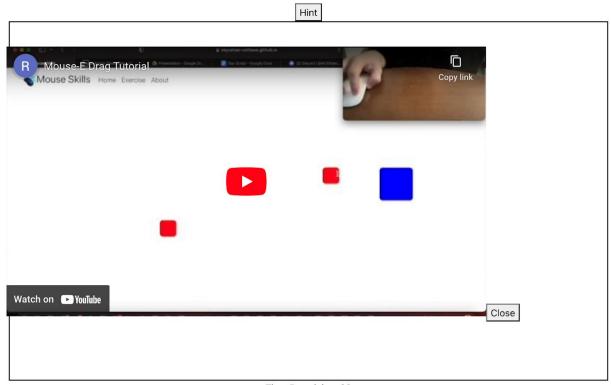


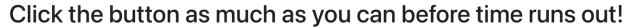
Figure 29: Left Click Exercise

Click the button as much as you can before time runs out!



Time Remaining: 23

Figure 30: Tutorial Video



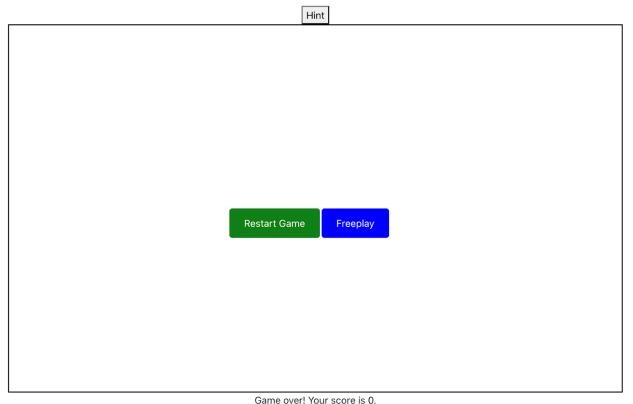
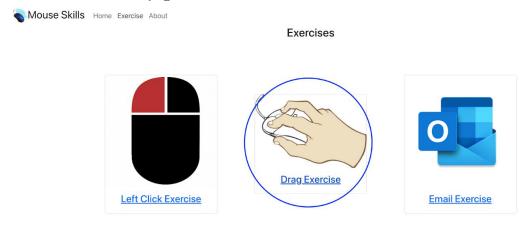


Figure 31: Options in Left Clicking Exercise

4.2 Dragging Exercise

The following instructions will be to help the user understand how to use our click and drag mouse skills exercise. Firstly, you will follow the same steps above to access the exercise page. Once you are here, you will see the option to select the "Drag Exercise", this will also have an image of a mouse with an animated hand on it. Upon clicking this exercise, the user will be brought to a screen where there are three red boxes (the objects), and one bigger blue box (the target). The user will be given three objects to start with and will only be given more once all of the objects have been successfully moved into the target box. The user completes the task by left clicking and holding on one of the objects and moving the object over top of the target, then dropping the object by releasing the left click button. Once the object is successfully released into the target box, the target box will move to a new and completely random location on or off of the user's screen. If it moves off of the user's screen, the user will have to hold down on the middle click button and drag the mouse around until they navigate to find the box and are able to bring another object into the target box and complete the task. Exactly like the previously mentioned "Left Click Exercise",

there is an integrated hint button that provides detailed step by step instructions and can be accessed and closed the same way as mentioned above. There is no end to this exercise meaning that the user can practice as much as they want for as long as they want and when they are done, they can simply exit the website the normal way you would with your browser or by using the keyboard shortcut "control + W". Another way that you can exit the exercise is by navigating to another tab on the website whether that be the "Home", "About", or "Exercise" page.



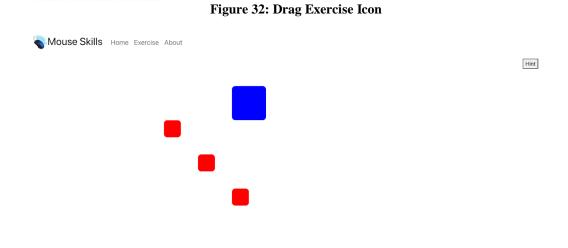


Figure 33: Drag Exercise

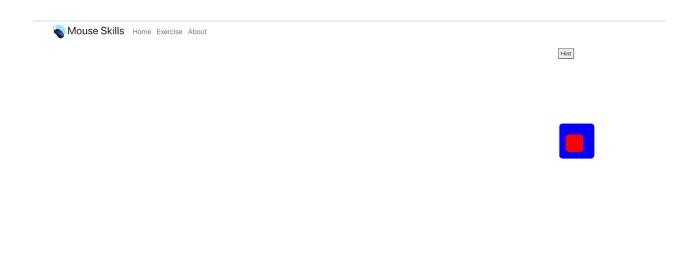


Figure 34: Objective of Drag Exercise

4.3 Email Feature

Learning how to use the email function in Outlook is a fundamental aspect of using the software. It can seem intimidating at first, but with a little bit of practice with our product, you'll quickly become proficient in using this essential feature.

Here are some steps you can take to learn how to use the email function in Outlook:

Familiarize Yourself with the Interface

1. The first step in learning how to use the email function in Outlook is to familiarize yourself with the interface. The main interface consists of several sections, including the Ribbon, Navigation Pane, and Reading Pane. Take some time to explore each of these sections to understand what each one does.

Compose an Email

2. To compose an email, click on the "New Email" button on the Ribbon. This will open a new email message window, where you can enter the recipient's email address, subject, and message. The email composer works much like a word processor, so you can format your message and add attachments as needed.

Customize Your Email

3. Outlook offers many options for customizing your email. You can choose different fonts, colors, and sizes for your text. You can also add images and hyperlinks to your message. Take some time to experiment with these features to get a feel for how they work.

Manage Your Inbox

4. Once you've sent an email, it will appear in your inbox. Managing your inbox is an important part of using Outlook's email function. You can mark emails as read or unread, flag them for follow-up, and delete them when you no longer need them.

Organize Your Emails with Folders

5. Outlook allows you to organize your emails by creating folders. To create a new folder, right-click on your inbox and select "New Folder." You can then drag and drop emails into your new folder to keep your inbox organized.

Use Email Templates

6. Outlook offers email templates, which are pre-written emails that you can use to save time. To access email templates, click on the "New Email" button on the Ribbon and then click on "More Items." From there, select "Choose Form" and choose the template you want to use.

Utilize Filters and Rules

7. Outlook also allows you to set up filters and rules to automatically sort your incoming emails. For example, you can set up a rule to move all emails from a particular sender to a specific folder. To set up filters and rules, click on the "Rules" button on the Ribbon and then click on "Manage Rules & Alerts."

Use Email Signatures

8. Outlook also allows you to create email signatures that will appear at the bottom of all your outgoing emails. To create an email signature, click on the "File" tab, select "Options," and then select "Mail." From there, click on "Signatures" and create a new signature.

Practice Makes Perfect

9. he most important step in learning how to use the email function in Outlook is to practice. The more you use it, the more comfortable you will become with the software. Try sending emails to yourself or to a friend to get some practice in.

In conclusion, learning how to use the email function in Outlook may seem overwhelming at first, but by following these steps and practicing regularly, you'll soon become proficient in using this essential feature. With its powerful features and customization options, Outlook is an excellent tool for managing your email communications.

Search New Email Accounts Notice 248 new messages Neweys for the whole family Sink Williams 272 photoCollegendinesca 165 Seria 160 Try the New Couldook Folders Inbox 366 Table Share Shar

Mouse Skills Email Simulator

Figure 35: Email Exercise Part 1

Mouse Skills Email Simulator

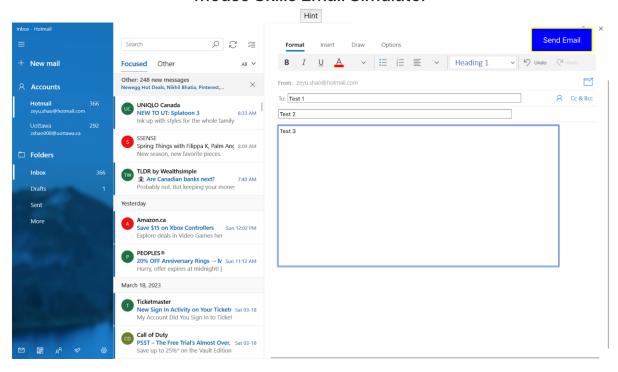


Figure 36: Email Exercise Part 2

Using the System 26

5 Troubleshooting & Support

Since this is a website hosted on a server, users that have access to the code are able to fix bugs or issues that arise, otherwise, for known issues, troubleshooting for them will be listed below.

5.1 Error Messages or Behaviors

On the second exercise where the user is meant to place the red squares into the blue square. The blue square will move off screen and the user is meant to use the middle cursor in order to move the screen. They may encounter the possibility that the middle cursor stays activated and the user will need to refresh the screen in order to fix the issue.

5.2 Special Considerations

While using some exercise, it would be encouraged to consider using the hint button where a step-by-step tutorial video will appear. Also, some heavy and fast maneuvers might break down the dragging exercise. To avoid this situation, simply refresh the website page.

5.3 Maintenance

Regular maintenance that should be performed on this prototype is to reload the website before each use session and to regularly restart the computer or web browser to ensure no faulty cache storage.

5.4 Support

In the event that support is needed, the user should speak with Travis as he will be the onsite support individual that will be best versed with the website. If an issue arises with the website, users are instructed to first refresh the site and if the problem persists to send a support ticket at the following email: omusl059@uottawa.ca, aelma032@uottawa.ca and ael

To build the final design, the team had developed three concept. After reviewing the concepts and receiving feedback from our client on our concepts, we were able to chose the following concept:

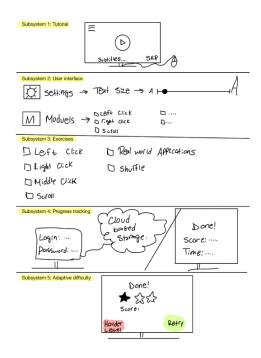


Figure 37: Concept of Final Product

Then, the team developed many prototypes in order to represent each subsystem. As a matter of fact, we have begun doing a detailed design by using a software named sigma. We used this software because it allowed us to properly have an aligned software detailed design where steps are described. Then, we built a physical model as a proof of concept. Afterwards, we went on to prototype the subsystems such as the user interface subsystem, the tutorial subsystem, the exercise subsystem and the login subsystem.

Note that due to some technical difficulties, the logging subsystem was not implemented into the final design because it was non functional. However, this was a prototype that the team developed and is willing to fix in the future.

6.1 Subsystem 1 User Interface

6.1.1 BOM (Bill of Materials)

Below, you will find the necessary materials and tools needed to accomplish a similar product.

Table 2: Bill of material Prototype 1

Item Name	Description	Unit of measure	Quantity	Unit cost	Extended Cost	Link
Cost of Labour	The cost for the team to work on this project	N/A	N/A	\$0	\$0	N/A
Bootstrap	Library to edit the display of the application	N/A	1	\$0	\$0	LINK
Vite.js	Library to develop a framework environment	N/A	1	\$0	\$0	<u>LINK</u>
Firebase	Library providing a framework environment	N/A	1	\$0	\$0	LINK
	Total product cost (without		\$0	_		
	Total product cost (including		\$0	-		

6.1.2 Equipment list

Due to the nature of the project being software, the only equipment needed is your computer as well as a stable internet connection in order to search the features you'd like to implement.

6.1.3 Instructions

To build this subsystem, the team used the Vue.js platform theme and from there, the team used to build the following code by researching codes and techniques from YouTube and Vue.js website. After setting up the main page, the team created a new file that would connect all other pages through a routing method. This was used by coding in html, CSS, JavaScript and typescript. The following code represents the main menu of our user interface. To replicate this code, copy and paste it into the VScode software, and run the code through the terminal. A link should be present in the terminal where you are able to press on it and look at the main menu of the website.

29



Figure 38: UI Menu

```
descript setup lang="ts">
import { RouterLink } from "vue-router";
import Mouse from "@/assets/mousebreakdown.png";

descript

des
```

```
<style scoped>
.menu {
  align-items: center;
  justify-content: center;
  display: flex;
  font-size: xx-large;
}
</style>
```

6.1.4 Testing & Validation of UI subsystem

The testing of the UI was done with the goal of having a simple and intuitive UI that will not confuse users. For this reason, the exercise section is one big square in order for users, regardless of their eyesight, to be able to find the exercise menu. This was tested through allowing relatives of varying ages to give feedback on the UI of the product. The simple colors used are also inline with simplifying the UI and the choice of beige was used as most individuals can see the color beige without any difficulty.





Figure 39: UI Menu

6.2 Subsystem 2: Login

The overall goal of this prototype was to develop a physical focused prototype with the objective of having a progress tracking prototype. However, this subsystem failed and had too many issues that were above the team's expertise since we needed to access the users computer and personal data in order for them to log in. As a result, this prototype is of low fidelity and was removed from the initial design concept.

6.2.1 Bom (Bill of Materials)

For this prototype we have used the following bill of material:

Table 3: Bill of material for Prototype 2

Item Name	Description	Unit of measure	Quantity	Unit cost	Extended Cost	Link
Cost of Labour	The cost for the team to work on this project	N/A	N/A	\$0	\$0	N/A
Bootstrap	Library to edit the display of the application	N/A	1	\$0	\$0	LINK
Vite.js	Library to develop a framework environment	N/A	1	\$0	\$0	<u>LINK</u>
Firebase	Library providing a framework environment	N/A	1	\$0	\$0	LINK
	Total product cost (without		\$0	_		
	Total product cost (including	ng taxes and sh	nipping)		\$0	_

6.2.2 Equipment list

Table 4: List of equipment for prototype 2

Item Name	Description	Туре	Source
Figma	Software to create a detailed design	Software	<u>LINK</u>
Vue.js	To code and create the application	Software	LINK
Computer	To use the software's in order to code	Equipment	LINK
Computer mouse	The computer mouse will be used in order to test functionality	Equipment	LINK

6.2.3 Instructions

To build this prototype, the team only needed to use the previous software's from prototype 1 and we also used various sources such as stack overflow to find some code. We also watched multiple YouTube videos in order to determine the required knowledge to code a login subsystem.

In the following images, a screenshot of the code of our login subsystem is presented and an image of its implementation onto the website. Similarly, to all other prototypes, this prototype was coded and tested with our target specifications.

To reproduce this code, copy and paste the below code into the VScode program, submit save and submit changes. Then in the terminal write "npm run dev" where a link of the website will show up, click on that link and then the website page you will see is the image in figure 9.

```
<input type="password" id="password" v-model="login.password"</pre>
  <button type="submit" @click.prevent="submitLogin">Submit</button>
    <h2>Sign Up</h2>
        <input type="text" id="name" v-model="signup.name" />
        <label for="email">Email:</label>
        <input type="email" id="email" v-model="signup.email" />
        <label for="password">Password:</label>
<input type="password" id="password" v-model="signup.password" />
<button type="submit" @click.prevent="submitSignup">Submit</button>
  data() {
      login: {
        password: ""
      signup: {
        email: "",
        password: ""
  methods: {
    submitLogin() {
      console.log("Submitting login form with:", this.login);
      this.login.password = "";
    submitSignup() {
      console.log("Submitting signup form with:", this.signup);
```

```
// Reset signup form
  this.signup.name = "";
  this.signup.email = "";
  this.signup.password = "";
}
}
</script>
```

Mouse Skills	Home	Exercise	About
Login Email: Password: Submit			
Sign Up			
Name: Email:		\dashv	
Password:			

Figure 40: Login Subsystem

In this code, we define variables and include various buttons and text boxes. To use it, the user only needs to fill in the information and press submit. As explained earlier in this manual, the system does not work because it would not save any information or account. Hence, we decided to remove it from our final design because it was not functional.

6.2.4 Testing & Validation of Login subsystem

To test this prototype, the team has used it in order to simulate a real life login system. The test indicated a failure because it was not functional hence, the subsystem was not utilised in the final design. However, we still tested it according to the target specifications to determine if it was functional and how well the performance of this subsystem would be effective in a final implementation. Below are values that we have derived from testing and assumptions.

Table 5: Testing prototype 2 with the established target specifications

	User interfac e design comple xity (Rating 1-5)	Transiti on time (s)	Numb er of tasks (# quanti ty)	Difficu lty of tasks (rating 1-5)	Numb er of modul es (#)	Differ ent colour s in the palette (# colors)	Age Recommend ation Age (#)	Font size (# font)	Co st (\$)
Ideal target specificati ons	2	1s	3	2	5	4	>18 years	18	\$0
Login prototype tested	2	1s	2	2	0	1	>15 years	Betwe en 14	\$0

6.3 Subsystem 3 Exercises

6.3.1 BOM (Bill of Materials)

Table 6: BOM for Exercises

Item Name	Description	Unit of measure	Quantity	Unit cost	Extended Cost	Link
Cost of Labour	The cost for the team to work on this project	N/A	N/A	\$0	\$0	N/A
Bootstrap	Library which provides modifiable premade components	N/A	1	\$0	\$0	LINK
Vue.js	Framework that creates the website using JS code	N/A	1	\$0	\$0	LINK
Node.js	Package/Library manager and support for running vue.js	N/A	1	\$0	\$0	LINK
	Total product cost (without ta	\$0	-			
	Total product cost (including	taxes and shi	pping)		\$0	_

6.3.2 Equipment List

The only physical equipment that was needed to build this subsystem is our personal laptops and the software equipment (applications) that we used to create our website is GitHub and Visual Studio Code (VS Code).

6.3.3 Instructions

All of our coding subsystems were built using the same method. This method consisted of trying to get as far as we could with our own coding knowledge, then expanding on this knowledge by searching for detailed and specific YouTube video tutorials on how to write code for a similar project, then finally if there were any gaps left in the code we would either search for very specific and detailed things or craft a detailed and specific prompt that we could plug into Chat GPT being sure to detail exactly what we wanted. For example, if we were looking for code to make a text box, we would ask Chat GPT a prompt that detailed everything from the positioning of the text box all

the way to the specific coding language that we were using for that section of our code. Most of the code and most of the Chat GPT prompts can be found in Appendix II: Other Appendices.

Exercise 1:

```
<script setup lang="ts">
import { ref } from 'vue';
var position x = Math.random() * (1000-50);
var position y = Math.random() * (600-50);
const showButton = ref(false);
const score = ref(0);
const gameOver = ref(false);
const startTime = ref(0);
const totalTime = ref(30);
const elapsedTime = ref(0);
const remainingTime = ref(totalTime.value);
const showHintModal = ref(false);
function showHint() {
  showHintModal.value = true;
function closeHint() {
  showHintModal.value = false;
function handleClick() {
  position x = Math.random() * (1000-50);
  position y = Math.random() * (600-50);
  if (showButton.value) {
    score.value++;
    showButton.value = false;
    setTimeout(() => {
      showButton.value = true;
    }, Math.random() * 1000 + 100);
    gameOver.value = true;
```

Product Documentation

38

```
startGame();
     function startGame() {
       score.value = 0;
       gameOver.value = false;
       showButton.value = true;
       startTime.value = Date.now();
       setInterval(() => {
         const currentTime = Date.now();
         elapsedTime.value = Math.round((currentTime - startTime.value)
1000);
         remainingTime.value =
                                          Math.max(totalTime.value
elapsedTime.value, 0);
       setTimeout(() => {
        showButton.value = false;
         gameOver.value = true;
       }, totalTime.value * 1000); // game ends after 30 seconds
     function freeplay() {
       freePlay.value = true;
       showButton.value = true;
       <div class="container-fluid" style="text-align: center; margin:</pre>
auto; height: 92vh;">
         <h1>Click the button as much as you can before time runs out!</h1>
         <button class="hint-btn" @click="showHint">Hint/button>
         <div class="hint-modal" v-if="showHintModal" style="position:</pre>
absolute; z-index: 10; top: 15%; left: 22%">
height="450px" src="https://www.youtube.com/embed/1P0k6Kc4SU8" title="Left
Click Tutorial" frameborder="0" allow="accelerometer; autoplay; clipboard-
                                                                web-share"
allowfullscreen></iframe>
         <div class="game-area" style="margin:auto;">
           <button v-if="(showButton && !gameOver) || (showButton &&</pre>
freePlay)" id="game-button" @click="handleClick" :style="{ top: position y
+ 'px', left: position x + 'px' }"></button>
```

```
v-if="gameOver
                                                        & &
                                                             !freePlay"
@click="startGame">Restart Game</button>
           <button class="freeplay"</pre>
                                      v-if="gameOver
                                                        & &
                                                             !freePlay"
@click="freeplay">Freeplay
         Game over! Your score is {{ score
         Time Remaining: {{remainingTime}}
     <style scoped>
     .game-area {
       width: 1000px;
       height: 600px;
       border: 2px solid black;
     #game-button {
       position: absolute;
       width: 50px;
       height: 50px;
       border: none;
       border-radius: 50%;
     .restart {
       position: absolute;
       top: 50%;
       left: 50%;
       background-color: green;
       font-size: 16px;
       padding: 12px 24px;
       border: none;
       border-radius: 5px;
       text-align: center;
       transform: translate(-140px, 0px);
       -ms-transform: translate(-600px, -300px);
     .freeplay {
       top: 50%;
       left: 50%;
       background-color: blue;
       font-size: 16px;
```

```
padding: 12px 24px;
border: none;
cursor: pointer;
border-radius: 5px;
text-align: center;
transform: translate(10px, 0px);
-ms-transform: translate(-600px, -300px);
}
</style>
```

Click the button as much as you can before time runs out!

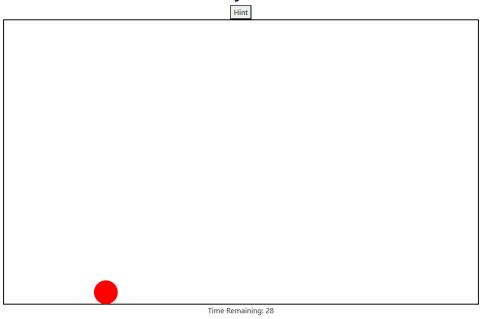


Figure 41: Left Click Exercise

Exercise 2:

```
class="item"
            v-for="(item, index) in items"
            :key="index"
            :style="{ top: item.top + 'px', left: item.left + 'px' }"
            ref="item"
            @mousedown="startDrag(index)"
            @mouseup="stopDrag(index)"
          <button class="hint-btn" @click="showHint">Hint
                                                              width="800px"
height="450px"
web-share" allowfullscreen></iframe>
           <button class="close-btn" @click="closeHint">Close</button>
       data() {
           dragging: false,
           mouseX: 0,
           mouseY: 0,
           boxTop: 100,
           boxLeft: 400,
           showMessage: false,
            items: [
             { top: 200, left: 200 },
            { top: 300, left: 300 },
            { top: 400, left: 400 },
           showHintModal: false,
       methods: {
          startDrag(index) {
           this.dragging = index;
           this.mouseX = event.clientX;
           this.mouseY = event.clientY;
          stopDrag(index) {
            this.dragging = false;
            const item = this.items[index];
              item.left < this.boxLeft + 100 &&</pre>
```

```
item.top > this.boxTop &&
        item.top < this.boxTop + 100</pre>
        this.items.splice(index, 1);
        if (this.items.length === 0) {
          this.respawnItems();
        this.showMessage = true;
    handleDrag(event) {
      if (this.dragging !== false) {
        const item = this.items[this.dragging];
        item.left += event.clientX - this.mouseX;
        item.top += event.clientY - this.mouseY;
        this.mouseX = event.clientX;
        this.mouseY = event.clientY;
    respawnBox() {
      this.boxTop = Math.random() * (window.innerHeight - 100);
      this.showMessage = false;
    respawnItems() {
      this.items = [
       { top: 200, left: 200 },
        { top: 300, left: 300 },
        { top: 400, left: 400 },
    showHint() {
      this.showHintModal = true;
    closeHint() {
     this.showHintModal = false;
    window.addEventListener("mousemove", this.handleDrag);
  beforeUnmounted() {
    window.removeEventListener("mousemove", this.handleDrag);
<style scoped>
.hint-btn {
```

```
position: absolute;
top: 20px;
right: 20px;
  height: 100vh;
  display: flex;
  justify-content: center;
  align-items: center;
position: absolute;
width: 100px;
height: 100px;
background-color: blue;
border-radius: 10px;
align-items: center;
.message {
font-size: 24px;
font-weight: bold;
text-align: center;
position: absolute;
height: 50px;
background-color: red;
border-radius: 10px;
```

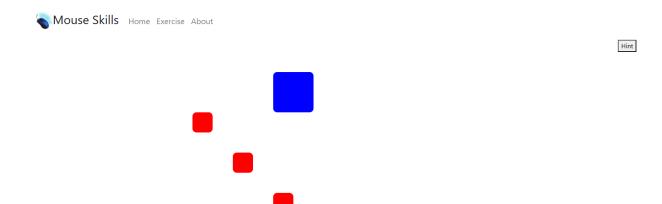


Figure 42: Drag and Drop Exercise

6.3.4 Testing & Validation of Exercises subsystem

The tests that were done were us simply creating a local host link for the website and trying out each of our exercises as we were working on it to ensure various text boxes were in the right place, colors were correct, and any moving objects would appear in the correct places at the correct times. One issue that we did encounter is that for our second exercise, the drag and drop exercise, we kept having the box go off of the screen. Initially, we thought this was a problem, however, we realized that this was a great time to incorporate training for the middle mouse click button. We ended up allowing the blue box to occasionally move out of the computer's screen and simply added the instruction to middle click and drag to navigate and find the blue box's new location.

Other tests included ensuring that the user would not get bored of the left click exercise due to the 30 second timer being too long. However, after just a few tests on many different people, we realized that people ended up staying for much longer to engage in the free play version of that exercise as it was fun for even users who already know how to use a mouse.

A final test that we did was ensuring that the clickable boxes on the Outlook exercise appeared in the same spot and were the same size (in relation to the background photo) at all times on all sized laptops and screens. This caused some trouble at first. However, after testing various different codes and looking at various different Reddit posts, we were able to find the flaw in our code and were able to fix the issue of the box appearing in random places, unproportionate to the background image.

6.4 Subsystem 4: Email

6.4.1 BOM (Bill of Materials

Table 7: Bill of material for Prototype 4

Item Name	Description	Unit of measure	Quantity	Unit cost	Extended Cost	Link
Cost of Labour	The cost for the team to work on this project	N/A	N/A	\$0	\$0	N/A
Bootstrap	Library which provides modifiable premade components	N/A	1	\$0	\$0	LINK
Vue.js	Framework that creates the website using JS code	N/A	1	\$0	\$0	<u>LINK</u>
Node.js	Package/Library manager and support for running vue.js	N/A	1	\$0	\$0	LINK
	Total product cost (without taxes or shipping)				\$0	_
	Total product cost (including	\$0	_			

6.4.2 Equipment List

Table 8: Equipment List for Prototype 4

Item Name	Description	Туре	Source
VS Code	IDE that allows for editing and running code	Software	<u>LINK</u>
Figma	Application that can be used to create wireframes and low level prototypes	Software	LINK
Github	Source control software that allows for the syncing of code	Software	LINK
Computer	To use use the softwares in order to code	Equipment	LINK
Computer mouse	The computer mouse will be used in order to test functionality	Equipment	LINK

6.4.3 Instructions

To create this subsystem, we first create a rough sketch on figma of how the exercise should look like. We used screenshots of the outlook application to act as the application our email exercise will be replicating. Then, basing the design off the figma sketch, we started coding this page in VS code in a JS file to make it look like our sketch. Once we had successfully coded the design and it was looking good on the website, we had to add functionality for it. We added a button which would compose a new email, and to do that we had to code a function which would run when the button is clicked. The function basically sets the state exercise to be in compose mode, which would reveal certain text boxes and a button called send email. Once the user has type what they wish to type in the text boxes, they click the button which has a function that turns the state of the exercise to the original state, but also sends out an alert which lets the user know they have successfully sent an email. After implementing the functionality, the email subsystem prototype is complete. Similarly, to all other prototypes, this prototype was coded and tested with our target specifications.

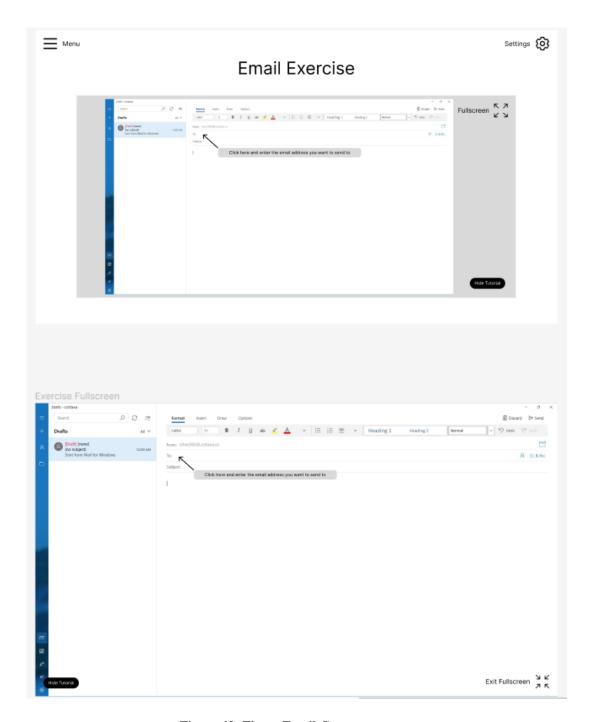


Figure 43: Figma Email Concept

Mouse Skills Email Simulator

Figure 44: Email Exercise

Call of Duty
PSST – The Free Trial's Almost Over,
Save up to 25%* on the Vault Edition

Product Documentation

49

Mouse Skills Email Simulator

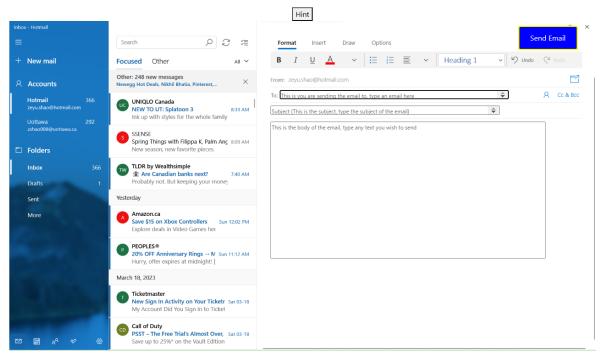


Figure 45: Email Exercise

6.5 Subsystem 5: Tutorial

6.5.1 BOM (Bill of Material)

	Bill of	material				
Item Name	Description	Unit of measure	Quantity	Unit cost	Extended Cost	Link
Cost of Labour	The cost for the team to work on this project	N/A	N/A	\$0	\$0	N/A
Bootstrap	Library to edit the display of the application	N/A	1	\$0	\$0	LINK
Vite.js	Library to develope a framework environment	N/A	1	\$0	\$0	LINK
Firebase	Library providing a framework environment	N/A	1	\$0	\$0	LINK
	Total product cost (without ta	xes or shipping)			\$0	
	Total product cost (including ta	xes and shipping)			\$0	
	List of equipment					
Item Name	Description	Туре	Prototype #	Source		
Figma	Software to create a a detailed design	Software	1	LINK		
Vue.js	To code and create the application	Software	1,2,3	LINK		
Computer	To use use the softwares in order to code	Equipment	1,2,3	LINK		
Computer mouse	The computer mouse will be used in order to test functionality	Equipment	2,3	LINK		
Visual studio code	Code editor	Software	1,2,3	LINK		
GitHub	Code distributer to other users	Software	1,2,3	LINK		

Figure 46: BOM of Subsystems

6.5.2 Equipment List

A standard-issue computer capable of running software listed above. Seek info for software specific operating requirements.

6.5.3 Instructions

Recording and editing video tutorials on how to navigate a specific website can be an effective way to teach someone how to use the website. Here are the step-by-step instructions on how to record and edit video tutorials:

Step 1: Plan your video.

Before you start recording your video, you need to plan what you want to cover. Think about the sections of the website you want to focus on and the information you want to convey. Write down a script or outline for your video, including any specific steps you want to show.

Step 2: Choose your screen recording software.

There are many screen recording software options available, both paid and free. Choose one that suits your needs and download it.

Step 3: Set up you're recording.

Open the website you want to record and open your screen recording software. Set up your recording options, such as selecting the area of your screen you want to record, the microphone input, and the recording quality.

Step 4: Start recording.

Once you've set up your recording options, start recording. Follow the script or outline you created and navigate through the website while recording. Speak clearly and explain the steps as you go.

Step 5: Edit your video.

After you've finished recording, open your video editing software. Trim any unnecessary footage, such as parts where you were searching for information, and make sure the video flows smoothly.

Step 6: Add voiceover or subtitles.

If you want to add voiceover or subtitles to your video, do so now. This can be helpful for viewers who need additional assistance or who prefer to read along with the video.

Step 7: Add titles and transitions.

Add titles to introduce each section of the video and use transitions to make the video flow smoothly between each section.

Step 8: Review and export your video.

Watch your video from start to finish to make sure everything looks and sounds good. If you're satisfied, export the video in the format you prefer, such as MP4, and upload it to your chosen platform.

In conclusion, recording and editing video tutorials on how to navigate a specific website is a great way to teach someone how to use it. By following these step-by-step instructions, you can create a high-quality video tutorial that is easy to understand and helpful for your viewers.

To implement the tutorial videos, we have used the following code:

6.6 Testing and Validation of Final Product

Initial Testing Methods and Results:

The testing of our product was conducted by each teammate, as well as friends and relatives of our teammates. The age range of testers is from 10 to 77 years old. As our client stated, the users of our product are adults, however we decided it would be best to have testers of all age ranges to get the opinion of a broader dynamic, however the opinions of our adult testers was taken into more consideration. Based on our testers experience, we asked the testers to rate target specifications, from 1-5, as our team did. We compared the results to our stated target specification stated in the previous deliverable. We also invited our testers to provide any comments about the prototype. The results are as follows:

User Interface Complexity:

Test Result: 2.5

Ideal Target Specification: 2

Marginal Target Specification: 1-3

Difficulty of Tasks:

Test Result: 3

Ideal Target Specification: 2 Marginal Target Specification: <3

Number of Unique Modules:

Test Result: 3

Ideal Target Specification: 5 Marginal Target Specification: <7

Recommended Age: Test Result: 16+

Ideal Target Specification: >18 Marginal Target Specification: >15

Some notes, quotes and comments from our testers:

- "Maybe should consider making product into a game format"
- Not exciting for children or people with low attention span
- Needs more tasks
- Good job with including all of the different mouse functions

- "Boring"
- A video tutorial would really help
- Instructions should be read to the user

After analysing our test results, our team understands that a lot of the feedback revolves around features that we plan to integrate into our future product, such as tutorials, and a "text to speech" reader. These were not possible to include in our first prototype, as our team is under time restraints. However, going forward, we will consider these results to tweak aspects such as user interface, and difficulty. Overall, we are satisfied with the findings and find them fairly consistent with our target specifications. We will use the same testing technique with future prototypes to get feedback as new features are developed.

Second-Round Testing Methods and Results:

The testing and analysis of our current product followed the same procedure as conducted last time (find previous results in Deliverable D). To recap, the testing method is as follows: The age range of testers is from 10 to 77 years old. As our client stated, the users of our product are adults, however we decided it would be best to have testers of all age ranges to get the opinion of a broader dynamic, however the opinions of our adult testers were taken into more consideration. Based on our testers experience, we asked the testers to rate target specifications, from 1-5, as our team did. We compared the results to our stated target specification stated in the previous deliverable. We also invited our testers to provide any comments about the prototype. We also ensured to use the same participants as last time to ensure the diversity dynamic is better controlled and would provide more consistent results. The results to testing are as follows:

User Interface Complexity:

Test Result: 2.1

Ideal Target Specification: 2

Marginal Target Specification: 1-3

Difficulty of Tasks: Test Result: 2.5

Ideal Target Specification: 2

Marginal Target Specification: <3

Number of Unique Modules:

Test Result: 5

Ideal Target Specification: 5 Marginal Target Specification: <7 Recommended Age: Test Result: 18+

Ideal Target Specification: >18 Marginal Target Specification: >15

Some notes, quotes and comments from our testers:

- Colours should be changed
- Needs less reading and more audio/videos
- Very big improvement compared to last testing
- Instructions should be read to the user
- Increase text size

After analysing our data, it is easy to see that our tester's input matches our target specifications closer than the results of our initial test in Deliverable D. All of our data falls within our marginal technical specifications, which deems the metrics as acceptable. Once again, the feedback from the testers revolve around features that we plan to incorporate into our final product. The main differences between the prototype from the first test include an increased amount of options for testers to use (more buttons to press, more windows, etc). Overall, we are once again satisfied with our findings.

Final Product Testing Methods and Results:

The testing and analysis of our current product followed the same procedure as conducted previously. To recap, the testing method is as follows: The age range of testers is from 10 to 77 years old. As our client stated, the users of our product are adults, however we decided it would be best to have testers of all age ranges to get the opinion of a broader dynamic, however the opinions of our adult testers was taken into more consideration. Based on our testers experience, we asked the testers to rate target specifications, from 1-5, as our team did. We compared the results to our stated target specification stated in the previous deliverable. We also invited our testers to provide any comments about the prototype.

We also ensured to use the same participants as last time to ensure the diversity dynamic is better controlled, and would provide more consistent results. The results to testing are as follows:

User Interface Complexity:

Test Result: 2

Ideal Target Specification: 2

Marginal Target Specification: 1-3

Difficulty of Tasks:

Test Result: 2

Ideal Target Specification: 2

Marginal Target Specification: <3

Number of Unique Modules:

Test Result: 3

Ideal Target Specification: 5 Marginal Target Specification: <7

Recommended Age: Test Result: 18+

Ideal Target Specification: >18 Marginal Target Specification: >15

Some notes, quotes and comments from our testers:

- Colours still should be changed
- Video tutorials are well done
- Very big improvement compared prototypes
- Increase text size

After analysing our data, the tester's input more closely matches the target specifications set previously. All of the comments regarding changes to make going forward are simple, and can be accommodated given the resources. Overall, we are once again satisfied with our findings and comfortable releasing our product in its current state.

7 Conclusions and Recommendations for Future Work

A lesson learned while prototyping the Mouse-E product is to always review the initial detailed design. Indeed, while we were prototyping, we realised that our detailed design was not fully realistic due to time constraints. Hence, we had to change our detailed design near the end date of design day. If we would have constantly focused on revaluating our detailed design, then, we would have realised that some subsystems are not possible to accomplish such as our progress tracking subsystem and our advanced difficulty subsystem. As a result, the most productive way in the future for this situation would be to always assess our concept and detailed design by having a good idea in mind of the constraints that are applied on the team.



Figure 47: First Email Exercise



Figure 48: Final Email Exercise

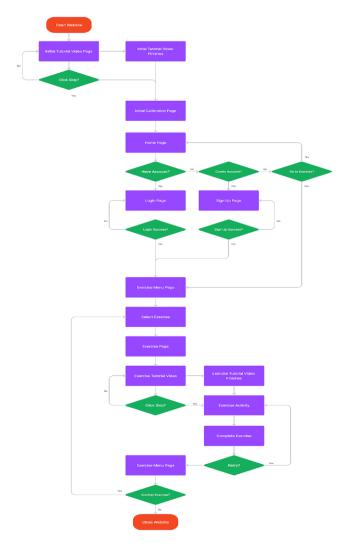


Figure 49: Detail Design

Another lesson learned throughout this project is to start learning how to code earlier. Indeed, the team started to learn how to code 1 month and a half before design day. This caused a lot of panic within the team because we were not aware of how difficult it was to code while using Vue.js. Hence, we had to learn constantly with little rest in order to have the main subsystems completed. As a result, we would recommend future students with this project or with a software project to start early in order to have time to test their prototypes more properly.

Finally, a third lesson learned by the team during the prototypes is to work through a simple prototype then upgrade on to a newer version. As a matter of fact, due to the previous issue mentioned in the second lesson learned, the team had to work straight away on building the software. However, we were thinking too big and trying to complete a fully implemented feature such as the email exercise from scratch instead of doing a small low fidelity prototype. Indeed,

this was not working because we did not complete the basics. This led to hours of time wasting because we were trying to complete tasks that needed to be done in multiple steps into one step. As a result, it was just not working while prototyping. Hence, we had to create a low fidelity prototype and then build on it to complete the final email prototype as shown in figure 47 and 48 Overall, the lesson learned is to start with low fidelity prototypes and then upgrade those prototypes to higher levels of fidelity. For the upcoming future students, the team would recommend them to start from something small and to continuously update it.

In conclusion, if the team had more time to work on this project, we would update our exercises and implement newer exercises. Indeed, we believe that having more exercises would benefit the users since they will have more opportunities to practise the basic mouse skills into different environments such as a YouTube exercise or a Disney+ exercise. The team believes it would be essential to work more on those exercises because the client told us during design day that having more exercises similar to the email exercise would definitely benefit the users. After completing the exercise, we can then start working on a functional login system that would allow the users to track their progress. Hence, due to time constraints, the team was not able to complete the progress tracking subsystem and the advanced difficulty subsystem. Those subsystems were unfortunately left abandoned and as a team we felt that they were important because it would enhance a better user experience while experimenting with our product.

8 Bibliography

Minimouse. (n.d.). Minimouse. [Website]. Retrieved January 18, 2023, from https://www.minimouse.us/index.html

Dailymail.com Reporter. (2015, August 22). *Solitaire and minesweeper were created to trick you into learning things*. Daily Mail Online. Retrieved January 29, 2023, from https://www.dailymail.co.uk/news/article-3207153/Solitaire-Minesweeper-created-trick-learning-things-computers.html

Aim Lab. (n.d.). Aim Lab on Steam. [Website]. Retrieved January 18, 2023, from https://store.steampowered.com/app/714010/Aim_Lab/

Aim Lab. (n.d.). Aim Lab on Steam. [Website]. Retrieved January 18, 2023, from https://store.steampowered.com/app/714010/Aim_Lab/.

Mouse Accuracy. (n.d.). Mouse Accuracy. [Website]. Retrieved January 18, 2023, from https://mouseaccuracy.com.

Thinkful. (2021, September 9). How long does it take to learn coding? Thinkful. https://www.thinkful.com/blog/how-long-does-it-take-to-learn-coding/

<u>Link for Prototype On Figma: https://www.figma.com/proto/MaVNLJsS7EUlsQQRfFIsbI/GNG-2101-Project-Wireframe?node-id=2%3A12&starting-point-node-id=2%3A12</u>

Moss, R. (2021, March 18). Short educational videos for students are better for learning. Boclips. https://www.boclips.com/blog/short-educational-videos-for-students-are-better-for-learning

BoIA. (2019, May 16). Why Do I Need Both Transcripts and Captions for Accessibility? BoIA - Making Websites Accessible. https://www.boia.org/blog/why-do-i-need-both-transcripts-and-captions-for-accessibility.

CUNY Libraries. (n.d.). Why it Matters - Accessibility and Universal Design - Research Guides at CUNY. CUNY Research Guides. Retrieved January 23, 2023, from https://guides.cuny.edu/accessibility/whyitmatters.

Bibliography 60

APPENDICES

9 APPENDIX I: Design Files

Table 9. Referenced Documents

Document Name	Document Location and/or URL	Issuance Date
Deliverable B	<u>Link</u>	2023-01-29
Deliverable C	<u>Link</u>	2023-02-09
Deliverable D	<u>Link</u>	2023-02-18
Figma design concept	<u>Link</u>	2023-02-18
Deliverable B	<u>Link</u>	2023-01-29
MakerRepo	<u>Link</u>	2023-04-10

10 APPENDIX II: Other Appendices

The link of the website: https://zeyushao-uottawa.github.io/Mouse-Skills/
Below were concept ideas that lead towards the building of our product:

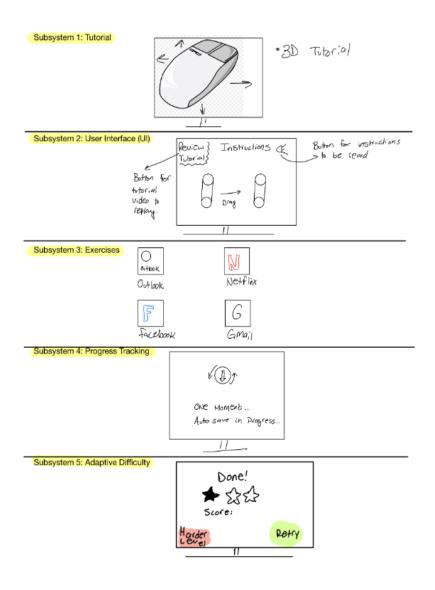


Figure 50: Concept 1

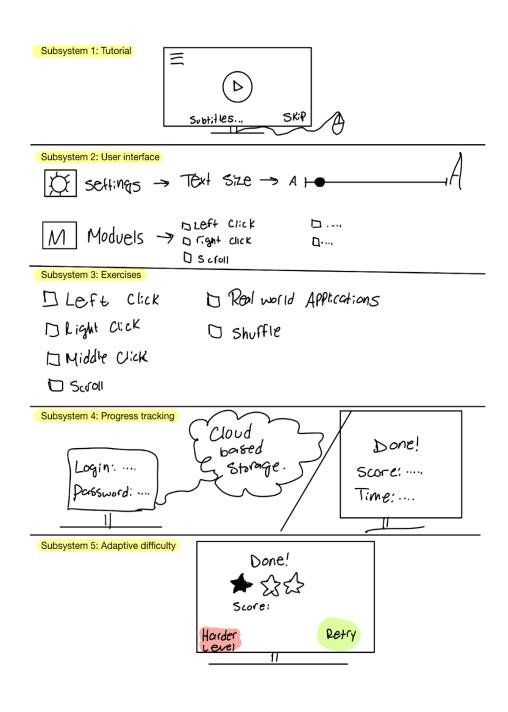


Figure 51: Concept 2

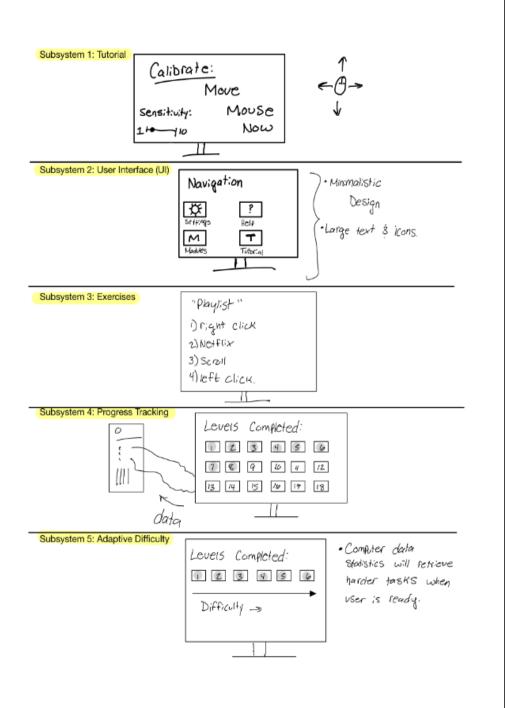


Figure 52: Concept 3