

Title : IVIP Group Assignment

Intake : UCDF2309ICT(SE)

Level of Study : Diploma Semester 1

Lecturer's Name : Ms. Yogeswari A/P Muthusamy

Group Members :

NO	Name	Student ID
1	Lim Chee Xuan	TP 075916
2	Paureen Tan Nie Nie	TP 075914
3	Phang Shea Wen	TP 075813
4	Sarah Manisha Oorjitham	TP 078367

Table of Contents

0.0 Proposal (Modified)	4
0.1 Introduction.....	4
0.2 Stage 1 – Analysis.....	4
0.3 Objective of the application	5
0.4 Target Users	5
0.5 Hardware, Software, Skills Required	5
0.6 Gantt Chart for Application Development.....	6
0.7 References.....	6
1.0 Introduction.....	7
2.0 Stage 2: Design	8
2.1 Functionality of Each Sprite	8
2.2 Design flowchart.....	24
2.3 Storyboard of the application	25
3.0 Stage 3: Development	28
3.1 Programming concepts.....	28
3.1.1 Opening Animation.....	28
3.1.2 Starting of the Virtual Tour.....	34
3.1.3 Level 1 of the Library	37
3.1.4 Level 2 of the Library	49

3.1.5 Level 3 of the Library	58
3.1.6 Throughout the Virtual Tour.....	72
3.1.7 Rating of the Application	85
3.1.8 Ending Animation.....	96
3.1.9 Custom Block Used Throughout the Application	97
3.1.10 Variables Used Throughout the Application.....	132
3.2 Backdrops functionality used in the application.....	136
4.0 Stage 4: Implementation.....	140
4.1 Application user manual	140
4.2 Screenshots of sample input/output and explanation	143
4.2.1 Input in the Application	143
4.2.2 Output in the Application.....	145
4.3 Testing results.....	147
5.0 Conclusion	151
6.0 References.....	153

0.0 Proposal (Modified)

0.1 Introduction

“Book Venture” introduces the new 3-tier library and provides guidance to the users on how to use the features and facilities in the library. The application will be showcased at the library's grand opening day to inform students and lecturers about the materials, facilities and services offered by the School of Computing. “Book Venture” aims to redefine how individuals interact with and utilize library facilities.

“Book Venture” is basically an application designed to expedite and enhance a user’s experience of the newly opened library. A virtual tour of the library’s design and layout, an information kiosk, as well as a robot assistant is showcased in the features of this application. The new 3-tier library consists of a cafe, information counter as well as information kiosk on the first floor; discussion rooms and quiet zone on the second floor and the library with books and study materials as well as a computer zone is located on the third floor. Through a virtual tour, “Book Venture” offers a simulated walk-through of the library, enabling users to explore and better understand the layout and amenities of the new 3-tier library, as well as the intended purpose for each section.

0.2 Stage 1 – Analysis

To build this application, we have done some research on the current library at Asia Pacific University, which is a prestigious School of Computing in Malaysia. Through our investigation of the library, we spotted multiple features that can be integrated into our new library. An incredible structure of the library is that it provides several discussion rooms, allowing both students and lecturers to conduct physical meetings in the library. Televisions are equipped in each room, which enables users to present their materials on screen and open for discussion. Since discussion rooms are individual and separated, both lecturers and students are able to have private discussions without concern of others listening to or overhearing their discussion. Apart from that, APU has lived up to its name as a computing school by providing computer facilities in their library. The availability of computers in the library increases the accessibility to online information for lecturers and students, thereby facilitating the learning process as information

can be found easily. Besides, APU also supplied their students with free Wi-Fi connections, thus clearing delays in searching for information through the computers in the library.

0.3 Objective of the application

The objective of this application is to deliver clear and comprehensive information about the new library, its facilities and amenities, as well as where each facility is located, to both the students and the lecturers. A concise briefing about the library is included in the application, as well as the features and the conveniences it brings. The application aims to provide detailed information about each feature and amenity available in the new library, as well as explain to the students and lecturers how they may fully take advantage of all the features available. Henceforth, students and lecturers could have a clearer image on how to utilize the library efficiently once the library is fully furnished and opened for use. Furthermore, this application aims to promote a better learning environment to everyone who uses the library, enhancing the user experience in the library.

0.4 Target Users

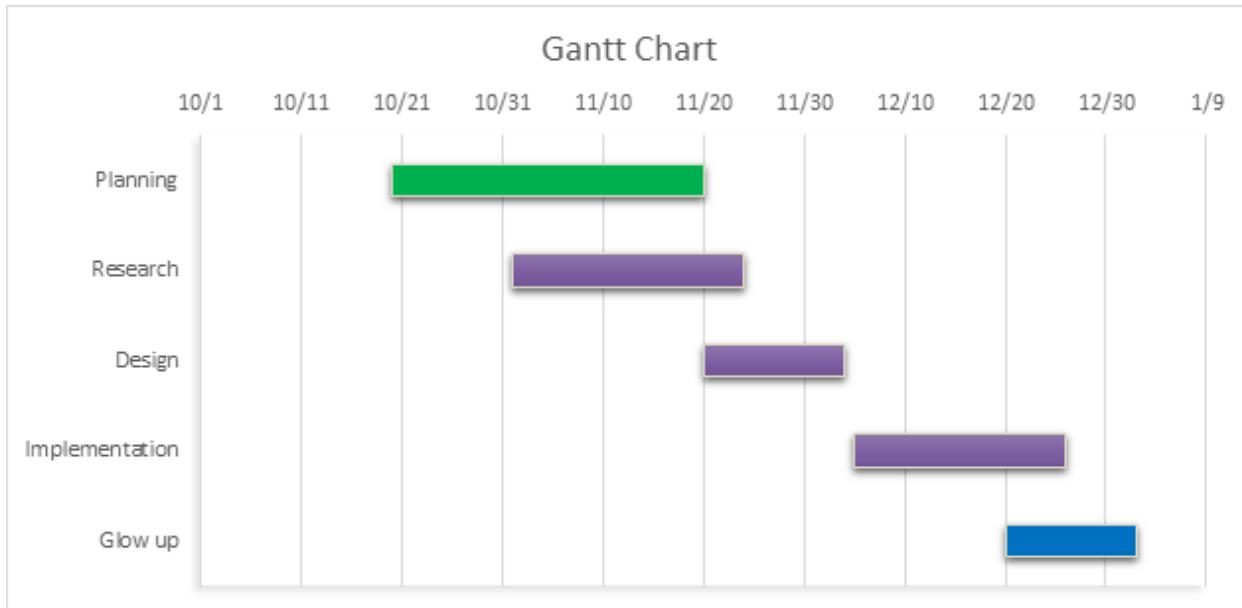
“Book Venture” is an application designed to empower a varied user base within the APU University community. The target users of this application are APU University students and prospective students interested in Computing and Technology. “Book Venture” is designed to fulfill students’ requirements and improve their educational journey. This application is a useful and valuable resource for anyone working on assignments at the undergraduate level, doing research as a postgraduate student or just prospective students exploring the library on campus. Beyond students, “Book Venture” can be used by APU staff and librarians for a more organized and efficient library management experience. This application can also assist students and lecturers in understanding the materials and facilities provided by the library, as well as where it is located.

0.5 Hardware, Software, Skills Required

“Book Venture” is designed to be accessible across a wide range of devices, ensuring that users have the flexibility to interact with it in the way that best suits their needs. For hardware, this

application can be accessible on computers, tablets, smartphones, and also interactive kiosks in the library. “Book Venture” is built using Snap! Programming environment, a user-friendly and flexible platform known for its visual programming blocks. The skills required in this application is proficiency in Snap! Programming. It is essential for the development of application.

0.6 Gantt Chart for Application Development



0.7 References

- APU Library Website. <https://library.apu.edu.my/>
- Qois, Z. A., & Wijayanti, L. (2021). The Effect of Library Virtual Tour on Library Image Construction: Study on Perpustakaan BPK RI. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 13133 LNCS. https://doi.org/10.1007/978-3-030-91669-5_16
- Shi, X., Tang, K. and Lu, H. (2021), "Smart library book sorting application with intelligence computer vision technology", *Library Hi Tech*, 39(1). 220-232.
<https://doi.org/10.1108/LHT-10-2019-0211>

1.0 Introduction

The team has developed an interactive and engaging application that is orientated to communicate an immersive as well as indulging virtual tour around the newly opened library to the future users during the library's grand opening day through Snap!, which is a visual programming language that is designated to manipulate visual elements throughout the program development process. As a 3-tier library building under the School of Computing, the library is complemented with additional areas comprising of an ambient cafe, individual discussion rooms, quite zone, and computing area. Through the virtual exploration around the library provided by this proposed application, users will be taking in a concise introduction to the library and will have the exposure to the facilities and services available in the library. Henceforth, users will not be left unguided and lost during the occasion of the library's grand opening day.

Upon ending the virtual tour, users are prompted to leave a rating for the overall experience of using the application, a rating about the new library, and the new amenities. This enables the team to acknowledge the performance of the application developed, and the school could have a clear knowing of whether the main objective of building this new library is achieved through the rating provided that represents the satisfaction of users towards it.

To hand out a thorough and deepening comprehension of the entire program's operating mechanism, this documentation is written as a systematic and organized tracking approach and record of the flow and steps involved in the application development process, as well as a breakdown analysis of the code written to build the proposed application. The specification on the elements involved in the coding procedure such as the functionality of each sprite, programming concepts, functionality of each backdrop, sample input and output, and testing results are documented. Furthermore, pseudocode of the entire code that is written beforehand which outlines the logic of the programming and clarifies the programming steps is also included in the documentation.

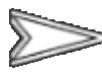
2.0 Stage 2: Design

2.1 Functionality of Each Sprite

- a) Each sprite's functionality used in the application.

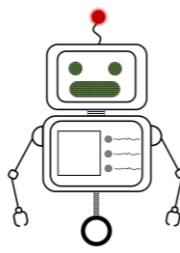
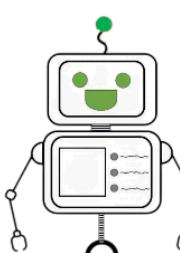
Table 2.1.1 Functionality of Sprite Involved in the Beginning of the Application

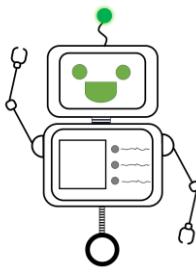
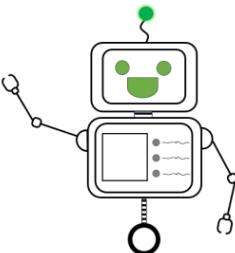
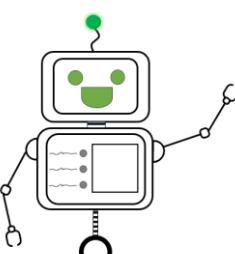
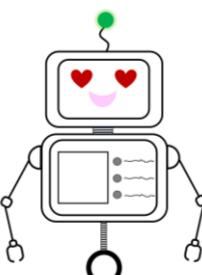
Beginning of the Application		
Sprite Name	Sprite/Costume	Functionality
Opening	glitch_1 	This sprite is an opening animation that greets the user to the new library with a glitch effect. It appears only at the beginning of the application.
	glitch_2 	This sprite has 6 costumes, namely “glitch_1”, “glitch_2”, “glitch_3”, “glitch_4”, “glitch_5”, and “glitch_6”.
	glitch_3 	To showcase a glitching effect, this sprite will switch costumes between all its 6 costumes for a specified number of repetitions.
	glitch_4 	
	glitch_5 	
	glitch_6 	

Door		This sprite is used to draw the outline of the door, that act as the entrance of the “Robot” sprite. It appears only once at the beginning of the application right after the “Opening” sprite finish running its script.
Write_book_venture		This sprite is used to write “SAY HELLO TO BOOK VENTURE” to introduce users to our application’s name, Book Venture. It appears only once and runs at the same time as the “Door” sprite runs its script.
Grey_thing		This sprite is used to make the scene visually appealing by blocking the text of instructions “Please select the level”, written by the sprite “Level”, after the user clicks on the robot.
Library_intro		This sprite is used to draw the outline of a 3-tier building which displays a simple 3D model of the building. It functions as a platform to showcase the number of levels written by the “Level” sprite and buttons representative sprites including the “Button_level_1”, “Button_level_2” and “Button_level_3” sprites on it. This sprite appears only once in the application, it acts as the sprite that firstly introduce the 3D model of the building to the users. The following displaying of the 3D model of

		the building will be taken over by its successive sprite, “Library” sprite.
--	--	---

Table 2.1.2 Functionality of Sprite Involved in Throughout the Application

Throughout the Application		
Sprite Name	Sprite/Costume	Functionality
Library		This sprite is a successive sprite of the “Library_info” sprite, with its script shorten and simplified. Therefore, when user is redirected back to the view and select the level of the library, it will not repeat the entire script of that sprite over again and will make the outcome less tedious, particularly considering to it being frequently called to run the script.
Robot	robot_stop 	This sprite act as a virtual assistant that guide users throughout the application. It initially acts as a button that initiates the greeting procedure when clicked at the beginning of the application and prompting users to input their names. It is programmed to be clickable only once. Henceforth, when it is clicked at any point of the application after the initial click on it at the start, it will not greet the users repeatedly upon clicking it.
	robot_start 	
	robot_wave	

		<p>This sprite has 6 costumes, namely “robot_stop”, “robot_start”, “robot_wave”, “robot_wave_2”, “robot_wave_3”, and “robot_love”.</p>
		<p>At the start of the application as the “door” sprite and “write_book_venture” runs its script, it changes its costume from “robot_stop”. When it is clicked by the user, it changes its costume to “robot_start” indicating that the robot is activated. Upon activating the robot, the robot greets the users by switching its costume between “robot_wave”, “robot_wave_2”, and “robot_start”, resulting in the robot waving its hand. While displaying the 3D model of the building, it switches costumes from “robot start” to “robot_wave_3” and back to “robot_start”, leading to the robot lifting its left hand which delivers an illusion that the 3D model of the building appears from its hand and later on putting its hand down once the entire 3D model of the building is fully displayed by the “Library_intro” sprite.</p>
		<p>At the cafe of level 1 of the library tour, it will switch costume from “robot_start” to “robot_love” once “Cafe_icon” sprite writes the end of its statement “You can have a cup of coffee here”.</p>
		

Level		<p>This sprite is used to write “Level 1”, “Level 2”, and “Level 3”, which are the number of levels on each floor within the library; and used to write out the instruction to ask users to click on a particular level they wanted to visit. At the beginning of the application, it will run its script at the same time as the button representative sprites including the “Button_level_1”, “Button_level_2”, and “Button_level_3” sprite which follows up to the script run by the “Library_intro” sprite. After that, it will run its script together with the button representative sprite right after the “Library” sprite run its script, and it only appears every time it is called by the users’ decision to be redirected back to view and select the level of the library by clicking the “Back” sprite.</p>				
Button_level_1	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px; text-align: center;">unselected</td> <td data-bbox="665 1353 739 1417">  </td> </tr> <tr> <td style="padding: 5px; text-align: center;">selected</td> <td data-bbox="665 1480 739 1543">  </td> </tr> </table>	unselected		selected		<p>This sprite represents a clickable button which directs users to level 1 within the library. At the beginning of the application, it will run its script at the same time as the “Level” sprite that follows up to the script run by the “Library_intro” sprite. After that, it will run its script in synchronization with the “Level” sprite right after the “Library” sprite run its script and it only appears every time it is called by the users’ decision to be redirected back to view and</p>
unselected						
selected						

		<p>select the level of the library by clicking the “Back” sprite.</p> <p>When displaying the 3D model of the building and this sprite shown itself on the 3D model drawn by the “Library_intro” or “Library” sprite, it will change costume to “Selected” as the mouse hovered over it and change costume to “Unselected” as the mouse departed from it.</p>
Button_level_2	<p>unselected</p>  <p>selected</p> 	<p>This sprite represents a clickable button which directs users to level 2 within the library. At the beginning of the application, it will run its script at the same time as the “Level” sprite that follows up to the script run by the “Library_intro” sprite. After that, it will run its script in synchronization with the “Level” sprite right after the “Library” sprite run its script and it only appears every time it is called by the users’ decision to be redirected back to view and select the level of the library by clicking the “Back” sprite.</p> <p>When displaying the 3D model of the building and this sprite shown itself on the 3D model drawn by the “Library_intro” or “Library” sprite, it will change costume to “Selected” as the mouse hovered over it or clicked it, and change costume to</p>

		“Unselected” as the mouse departed from it.
Button_level_3	<p>unselected </p> <p>selected </p>	<p>This sprite represents a clickable button which directs users to level 3 within the library. At the beginning of the application, it will run its script at the same time as the “Level” sprite that follows up to the script run by the “Library_intro” sprite. After that, it will run its script in synchronization with the “Level” sprite right after the “Library” sprite run its script and it only appears every time it is called by the users’ decision to be redirected back to view and select the level of the library by clicking the “Back” sprite.</p> <p>When displaying the 3D model of the building and this sprite shown itself on the 3D model drawn by the “Library_intro” or “Library” sprite, it will change costume to “Selected” as the mouse hovered over it and change costume to “Unselected” as the mouse departed from it.</p>
Write_back		This sprite writes out the function of the sprite “Back” to inform users that they can be directed back to the 3D model of the library by clicking the “Back” sprite. It runs its script in alignment with the “Back” sprite. It is shown in the application each

		time to inform users that they are provided with the option to go back to the previous scene.
Back	back 	This sprite represents a clickable button which directs users back to the previous scene. It runs its script in alignment with the “Write_back” sprite. It is shown in the application each time to provide users with the option to be redirected back to the previous scene by clicking on it.
Arrow	arrow_1 	This sprite is used to provide demonstration to users by pointing towards the object mentioned in the statement written by other sprites.
	arrow_2 	This sprite has 2 costumes, namely “arrow_1” and “arrow_2”. As this sprite appears on the stage and pointing towards a particular object, it will be switching costumes between “arrow_1” and “arrow_2” to enhance the visualization and presentation of the arrow in order to grab users’ attention towards the object.
Write_click_icon		This sprite is used to write “Please click on the icon” on stage to inform users to click on any of the icon displayed in order to

		view a zoomed in image of the facilities. It appears in level 1, level 2, and level 3 of the library.
--	--	---

Table 2.1.3 Functionality of Sprite Involved in the Level 1 of the Library of the Application

Level 1 of the Library in the Application		
Sprite Name	Sprite/Costume	Functionality
Cafe_icon	cafe_icon_1 	This sprite guides users to the cafe in Level 1 of the library when clicked. It will show itself when users selected to view Level 1 of the library by clicking the “Button_level_1” sprite displayed on the 3D model of the library.
	cafe_icon_2 	This sprite has 2 costumes, namely “cafe_icon_1” and “cafe_icon_2”. At level 1 of the library tour, as users use their mouse to hover over or click on this sprite, it will switch costume to “cafe_icon_2”. When users’ mouse is departed from it, it will switch costume to “cafe_icon_1”.
Information_kiosk_icon	touch_icon_1	

	 touch_icon_2 	<p>This sprite guides users to the information kiosk in Level 1 of the library when clicked. It will show itself when users selected to view Level 1 of the library by clicking the “Button_level_1” sprite displayed on the 3D model of the library.</p> <p>This sprite has 2 costumes, namely “touch_icon_1” and “touch_icon_2”.</p> <p>At level 1 of the library tour, as users use their mouse to hover over or click on this sprite, it will switch costume to “touch_icon_2”. When users’ mouse is departed from it, it will switch costume to “touch_icon_1”.</p>
Counter_icon	 counter_icon_1  counter_icon_2	<p>This sprite guides users to the counter in Level 1 of the library when clicked. It will show itself when users selected to view Level 1 of the library by clicking the “Button_level_1” sprite displayed on the 3D model of the library.</p> <p>This sprite has 2 costumes, namely “counter_icon_1” and “counter_icon_2”.</p> <p>At level 1 of the library tour, as users use their mouse to hover over or click on this sprite, it will switch costume to “counter_icon_2”. When users’ mouse is</p>

		departed from it, it will switch costume to “counter_icon_1”.
--	--	---

Table 2.1.4 Functionality of Sprite Involved in the Level 2 of the Library of the Application

Level 2 of the Library in the Application		
Sprite Name	Sprite/Costume	Functionality
Discussion_room_icon	 	<p>This sprite guides users to the discussion room in Level 2 of the library when clicked. It will show itself when users selected to view Level 2 of the library by clicking the “Button_level_2” sprite displayed on the 3D model of the library.</p> <p>This sprite has 2 costumes, namely “discussion_room_icon_1” and “discussion_room_icon_2”.</p> <p>At level 2 of the library tour, as users use their mouse to hover over or click on this sprite, it will switch costume to “discussion_room_icon_2”. When users’ mouse is departed from it, it will switch costume to “discussion_room_icon_1”.</p>
Quiet_zone_icon		<p>This sprite guides users to the quiet zone in Level 2 of the library when clicked. It will show itself when users selected to view Level 2 of the library by clicking the</p>

		<p>“Button_level_2” sprite displayed on the 3D model of the library.</p> <p>This sprite has 2 costumes, namely “quiet_zone_icon_1” and “quiet_zone_icon_2”.</p> <p>At level 2 of the library tour, as users use their mouse to hover over or click on this sprite, it will switch costume to “quiet_zone_icon_2”. When users’ mouse is departed from it, it will switch costume to “quiet_zone_icon_1”.</p>
--	---	---

Table 2.1.5 Functionality of Sprite Involved in the Level 3 of the Library of the Application

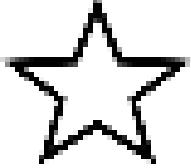
Level 3 of the Library in the Application		
Sprite Name	Sprite/Costume	Functionality
Book_shelf_icon	book_shelf_icon_1 	<p>This sprite guides users to the bookshelf in Level 3 of the library when clicked. It will show itself when users selected to view Level 3 of the library by clicking the “Button_level_3” sprite displayed on the 3D model of the library.</p>
	book_shelf_icon_2 	<p>This sprite has 2 costumes, namely “book_shelf_icon_1” and “book_shelf_icon_2”.</p>

		At level 3 of the library tour, as users use their mouse to hover over or click on this sprite, it will switch costume to “book_shelf_icon_2”. When users’ mouse is departed from it, it will switch costume to “book_shelf_icon_1”.
Computer_icon	computer_icon_1 	This sprite guides users to the computer zone in Level 3 of the library when clicked. It will show itself when users selected to view Level 3 of the library by clicking the “Button_level_3” sprite displayed on the 3D model of the library.
	computer_icon_2 	This sprite has 2 costumes, namely “computer_icon_1” and “computer_icon_2”. At level 3 of the library tour, as users use their mouse to hover over or click on this sprite, it will switch costume to “computer_icon_2”. When users’ mouse is departed from it, it will switch costume to “computer_icon_1”.
Notepad		This sprite allows users to open a notepad tab by clicking on it. It is shown when users clicked on the “Computer_icon” sprite that will direct users to a computer screen.
Close_tab		This sprite is used to close the notepad tab, returning users to the computer home

		screen. It is shown when users clicked on the “Notepad” sprite after clicking on the “Computer_icon” sprite that will direct users to a computer screen.
Write_notepad		This sprite is used for writing words in the notepad, allowing users to interact with it by typing on their keyboard. Users can input any keys to interact with the notepad. It runs its script when users clicked on the “Notepad” sprite after clicking on the “Computer_icon” sprite that will direct users to a computer screen.

Table 2.1.6 Functionality of Sprite Involved in the Rating of the Application

Rating of the Application		
Sprite Name	Sprite/Costume	Functionality
End_tour	end_tour 	This sprite acts as a clickable button that gives users the option to end the virtual tour. It runs its script at the same time as the “Library_intro” sprite and “Library” sprite. It is displayed on the stage when the 3D model of the library is shown on the stage that opens selection of level of the virtual tour to users. Thus, when users have completed the virtual tour, they can simply click on this sprite to end it.

star_1	star_unselected 	These sprites are used as one of the stars provided to the users to give rating for the three following areas, which are the application, library, and facilities. It only appears once the users decided to end the virtual tour by clicking the “End_tour” sprite. This sprite runs its script at the same time as the “Review” sprite.
star_2	star_selected 	These sprites all have two costumes, namely “star_unselected” and “star_selected”.
star_3		
star_4		It initially appears in costume “star_unselected”. When users’ mouse hovers over it, it switched costume to “star_selected”, and switches back as the mouse is departed from it. When users clicked on it, it will switch to costume “star_selected” forever, indicating that the rating is made. As users click on it again to remove the rating, it will switch back to costume “star_unselected”.
star_5		Among these 5 sprites, “Star_5” sprite has additional responsibility of calling the “Submit_rating” sprite to appear on stage so that users can submit the final rating for the respective areas.

Review	star 	This sprite is used to write the three areas for users to rate on and show the overall rating which is the average rating for the three areas calculated by the “Submit_rating” sprite. This sprite is also responsible for activating the rating representative sprites, which are “star_1”, “star_2”, “star_3”, “star_4”, and “star_5” to appear on the stage.
Submit_rating	submit_rating 	This sprite is used to collect the ratings given by the users for the three respective areas written by the “Review” sprite and calculate the overall average rating. The overall average rating will then be displayed by the “Review” sprite.

Table 2.1.7 Functionality of Sprite Involved in the Ending of the Application

Ending of the Application		
Sprite Name	Sprite/Costume	Functionality
Ending		This sprite is an ending animation that thanks the users who used this application and shows the name of members that were involved in creating it. It appears only at the end of the application.

2.2 Design flowchart

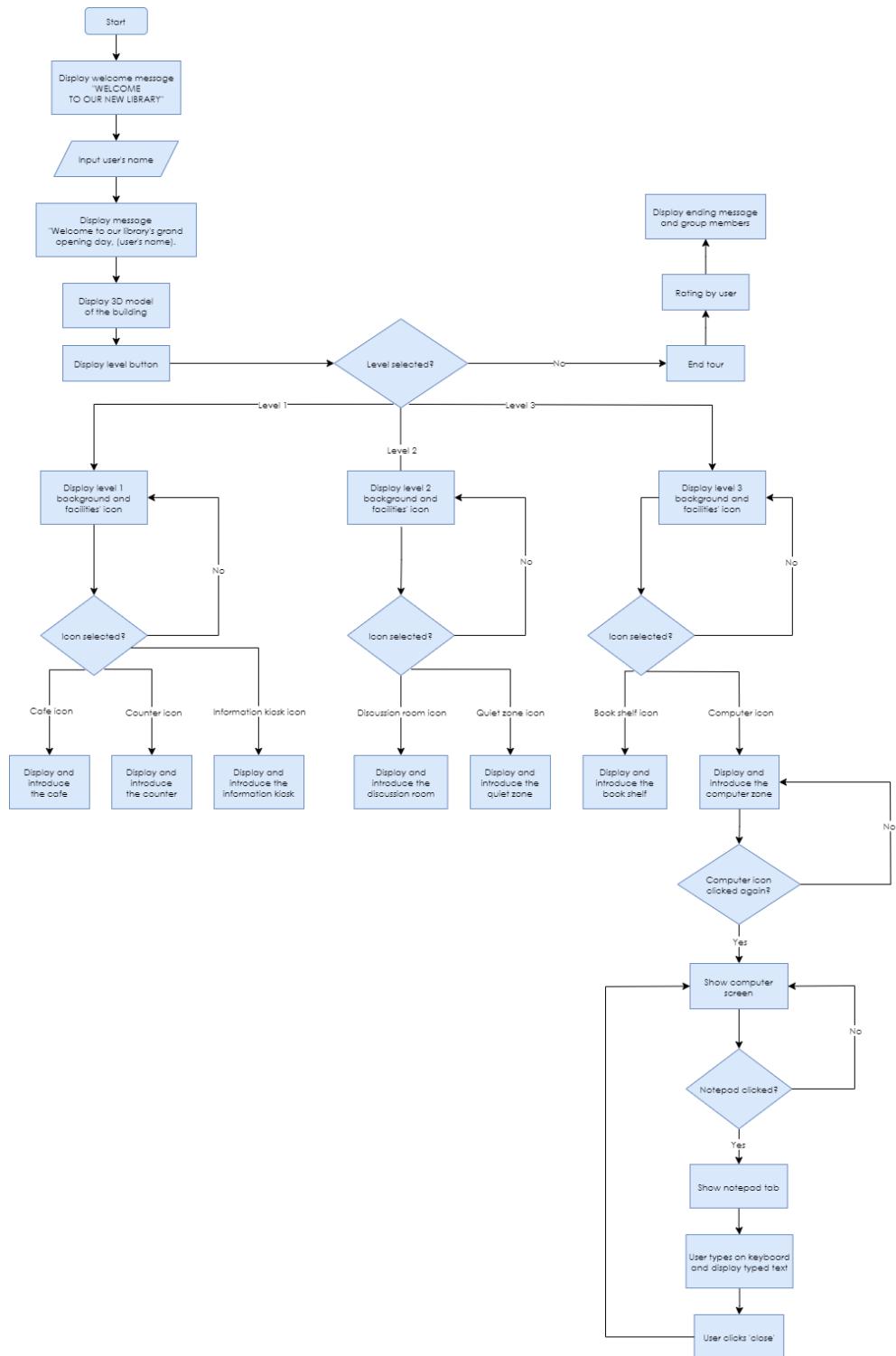


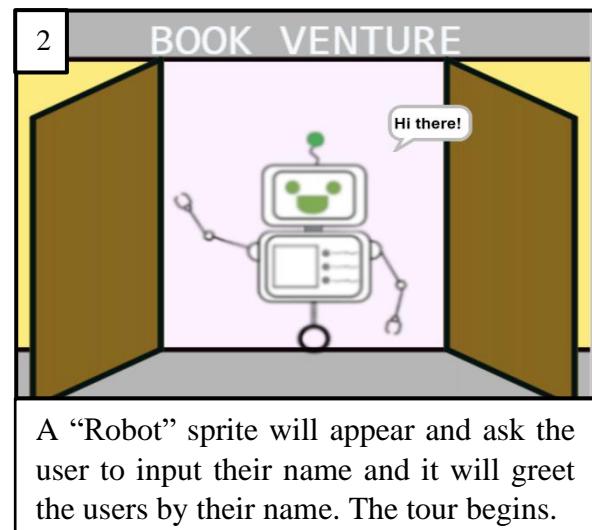
Figure 2.2 Flowchart of the Virtual Tour Application of the School of Computing

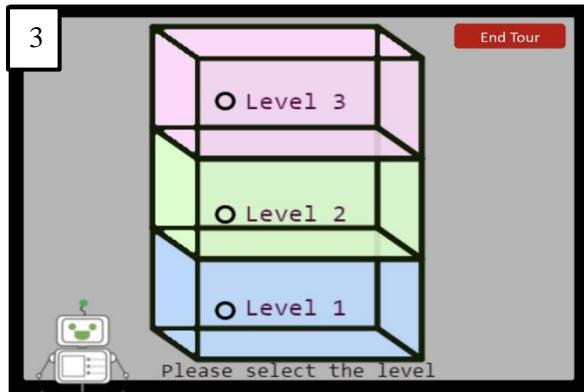
2.3 Storyboard of the application

The storyboard provides the team with an overview and narrative of the virtual tour application for the new library of The School of Computing in Asia Pacific University. There are two parts to the application for the new library; the first part is a tour of every level of the new library. In this part, the user can have a look at the facilities or rooms for each level in the library. When the facilities are clicked by the user, there will be a robot existing to explain the uses of the facilities or rooms of the library.

In the second part of the application, the user would take part in a rating session to evaluate the performance of the new library and the Virtual Tour. The question would be asked in the rating session include:

1. How's the new library "BOOK VENTURE"?
2. How's the virtual tour?
3. How's the facilities?





There will be a 3D building model and three levels to let the user choose which of the levels they want to visit. If the user has visited all the levels in the application, user can click on the end tour icon to end the virtual tour.



For the level 1, an instruction will appear asking user to click the icon that will introduce the facilities in the level 1. There is a back icon that allows user to be redirected back to the previous scene for level selection.



While the user is at level 2, an instruction will appear asking user to click the icon that will introduce the rooms in the level 2. There also will have the back icon to exit the scenario of level 2.



There also will have an instruction to the users for the purpose of clicking the icon that will introduce the facilities in level 3. User can click on the back icon to return to the level selection page.

7

How's the new library "BOOK
VENTURE" ?



Submit

After the user has decided to end the tour, there would have a rating session to give rating toward satisfaction about the performance of the new library and the Virtual Tour.

3.0 Stage 3: Development

3.1 Programming concepts

3.1.1 Opening Animation

Sprite Involved 1: “Opening” Sprite

To initiate the application, it is mandatory for users to click on the flag button .

Table 3.1.1.1 Programming Concept of “Opening” Sprite in Opening Animation

Code	Concept
	When flag button  is clicked, it will stop all other script from running except for this script that is currently running. This step is necessary in certain circumstances where users have the intention to click on the flag button  again to restart the application although it is halfway running. This will stop continuation of previously running script and avoid conflict with the currently running script on the stage. Before broadcasting “opening” to trigger the following script, “Opening” sprite is hidden and sent to a specific location on the stage. Ghost effect is reset to 0, ensuring that the sprite will appear when it is shown in the second script. “Opening” is also broadcasted to the “Stage” so that it can switch the background costume to “Black” and play the sound “opening”.
	When the second script received “opening” broadcasted, “Opening” sprite will show itself and switch costume to “glitch_1” to display “WELCOME TO OUR NEW LIBRARY” on the stage. “Opening” sprite glides to the center of the stage, and switch costumes among “glitch_1”, “glitch_2”, “glitch_3”, “glitch_4”, “glitch_5”, and “glitch_6” in sequence through “Glitch” custom Looks block . A streamlined and fluid glitching effect of “WELCOME TO OUR NEW LIBRARY” can be showcased on stage. Once the

	<p><u>“Glitch” custom Looks block</u> finish running, the ghost effect of the “Opening” sprite which is initially set to 0 is increased by 2 for 50 repetitions, which at the end results in a value of 100 of the ghost effect. The “Opening” sprite will gradually fades away. Upon completion of the final script for the “Opening” sprite, the subsequent script run by “Door” sprite and another script run by “Write_book_venture: will be activated to present the next robot entrance scene by broadcasting “door”. “Door” will also be broadcasted to the “Stage” in order to switch the background costume to “Grey” and play the sound “door_opening”.</p>
--	---

Sprite Involved 2: “Door” Sprite

Table 3.1.1.2 Programming Concept of “Door” Sprite in Opening Animation

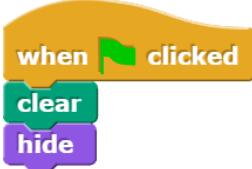
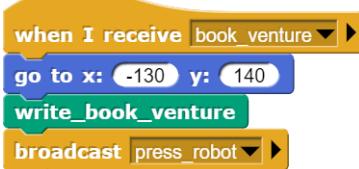
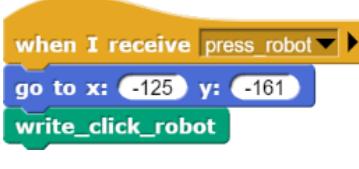
Code	Concept
<pre>when green flag clicked clear hide</pre>	<p>When flag button is clicked, all drawing done on the stage previously will be cleared. Hence, when users click on the flag button again, the previous drawing drawn will not remain on the stage, disrupting the overall presentation of the opening animation.</p>



When the second script receives “door” broadcasted by the “Opening” sprite, “Door” sprite will hide itself and draw the outline of a wall through “[draw_outline](#)” custom Looks block. A variable named “[angle](#)” is created, which represent the angles involved in drawing the outline of the door. The value of “[angle](#)” variable is initialized to 90. “Door” sprite continues to draw the outline of the left door through “[draw_door_left](#)” custom Looks block as well as the outline of the right door through “[draw_door_right](#)” custom Looks block. Once the door and the wall are drawn, “Door” sprite will fill the wall with █ colour via “[fill_wall](#)” custom Pen block. Both the left and right door are █ coloured via their respective custom Pen blocks. After 3 seconds, “Door” sprite enters a loop that iterates until the “[angle](#)” variable is greater than 240. In each loop, all drawings on the stage will be cleared. Outline of the wall is drawn via “[draw_outline](#)” custom Looks block and colouration of the wall is done via the “[fill_wall](#)” custom Pen block. Value of “[angle](#)” variable increases by 3. After that, the interior which is referring to the room behind the walls is painted with █ colour via the “[paint_interior](#)” custom Pen block. The outline of both the left and right door is drawn at a new angle, which is the original angle 90° increase by 3° , and the colouration of the door is also done via their respective custom Pen block. The repetition of each loop will bring out a door opening motion. Once the loop ends, “Door” sprite will broadcast “book_venture” to “Write_book_venture” sprite to write the relevant information and will broadcast to “Robot” sprite so that it will show up at the middle of the door.

Sprite Involved 3: “Write_book_venture” Sprite

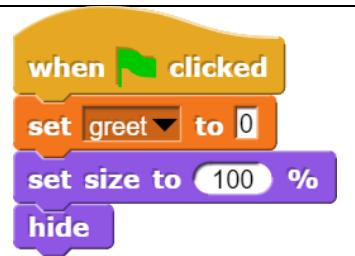
Table 3.1.1.3 Programming Concept of “Write_book_venture” Sprite in Opening Animation

Code	Concept
	When flag button  is clicked, all drawing done on the stage previously will be cleared. Hence, when users click on the flag button  again, the previous drawing drawn will not remain on the stage, disrupting the overall presentation of the opening animation.
	When the second script receives “door” broadcasted by the “Opening Sprite”, “Write_book_venture” sprite will hide itself and go to the specified location which is on top of the door drawn by the “Door” sprite and write “SAY HELLO TO” via the “write_hello” custom Pen block .
	When the third script receives “book_venture” broadcasted by the “Door” sprite, “Write_book_venture” sprite will move to the specified location which is on top of the door drawn by the “Door” sprite and write “BOOK VENTURE” via the “write_book_venture” custom Pen block . Once code in “write_book_venture” block is completed, “press_robot” is broadcasted to the next script.
	When the fourth script receives “press_robot” broadcasted from the previous script, “Write_book_venture” sprite will move to the given location which is at the bottom of the door drawn by the “Door” sprite and write “Please click the robot”.

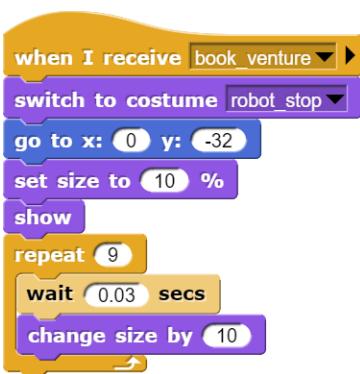
Sprite Involved 4: “Robot” Sprite

Table 3.1.1.4 Programming Concept of “Robot” Sprite in Opening Animation

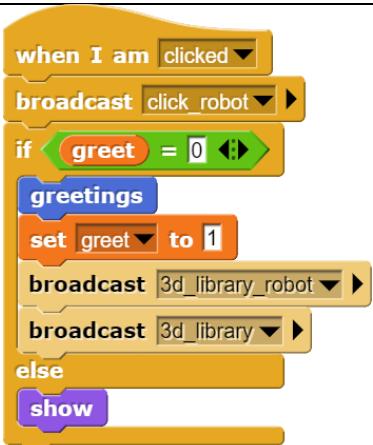
Code	Concept
------	---------



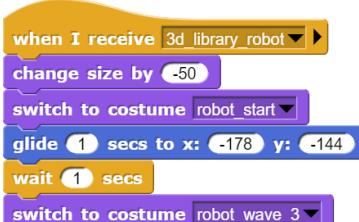
When flag button is clicked, the value of the variable “[greet](#)” created is set back to 0, so values assigned to that variable previously will be initialize back to 0. Same goes to the size of the “Robot” sprite, it initializes its size to 100%, thus previous modifications made to the value of its size is reset back to 100%. The “Robot” sprite then hide itself on the stage, preventing itself from blocking the opening animation.



When the second script receives “book_venture” broadcasted by the “Door” sprite, “Robot” sprite will switch costume to “robot_stop”, meaning that the robot is shut down, and move to the fixated position. “Robot” sprite then shrinks in size by setting its size to 10% of the original size and show itself. As “Robot” sprite appears as the shrank version of itself, it enters a loop that iterates 9 times. For each iteration, “Robot” sprite will increase its size by 10. This delivers the phenomenon of the robot appearing between the doors drawn by the “Door” sprite and increasing to its original size.



When “Robot” sprite is clicked, it will broadcast “click_robot” to “Grey_thing” sprite so that it will appear on stage and block out the “Please select the level” written by the “Level” sprite and will broadcast to the “Stage” so that it can turn down the volume of the sound “door_opening” displayed. If the “[greet](#)” variable equals to zero, implying that no greeting has been done before, “Robot” will greet the users and take input of the users’ name to greet them by their names via the [“greetings” custom Motion block](#). The value of “[greet](#)” value is assigned as 1. Once done, “3d_library_robot” will be broadcasted to the next script under the same sprite; and “3d_library” will be broadcasted to “Library_intro” to display the 3D model of the library, “Grey_thing” to hide itself and “End_tour” to provide user the option to end the tour. “3d_library” will also be broadcasted to

	the “Stage” to stop all sounds playing and switch background costume to “Grey_black_border”.
 <pre> when I receive [3d_library_robot v] change (size) by -50 switch to costume [robot_start v] glide (1) secs to x: (-178) y: (-144) wait (1) secs switch to costume [robot_wave_3 v] </pre>	When the fourth script receives “3d_library_robot”, “Robot” sprite will shrink in size by 50 and switch costume to “robot_start”. “Robot” sprite then glides to the desired location and switch costume to “robot_wave_3”, which gives the illusion that as the robot lifts up its left hand, the 3D model of the library has magically appeared from its hand.

Sprite Involved 5: “Grey_thing” Sprite

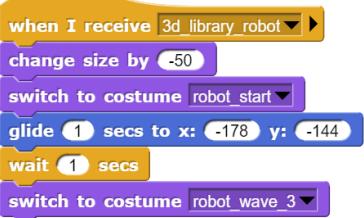
Table 3.1.1.5 Programming Concept of “Grey_thing” Sprite in Opening Animation

Code	Concept
 <pre> when green flag is clicked hide </pre>	When flag button  is clicked, “Grey_thing” sprite will hide itself on the stage. Therefore, when users click on the flag button  again, “Grey_thing” sprite will not appear on stage, obstructing the overall presentation of the opening animation.
 <pre> when I receive [click_robot v] if [greet v] = [0 v] then set [greet v] to [1] wait (0.03) secs show else hide end </pre>	When the second script receives “click_robot” broadcasted by “Robot” sprite, indicating that the robot is clicked, “Grey_thing” sprite will set the value of “greet” variable to 1 and show itself on stage, provided that the value “greet” variable equals to 0 when the “Robot” sprite is clicked. However, if the value of “greet” variable does not equal to 0 when the “Robot” sprite is clicked, it will remain hidden.
 <pre> when I receive [3d_library v] hide </pre>	When the third script receives “3d_library” broadcasted by the “Robot” sprite, “Grey_thing” sprite will hide itself on the stage, refraining from interfering the following display of the 3D model of library by the “Library_intro” sprite, in the meantime also preventing from remaining on the stage and interrupting the other upcoming presentation.

3.1.2 Starting of the Virtual Tour

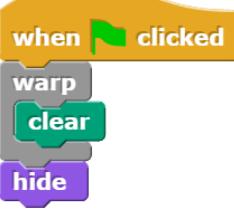
Sprite Involved 1: “Robot” Sprite

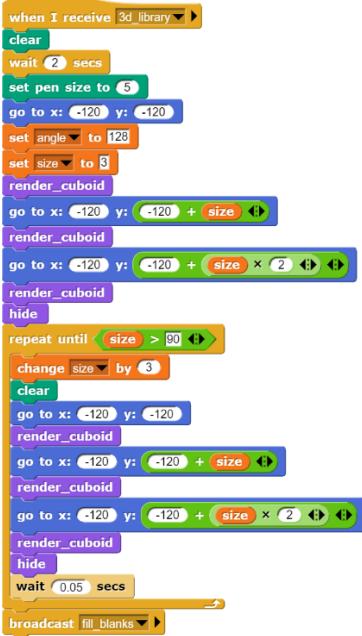
Table 3.1.2.1 Programming Concept of “Robot” Sprite in Starting of the Virtual Tour

Code	Concept
	When the fourth script receives “3d_library_robot”, “Robot” sprite will shrink in size by 50 and switch costume to “robot_start”. “Robot” sprite then glides to the desired location and switch costume to “robot_wave_3”, which gives the illusion that as the robot lifts its left hand, the 3D model of the library has magically appeared from its hand.
	When the fifth script receives “click_level”, “Robot” sprite switch costume to “robot_start”, showing that after displaying the 3D model and the levels of the library, the robot put down its hand.

Sprite Involved 2: “Library_intro” Sprite

Table 3.1.2.2 Programming Concept of “Library_intro” Sprite in Starting of the Virtual Tour

Code	Concept
	When flag button  is clicked, all drawing done on the stage previously will be cleared. Hence, when users click on the flag button  again, the previous drawing drawn will not remain on the stage, disrupting the overall presentation of the opening animation.

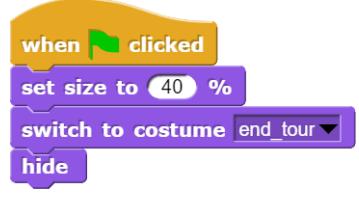
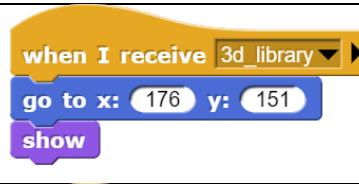
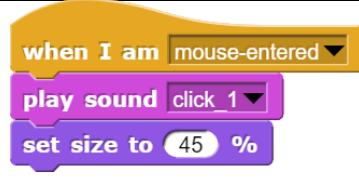
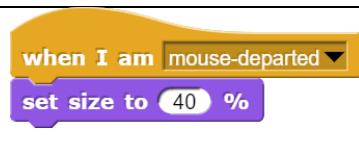


When the second script receives “3d_library” broadcasted by “Robot” sprite, “Library_intro” sprite will clear the stage and start drawing the 3D model of the library. “Library_intro” sprite first set its pen size to 5 and move to the specified location. “[angle](#)” variable which has been previously created to be used to give the door opening presentation will also be utilized in displaying the 3D model of the library. “[angle](#)” variable is set to 128 and the value of a newly created variable, which is “[size](#)” variable that is essential in identifying the length, width, and height of the 3D model of the library is initialize to 3. “Library_intro” sprite will then draw the first cuboid for the first level after the values of variable are initialized via the [“render_cuboid” custom Motion block](#). “Library_intro” sprite then move to the next location on the same x-axis but at a higher y-axis coordinate, which is where it is going to draw the second cuboid for the second level via the [“render_cuboid” custom Motion block](#). “Library_intro” repeat the same thing did to the second level for the third level but at a much higher y-axis coordinate. “Library_intro” sprite later enters a loop that for each loop, it will clear everything that has been previously drawn including the firstly drawn 3D model of the library and replacing it with a newly drawn one with the value of “[size](#)” variable increasing by 3. This loop will iterate until the value of “[size](#)” variable exceeds 90. This iteration of loop will showcase a gradually appearing and increasing in size of the 3D model of the library from the “Robot” sprite’s left hand, enhancing the overall visualization. Once done, “Library_intro” sprite will broadcast “fill_blanks” to the next script.

	<p>When the third script receives “fill_blanks” broadcasted by “Library_intro” sprite, it will fill the 3D model of the library with three vary colours, which are  colour for the first level,  colour for the second level, and  colour for the third level via the “fill_building” custom Pen block. This majorly makes the 3D model of the library more visually appealing. After that, “level_of_library” is then broadcasted to the “Level” sprite and the button representative sprites, including “Button_level_1”, “Button_level_2”, and “Button_level_3” sprites to reveal the buttons and level of the library for selection of the tour.</p>
---	---

Sprites Involved 3: “End_tour” Sprite

Table 3.1.2.3 Programming Concept of “End_tour” Sprite in Starting of the Virtual Tour

Code	Concept
	<p>When flag button  is clicked, “End_tour” sprite switch costume to “end_tour” with size of 40% of the original size and then hides itself on the stage.</p>
	<p>When “End_tour” receives “3d_library” broadcasted by the “Robot” sprite, it will go to the specified location, which is at the top right of the stage and show itself.</p>
	<p>When users’ mouse hovers over the “End_tour” sprite, it will increase in size by 5% from the initially set 40% of its original size and play sound “click_1”.</p>
	<p>When users’ mouse moves away from the “End_tour” sprite, it will set its size back to 40% of its original size.</p>

Other Sprites Involved: “Level” Sprite and Button Representative Sprites

The “Level” sprite and button representative sprites including “Button_level_1”, “Button_level_2”, and “Button_level_3” sprites will perform similar task throughout the entire application. To acknowledge what these following sprites will perform in this phase, please refer to:

“Level” Sprite: [*Table 3.1.6.3 Programming Concept of “Level” Sprite Throughout the Application*](#)

“Button_level_1” Sprite: [*Table 3.1.6.4 Programming Concept of “Button_level_1” Sprite Throughout the Application*](#)

“Button_level_2” Sprite: [*Table 3.1.6.5 Programming Concept of “Button_level_2” Sprite Throughout the Application*](#)

“Button_level_3” Sprite: [*Table 3.1.6.6 Programming Concept of “Button_level_3” Sprite Throughout the Application*](#)

3.1.3 Level 1 of the Library

Sprite Involved 1: “Button_level_1” Sprite

Table 3.1.3.1 Programming Concept of “Button_level_1” Sprite in Level 1 of the Library

Code	Concept
 <pre> when I am clicked play sound [Click2 v] until done stop [all but this script v] clear broadcast [level_1 v] hide end </pre>	<p>When “Button_level_1” sprite is clicked, “Button_level_1” sprite will play the sound “Click2” to allow users to know that the button is clicked. After that, it will stop all other script from running except for this script that is currently running and clear all the drawing that are drawn on the stage previously. Once done, “level_1” will be broadcasted to “Button_level_2” and “Button_level_3” to let the button to disappear in Level 1 of the library; and is also broadcasted to the sprites involved in level 1, which are “Robot”, “Cafe_icon”, “Counter_icon”, “Information_kiosk_icon”, “Back”, “Write_Back”, and “Write_click_icon” sprites so that they will appear on stage.</p>

	“level_1” is also broadcasted to the “Stage” so that it can switch the background costume to “Level_1”.
--	---

Sprite Involved 2: “Button_level_2” Sprite

Table 3.1.3.2 Programming Concept of “Button_level_2” Sprite in Level 1 of the Library

Code	Concept
	When the script receives “level_1” broadcasted by “Button_level_1” sprite, “Button_level_2” sprite hides itself to avoid itself from remaining on the stage when the stage switches to the level 1 of the library.

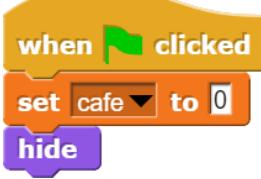
Sprite Involved 3: “Button_level_3” Sprite

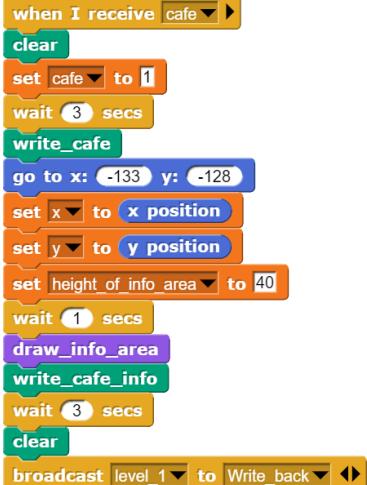
Table 3.1.3.3 Programming Concept of “Button_level_3” Sprite in Level 1 of the Library

Code	Concept
	When the script receives “level_1” broadcasted by “Button_level_1” sprite, “Button_level_3” sprite hides itself to avoid itself from remaining on the stage when the stage switches to the level 1 of the library.

Sprite Involved 4: “Cafe_icon” Sprite

Table 3.1.3.4 Programming Concept of “Cafe_icon” Sprite in Level 1 of the Library

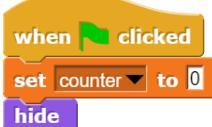
Code	Concept
	When flag button  is clicked, a variable “cafe” created is initialized with the value of 0 and the “Cafe_icon” sprite is then hidden. This variable is vital in the script written in the “Back” sprite that allow users to be redirected back to the previous scene.

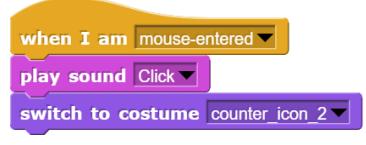
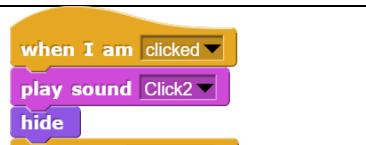
 <pre>when I am [mouse-entered v] play sound [Click v] switch to costume [cafe icon 2 v]</pre>	<p>When users’ mouse hover over the “Cafe_icon” sprite, it will play “Click” sound and switch costume to “cafe_icon_2”, meaning that the cafe icon is to be selected.</p>
 <pre>when I am [mouse-departed v] switch to costume [cafe icon 1 v]</pre>	<p>When users’ mouse moves away from the “Cafe_icon” sprite, it will switch back to costume “cafe_icon_1”, implying that the cafe icon is not selected.</p>
 <pre>when I am [clicked v] play sound [Click2 v] hide broadcast [cafe v]</pre>	<p>When “Cafe_icon” sprite is clicked, it will play sound “Click2” and hide itself. This sound displayed brings out the calming ambience one will find in most cafe. “Cafe” is then broadcasted to the next script to switch the stage to the next scene, which is the cafe.</p>
 <pre>when I receive [cafe v] clear set [cafe v] to [1] wait (3) secs write_cafe go to x: [-133] y: [-128] set [x v] to [x position] set [y v] to [y position] set [height of info area v] to [40] wait (1) secs draw_info_area write_cafe_info wait (3) secs clear broadcast [level 1 v] to [Write_back v]</pre>	<p>When the sixth script receives “cafe” broadcasted by the previous script, “Cafe_icon” sprite will clear the stage and the variable “cafe” is set to 1. “Cafe_icon” sprite that is hidden in the previous script will write the word “Cafe” on the stage at a particular location through the “write_cafe” custom Pen block. It then goes to the next specified location; and the x coordinate of that location is assigned to the variable “x” while the y coordinate of that location is assigned to the variable “y”. Here, two variables which are “x” and “y” are created for specifying the appropriate location to position the information box drawn by the “draw_info_area” custom Looks block. On the other hand, the “height_info_area” variable is created to specify the height of the information box drawn by the same “draw_info_area” custom Looks block. Once the information box is drawn at the desired location, the information that are intended to be delivered will be written in that information box drawn via the “write_cafe_info”</p>

	<u>custom Pen block</u> . After 1 second, it will clear all the drawings and information written on the stage. “level_1” is specifically broadcasted to “Write_back” sprite so that it can rewrite the word “Back” below the “Back” sprite on the stage.
	When “Cafe_icon” sprite receives “counter” broadcasted by the “Counter_icon” sprite, it will hide itself so that it will not appear on the scene where the counter is displayed and disrupting the presentation of the counter.
	When “Cafe_icon” sprite receives “kiosk” broadcasted by the “Information_kiosk_icon” sprite, it will hide itself so that it will not appear on the scene where the information kiosk is displayed and disrupting the presentation of the information kiosk.
	When “Cafe_icon” sprite receives “3d_library_back” broadcasted by “Back_sprite”, indicating that users are going to be redirected back to view the 3D model of the library to select the next level to tour, it will hide itself on the stage to avoid blocking other elements presented as the 3D model of the library is displayed by the “Library” sprite.

Sprite Involved 5: “Counter_icon” Sprite

Table 3.1.3.5 Programming Concept of “Counter_icon” Sprite in Level 1 of the Library

Code	Concept
	When flag button  is clicked, a variable “ <u>counter</u> ” created is initialized with the value of 0 and the “Counter_icon” sprite is then hidden. This variable is vital in the script written in the “Back” sprite that allow users to be redirected back to the previous scene.

 <pre>when I am [mouse-entered v] play sound [Click v] switch to costume [counter_icon 2 v]</pre>	<p>When users’ mouse hover over the “Counter_icon” sprite, it will play “Click” sound and switch costume to “counter_icon_2”, meaning that the counter icon is to be selected.</p>
 <pre>when I am [mouse-departed v] switch to costume [counter_icon 1 v]</pre>	<p>When users’ mouse moves away from the “Counter_icon” sprite, it will switch back to costume “counter_icon_1”, implying that the counter icon is not selected.</p>
 <pre>when I am [clicked v] play sound [Click2 v] hide broadcast [counter v]</pre>	<p>When “Counter_icon” sprite is clicked, it will play sound “Click2” and hide itself. “Counter” is then broadcasted to the next script to switch the stage to the next scene, which is the counter.</p>
 <pre>when I receive [counter v] clear set [counter v] to (1) wait (3) secs write [counter v] go to x: (-134) y: (-76) set [x v] to [x position v] set [y v] to [y position v] set [height_of_info_area v] to (88) wait (1) secs draw [info area v] write [counter info v] wait (3) secs clear broadcast [level 1 v] to [Write back v]</pre>	<p>When the sixth script receives “counter” broadcasted by the previous script, “Counter_icon” sprite will clear the stage and the variable “<u>counter</u>” is set to 1. “Counter_icon” sprite that is hidden in the previous script will write the word “Counter” on the stage at a particular location through the “<u>write_counter</u>” custom Pen block. It then goes to the next specified location; and the x coordinate of that location is assigned to the variable “<u>x</u>” while the y coordinate of that location is assigned to the variable “<u>y</u>”. Here, two variables which are “<u>x</u>” and “<u>y</u>” are created for specifying the appropriate location to position the information box drawn by the “<u>draw_info_area</u>” custom Looks block. On the other hand, the “height_info_area” variable is created to specify the height of the information box drawn by the same “<u>draw_info_area</u>” custom Looks block. Once the</p>

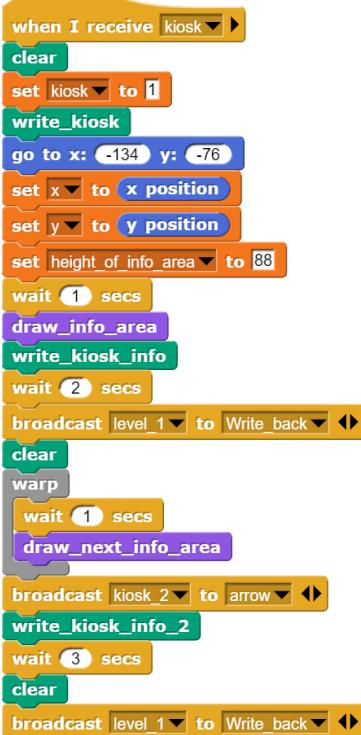
	information box is drawn at the desired location, the information that are intended to be delivered will be written in that information box drawn via the “write_counter_info” custom Pen block . After 3 second, it will clear the drawings and information written on the stage. “level_1” is specifically broadcasted to “Write_back” sprite so that it can rewrite the word “Back” below the “Back” sprite on the stage.
	When “Counter_icon” sprite receives “cafe” broadcasted by the “Cafe_icon” sprite, it will hide itself so that it will not appear on the scene where the cafe is displayed and disrupting the presentation of the cafe.
	When “Counter_icon” sprite receives “kiosk” broadcasted by the “Information_kiosk_icon” sprite, it will hide itself so that it will not appear on the scene where the information kiosk is displayed and disrupting the presentation of the information kiosk.
	When “Counter_icon” sprite receives “3d_library_back” broadcasted by “Back_sprite”, indicating that users are going to be redirected back to view the 3D model of the library to select the next level to tour, it will hide itself on the stage to avoid blocking other elements presented as the 3D model of the library is displayed by the “Library” sprite.

Sprite Involved 6: “Information_kiosk_icon” Sprite

Table 3.1.3.6 Programming Concept of “Information_kiosk_icon” Sprite in Level 1 of the Library

Code	Concept

	<p>When flag button is clicked, a variable “kiosk” created is initialized with the value of 0 and the “Information_kiosk_icon” sprite is then hidden. This variable is vital in the script written in the “Back” sprite that allow users to be redirected back to the previous scene.</p>
	<p>When the second script receives “level_1” broadcasted by the “Button_level_1” sprite, “Information_kiosk_icon” sprite will move to the specified location, which is beside the information kiosk in level 1. It then switches costume to “touch_icon_1” and show itself on stage.</p>
	<p>When users’ mouse hover over the “Information_kiosk_icon” sprite, it will play “Click” sound and switch costume to “touch_icon_2”, meaning that the information kiosk icon is to be selected.</p>
	<p>When users’ mouse moves away from the “Information_kiosk_icon” sprite, it will switch back to costume “touch_icon_1”, implying that the information kiosk icon is not selected.</p>
	<p>When “Information_kiosk_icon” sprite is clicked, it will play sound “Click2” and hide itself. “Kiosk” is then broadcasted to the next script to switch the stage to the next scene, which is the information kiosk.</p>



When the sixth script receives “kiosk” broadcasted by the previous script, “Information_kiosk_icon” sprite will clear the stage and the variable “[kiosk](#)” is set to 1. “Information_kiosk_icon” sprite that is hidden in the previous script will write the word “Information Kiosk” on the stage at a particular location through the [“write_kiosk” custom Pen block](#). It then goes to the next specified location; and the x coordinate of that location is assigned to the variable “[x](#)” while the y coordinate of that location is assigned to the variable “[y](#)”. Here, two variables which are “[x](#)” and “[y](#)” are created for specifying the appropriate location to position the information box drawn by the [“draw_info_area” custom Looks block](#). On the other hand, the “height_info_area” variable is created to specify the height of the information box drawn by the same [“draw_info_area” custom Looks block](#). Once the information box is drawn at the desired location, the information that are intended to be delivered will be written in that information box drawn via the [“write_kiosk_info” custom Pen block](#). Later, “level_1” is specifically broadcasted to “Write_back” so that it can rewrite the word “Back” below the “Back” sprite on the stage.

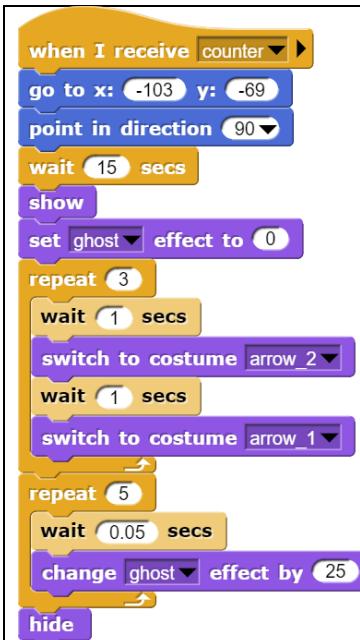
After broadcasting “level_1” to “Write_back” sprite, it will clear the previously drawn drawings and written information on the stage and immediately redraw the information box shortly via the [“draw_next_info_area” custom Looks block](#). “Information_kiosk_icon” sprite then broadcast “kiosk_2” to “Arrow” sprite so that it appears on stage and point towards the information kiosk; and write the following information that is needed to be delivered to the users via the [“write_kiosk_info_2” custom Pen block](#). After 3 seconds, it will clear the stage once again. “level_1” is broadcasted specifically to “Write_back”

	sprite so that it can rewrite the word “Back” below the “Back” sprite on the stage.
	When “Information_kiosk_icon” sprite receives “cafe” broadcasted by the “Cafe_icon” sprite, it will hide itself so that it will not appear on the scene where the cafe is displayed and disrupting the presentation of the cafe.
	When “Information_kiosk_icon” sprite receives “counter” broadcasted by the “Counter_icon” sprite, it will hide itself so that it will not appear on the scene where the counter is displayed and disrupting the presentation of the counter.
	When “Information_kiosk_icon” sprite receives “3d_library_back” broadcasted by “Back_sprite”, indicating that users are going to be redirected back to view the 3D model of the library to select the next level to tour, it will hide itself on the stage to avoid blocking other elements presented as the 3D model of the library is displayed by the “Library” sprite.

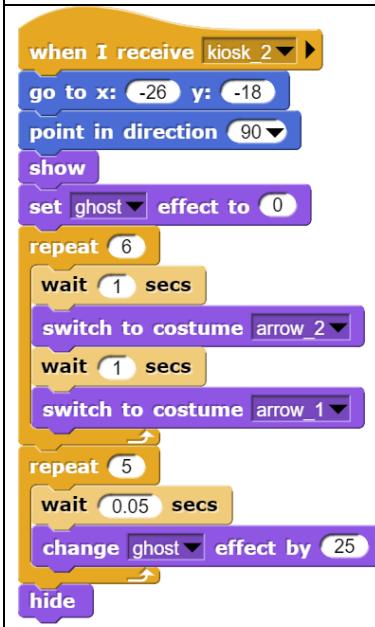
Sprite Involved 7: “Arrow” Sprite

Table 3.1.3.7 Programming Concept of “Arrow” Sprite in Level 1 of the Library

Code	Concept
	When flag button  is clicked, “Arrow” sprite change costume to “arrow_1” and hide itself on stage to prevent interfering the opening animation and the following presentation of the application.



When “Arrow” sprite receives “counter” broadcasted by “Counter_icon”, it will go to the specific position and point in a direction of 90°, which is pointing towards the counter. After waiting for 15 seconds, the ghost effect of the “Arrow” sprite is set 0 as it shows itself. It then enters a loop that is executed for 3 repetitions, which is set in respect to the time span of the scene, where for each loop it interchangeably switches costume between “arrow_2” and “arrow_1”, enhancing the overall presentation. It then enters another loop that increases the ghost effect by 25 for each loop that goes on for 5 iterations until the ghost effect reach 100. “Arrow” sprite then hide itself once the loop ends.

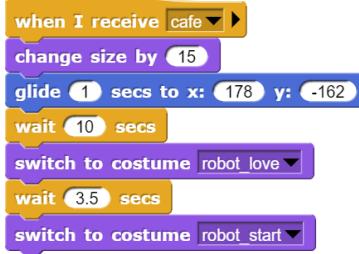
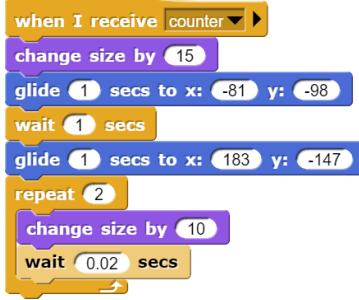


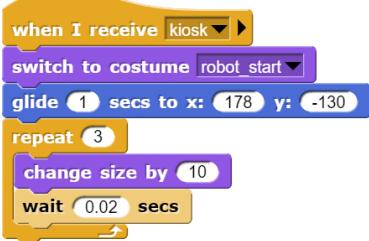
When “Arrow” sprite receives “kiosk_2” broadcasted by “Information_kiosk_icon”, it will go to the specific position and point in a direction of 90°, which is pointing towards the information kiosk. After that, the ghost effect of the “Arrow” sprite is set 0 as it shows itself. It then enters a loop that is executed for 6 repetitions, which is set in respect to the time span of the scene, where for each loop it interchangeably switches costume between “arrow_2” and “arrow_1”, enhancing the overall presentation. It then enters another loop that increases the ghost effect by 25 for each loop that goes on for 5 iterations until the ghost effect reach 100. “Arrow” sprite then hide itself once the loop ends.

Sprite Involved 8: “Robot” Sprite

Table 3.1.3.8 Programming Concept of “Robot” Sprite in Level 1 of the Library

Code	Concept
------	---------

 <pre> when I receive [cafe v] change size by [15] glide [1 sec] to x: [178] y: [-162] wait [10 sec] switch to costume [robot_love v] wait [3.5 sec] switch to costume [robot_start v] </pre>	<p>When the seventh script receives “cafe” broadcasted by the “Cafe_icon” sprite, which means that users interact with the “Cafe_icon” sprite by clicking on it, the “Robot” sprite will change its size by enlarging itself by 15 and glide 1 second to a certain position. After 10 seconds, when the “Cafe_icon” sprite is writing some words, which are “You can have a cup of coffee here.”, the “Robot” sprite will switch costume to “robot_love”, which shows that the “Robot” sprite loves this kind of environment. After 3.5 seconds, when the “Cafe_icon” sprite finishes writing and the dialogue ends, the “Robot” sprite will switch costume back to “robot_start”.</p>
 <pre> when I receive [counter v] change size by [15] glide [1 sec] to x: [-81] y: [-98] wait [1 sec] glide [1 sec] to x: [183] y: [-147] repeat (2) change size by [10] wait [0.02 sec] end </pre>	<p>When the eighth script receives “counter” broadcasted by the “Counter_icon” sprite, which means that users interact with the “Counter_icon” sprite by clicking on it, the “Robot” sprite will change its size by enlarging itself by 15 and glide 1 second to a certain position, which is next to the counter. After 1 second, the “Robot” sprite will glide 1 second to another certain position and enlarge its size by 10 two times to show that the “Robot” sprite wants to introduce the counter in the library by writing some words from “Counter_icon” sprite, which are “Need assistance? Our librarians are here to help you at the counter.”.</p>



When the ninth script receives “kiosk” broadcasted by the “Information_kiosk_icon” sprite, which means that users interact with the “Information_kiosk_icon” sprite by clicking on it, the “Robot” sprite switches costume to “robot_start” and glide 1 second to a certain position. After that, the “Robot” sprite will change its size by enlarging itself by 10 three times to show that the “Robot” sprite wants to introduce the information kiosk in the library by writing some words from “Information_kiosk_icon” sprite, which are “Need help? Information kiosk is here to guide you” and “All confusions about the library can be answered by information kiosk.”.

Other Sprites Involved: “Library” Sprite, “Level” Sprite, Button Representative Sprites, “End_tour” Sprite, “Back” Sprite, “Write_back” Sprite, “Write_click_icon” Sprite

These sprites will perform similar task throughout the entire application. To acknowledge what these following sprites will perform in this phase, please refer to:

“Library” Sprite: [Table 3.1.6.5 Programming Concept of “Button_level_1” Sprite Throughout the Application](#)

“Level” Sprite: [Table 3.1.6.3 Programming Concept of “Level” Sprite Throughout the Application](#)

“Button_level_1” Sprite: [Table 3.1.6.4 Programming Concept of “Button_level_1” Sprite Throughout the Application](#)

“Button_level_2” Sprite: [Table 3.1.6.5 Programming Concept of “Button_level_2” Sprite Throughout the Application](#)

“Button_level_3” Sprite: [Table 3.1.6.6 Programming Concept of “Button_level_3” Sprite Throughout the Application](#)

“End_tour” Sprite: [Table 3.1.6.9 Programming Concept of “End_tour” Sprite Throughout the Application](#)

“Back” Sprite: [Table 3.1.6.1 Programming Concept of “Back” Sprite Throughout the Application](#)

“Write_back” Sprite: [Table 3.1.6.2 Programming Concept of “Write_back” Sprite Throughout the Application](#)

“Write_click_icon” Sprite: [Table 3.1.6.8 Programming Concept of “Write_click_icon” Sprite Throughout the Application](#)

3.1.4 Level 2 of the Library

Sprite Involved 1: “Button_level_2” Sprite

Table 3.1.4.1 Programming Concept of “Button_level_2” Sprite in Level 2 of the Library

Code	Concept
 <pre> when I am clicked play sound [Click2 v] stop [all but this script v] clear broadcast [level_2 v] hide </pre>	<p>When “Button_level_2” sprite is clicked, “Button_level_2” sprite will play the sound “Click2” to allow users to know that the button is clicked. After that, it will stop all other script from running except for this script that is currently running and clear all the drawing that are drawn on the stage previously. Once done, “level_2” will be broadcasted to “Button_level_1” and “Button_level_3” to let the button to disappear in Level 2 of the library. “level_2” is also broadcasted to the “Stage” so that it can switch the background costume to “Level_2”.</p>

Sprite Involved 2: “Button_level_1” Sprite

Table 3.1.4.2 Programming Concept of “Button_level_1” Sprite in Level 2 of the Library

Code	Concept
 <pre> when I receive [level_2 v] hide </pre>	<p>When the script receives “level_2” broadcasted by “Button_level_2” sprite, “Button_level_1” sprite hides itself to avoid itself from remaining on the stage when the stage switches to the level 2 of the library.</p>

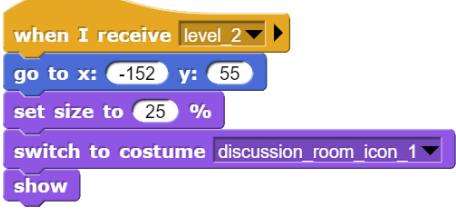
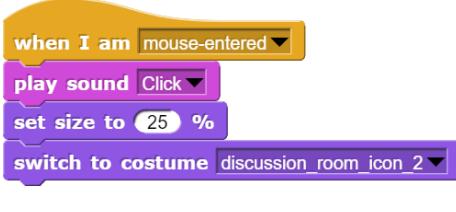
Sprite Involved 3: “Button_level_3” Sprite

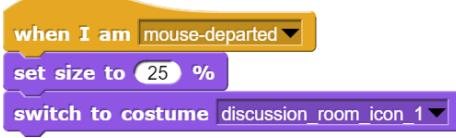
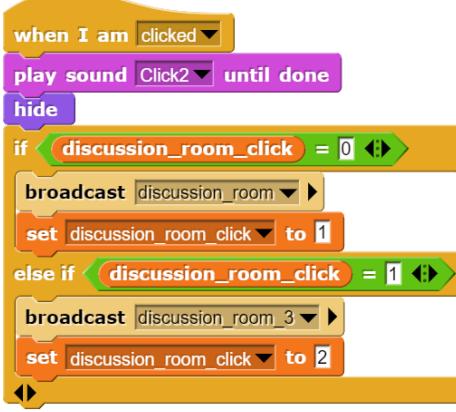
Table 3.1.4.3 Programming Concept of “Button_level_3” Sprite in Level 2 of the Library

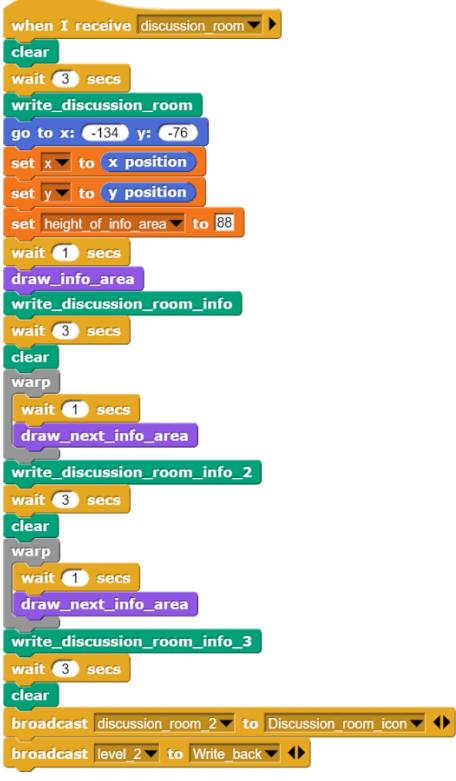
Code	Concept
	When the script receives “level_2” broadcasted by “Button_level_2” sprite, “Button_level_3” sprite hides itself to avoid itself from remaining on the stage when the stage switches to the level 2 of the library.

Sprite Involved 4: “Discussion_room_icon” Sprite

Table 3.1.4.4 Programming Concept of “Discussion_room_icon” Sprite in Level 2 of the Library

Code	Concept
	When flag button  is clicked, a variable “discussion_room_click” created is initialized with the value of 0 and the “Discussion_room_icon” sprite is then hidden. This variable is vital in the script written in the “Back” sprite that allow users to be redirected back to the previous scene.
	When the second script receives “level_2” broadcasted by the “Button_level_2” sprite, “Discussion_room_icon” sprite will move to the specified location, which is above the discussion room in level 2 and shrinks in size by setting its size to 25% of the original size. It then switches costume to “discussion_room_icon_1” and show itself on stage.
	When users’ mouse hover over the “Discussion_room_icon” sprite, it will play “Click” sound. It shrinks in size by setting its size to 25% of the original size and switch costume to

	“discussion_room_icon_2”, meaning that the discussion room icon is to be selected.
 <pre> when I am mouse-departed set size to 25 % switch to costume [discussion_room_icon_1] </pre>	When users' mouse moves away from the “Discussion_room_icon” sprite, it will shrink in size by setting its size to 25% of the original size and switch back to costume “discussion_room_icon_1”, implying that the discussion room icon is not selected.
 <pre> when I am clicked play sound [Click2 v] until done hide if [discussion_room_click] = [0] then broadcast [discussion_room v] set [discussion_room_click] to [1] else if [discussion_room_click] = [1] then broadcast [discussion_room_3 v] set [discussion_room_click] to [2] end end </pre>	When “Discussion_room_icon” sprite is clicked, it will play sound “Click2” and hide itself. The variable <u>“discussion_room_click”</u> plays a major role in ensuring itself can execute the right function which is redirecting the users to the next scene correctly; and ensuring “Back” sprite can produce the desired outcome, which is redirecting the users back to the intended previous scene. To be specific, there are two scenes involved in introducing the discussion room, thus the “Discussion_room_icon” will be clicked at most twice in a row according to order. The first click is to direct users to view the magnified image of the hallway facing the discussion rooms, while the second click is to direct users to the discussion room to show the interior. So if the value of <u>“discussion_room_click”</u> variable remains 0 , which means that this is the first click made and “Discussion_room_icon” sprite has not previously been clicked before, “discussion_room” is broadcasted to the “Stage” so that it can change background costume to “Discussion_room” and is broadcasted to “Book_shelf_icon” sprite so that it can hide itself on stage. It is also broadcasted to its next script to share information about the discussion room. The value of <u>“discussion_room_click”</u> is set to 1, recording the first

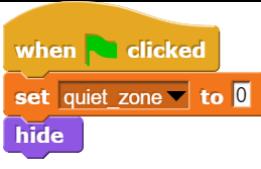
	<p>click. If the value of “discussion_room_click” variable is equated to 1, implying that the “Discussion_room_icon” sprite has previously been clicked once, it will broadcast “discussion_room_3” to “Stage” to switch background costume to “Discussion_room_2”. The value of “discussion_room_click” is set to 2, tracking the second click.</p>
 <pre> when I receive discussion_room clear wait (3) secs write_discussion_room go to x: (-134) y: (-76) set [x v] to [x position] set [y v] to [y position] set [height of info area v] to (88) wait (1) secs draw_info_area write_discussion_room_info wait (3) secs clear warp wait (1) secs draw_next_info_area write_discussion_room_info_2 wait (3) secs clear warp wait (1) secs draw_next_info_area write_discussion_room_info_3 wait (3) secs clear broadcast [discussion_room_2 v] to [Discussion_room_icon v] broadcast [level 2 v] to [Write back v] end </pre>	<p>When the sixth script receives “discussion_room” broadcasted by the previous script, “Discussion_room_icon” sprite will clear the stage. “Discussion_room_icon” sprite that is hidden in the previous script will write the word “Discussion room” on the stage at a particular location through the “write_discussion_room” custom Pen block. It then goes to the next specified location; and the x coordinate of that location is assigned to the variable “x” while the y coordinate of that location is assigned to the variable “y”. Here, two variables which are “x” and “y” are created for specifying the appropriate location to position the information box drawn by the “draw_info_area” custom Looks block. On the other hand, the “height_info_area” variable is created to specify the height of the information box drawn by the same “draw_info_area” custom Looks block. Once the information box is drawn at the desired location, the information that are intended to be delivered will be written in that information box drawn via the “write_discussion_room_info” custom Pen block. After 3 seconds, it will clear the previously drawn drawings and written information on the stage and immediately redraw the information box shortly via</p>

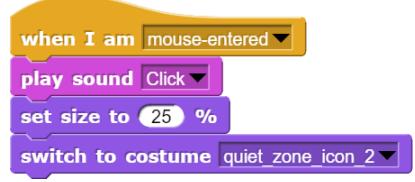
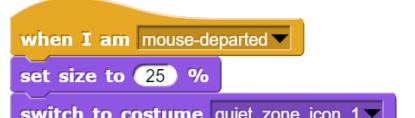
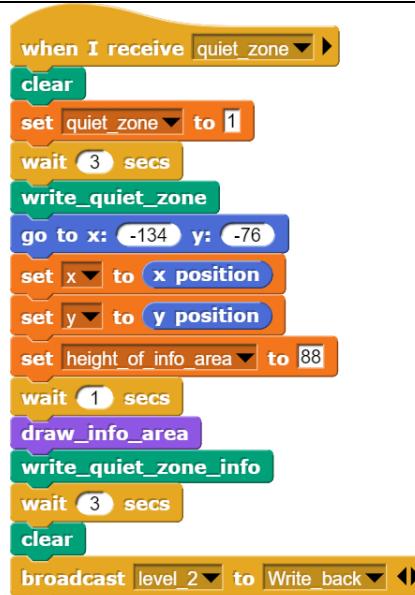
	<p><u>“draw next info area” custom Looks block</u></p> <p>“Discussion_room_icon” sprite then write the following information that is needed to be delivered to the users via the <u>“write discussion room info 2” custom Pen block</u>. After 3 seconds, it will clear the drawings and information written on the stage and immediately redraw the information box shortly <u>via “draw next info area” custom Looks block</u>. Discussion_room_icon” sprite then write the subsequent information to share to the users via the <u>“write discussion room info 3” custom Pen block</u>. “discussion_room_2” is specifically broadcasted to its next script while “level_2” is specifically broadcasted to “Write_back” sprite so that it can rewrite the word “Back” below the “Back” sprite on the stage.</p>
 <pre> when I receive [discussion room 2 v] go to x: 120 y: 143 show </pre>	<p>When “Discussion_room_icon” sprite receives “discussion_room_2” broadcasted from the previous script, “Discussion_room_icon” sprite that is hidden will move to a specific location and show itself to execute the following function that allow users to see the interior of the discussion room when clicked. This means that the “Discussion_room_icon” sprite might be clicked more than once in a row.</p>
 <pre> when I receive [discussion room 3 v] hide wait [1] secs write [discussion room info 4] wait [3] secs clear broadcast [level 2 v] to [Write back] </pre>	<p>When “Discussion_room_icon” sprite receives “discussion_room_3” broadcasted by the fifth script when it is clicked the second time, it will hide itself and move to a specific location, which is on the television’s screen. It will then write the information intended to be display via the <u>“write discussion room info 4” custom Pen block</u>. After 3 seconds, it will clear the stage and broadcast “level_2” specifically to the “Write_back”</p>

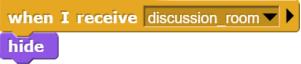
	sprite so that it can rewrite the word “Back” below the “Back” sprite on the stage.
	When “Discussion_room_icon” sprite receives “quite_zone” broadcasted by the “Quiet_zone_icon” sprite, it will hide itself so that it will not appear on the scene where the quiet zone is displayed and disrupting the presentation of the quiet zone.
	When “Discussion_room_icon” sprite receives “3d_library_back” broadcasted by “Back_sprite”, indicating that users are going to be redirected back to view the 3D model of the library to select the next level to tour, it will hide itself on the stage to avoid blocking other elements presented as the 3D model of the library is displayed by the “Library” sprite.

Sprite Involved 5: “Quiet_zone_icon” Sprite

Table 3.1.4.5 Programming Concept of “Quiet_zone_icon” Sprite in Level 2 of the Library

Code	Concept
	When flag button  is clicked, a variable “quiet_zone” created is initialized with the value of 0 and the “Quiet_zone_icon” sprite is then hidden. This variable is vital in the script written in the “Back” sprite that allow users to be redirected back to the previous scene.
	When the second script receives “level_2” broadcasted by the “Button_level_2” sprite, “Quiet_zone_icon” sprite will move to the specified location, which is above the quiet zone in level 2 and shrinks in size by setting its size to 25% of the original size. It then switches costume to “quiet_zone_icon_1” and show itself on stage.

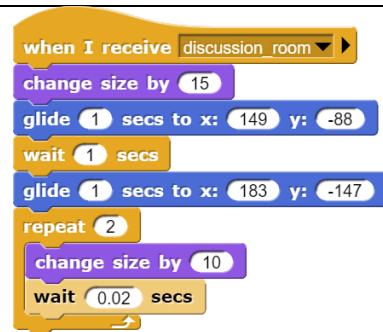
 <pre> when I am mouse-entered play sound Click v set size to 25 % switch to costume quiet_zone_icon_2 </pre>	<p>When users' mouse hover over the "Quiet_zone_icon" sprite, it will play "Click" sound. It shrinks in size by setting its size to 25% of the original size and switch costume to "quiet_zone_icon_2", meaning that the quiet zone icon is to be selected.</p>
 <pre> when I am mouse-departed set size to 25 % switch to costume quiet_zone_icon_1 </pre>	<p>When users' mouse moves away from the "Quiet_zone_icon" sprite, it will shrink in size by setting its size to 25% of the original size and switch back to costume "quiet_zone_icon_1", implying that the quiet zone icon is not selected.</p>
 <pre> when I am clicked play sound Click2 v until done hide broadcast quiet_zone </pre>	<p>When "Quiet_zone_icon" sprite is clicked, it will play sound "Click2" and hide itself. "Quiet_zone" is then broadcasted to the next script to switch the stage to the next scene, which is the quiet zone.</p>
 <pre> when I receive quiet_zone clear set quiet_zone to 1 wait (3) secs write_quiet_zone go to x: (-134) y: (-76) set [x v] to [x position] set [y v] to [y position] set [height_of_info_area v] to (88) wait (1) secs draw_info_area write_quiet_zone_info wait (3) secs clear broadcast level_2 to Write_back </pre>	<p>When the sixth script receives "quiet_zone" broadcasted by the previous script, "Quiet_zone_icon" will clear the stage and the variable "quiet_zone" is set to 1. "Quiet_zone_icon" sprite that is hidden in the previous script will write the word "Quiet zone" on the stage at a particular location through the "write_quiet_zone" custom Pen block. It then goes to the next specified location; and the x coordinate of that location is assigned to the variable "<u>x</u>" while the y coordinate of that location is assigned to the variable "<u>y</u>". Here, two variables which are "<u>x</u>" and "<u>y</u>" are created for specifying the appropriate location to position the information box drawn by the "draw_info_area" custom Looks block. On the other hand, the "height_info_area" variable is created to specify the height of the information box drawn by the same "draw_info_area" custom Looks block. Once the information box is drawn at the desired location, the</p>

	information that are intended to be delivered will be written in that information box drawn via the “ write_quiet_zone_info ” custom Pen block. After 3 seconds, it will clear the stage and “level_2” is broadcasted to “Write_back” sprite so that it can rewrite the word “Back” below the “Back” sprite on the stage.
	When “Quiet_zone_icon” sprite receives “discussion_room” broadcasted by the “Discussion_room_icon” sprite, it will hide itself so that it will not appear on the scene where the discussion room is displayed and disrupting the presentation of the discussion room.
	When “Quiet_zone_icon” sprite receives “3d_library_back” broadcasted by “Back_sprite”, indicating that users are going to be redirected back to view the 3D model of the library to select the next level to tour, it will hide itself on the stage to avoid blocking other elements presented as the 3D model of the library is displayed by the “Library” sprite.

Sprite Involved 6: “Robot” Sprite

Table 3.1.4.6 Programming Concept of “Robot” Sprite in Level 2 of the Library

Code	Concept
	When the tenth script receives “level_2” broadcasted by the “Button_level_2” sprite, the “Robot” sprite switches costume to “robot_start” and shows itself on the stage. It then shrinks in size by setting its size to 50% of the original size and glides 1 second to a certain position to show that the “Robot” sprite is moving and standing to the center of level 2 of the library.



When the eleventh script receives “discussion_room” broadcasted by the “Discussion_room_icon” sprite, which means that users interact with the “Discussion_room_icon” sprite by clicking on it, the “Robot” sprite will change its size by enlarging itself by 15 and glide 1 second to a certain position. After 1 second, the “Robot” sprite will glide 1 second to another certain position and enlarge its size by 10 two times to show that the “Robot” sprite wants to introduce the discussion room in the library by writing some words from “Discussion_room_icon” sprite, which are “You can book a discussion room at the counter for academic purposes.” and “For more details, please visit the counter.”.



When the twelfth script receives “quiet_zone” broadcasted by the “Quiet_zone_icon” sprite, which means that users interact with the “Quiet_zone_icon” sprite by clicking on it, the “Robot” sprite will change its size by enlarging itself by 15 and glide 1 second to a certain position. After 1 second, the “Robot” sprite will glide 1 second to another certain position and enlarge its size by 10 two times to show that the “Robot” sprite wants to introduce the quiet zone in the library by writing some words from “Quiet_zone_icon” sprite, which are “Shhh ... Please keep your voice down. This is a place you can study alone with no interruptions.”.

Other Sprites Involved: Library” Sprite, “Level” Sprite, Button Representative Sprites, “End_tour” Sprite, “Back” Sprite, “Write_back” Sprite, “Write_click_icon” Sprite

These sprites will perform similar task throughout the entire application. To acknowledge what these following sprites will perform in this phase, please refer to:

“Library” Sprite: [Table 3.1.6.5 Programming Concept of “Button_level_1” Sprite Throughout the Application](#)

“Level” Sprite: [Table 3.1.6.3 Programming Concept of “Level” Sprite Throughout the Application](#)

“Button_level_1” Sprite: [Table 3.1.6.4 Programming Concept of “Button_level_1” Sprite Throughout the Application](#)

“Button_level_2” Sprite: [Table 3.1.6.5 Programming Concept of “Button_level_2” Sprite Throughout the Application](#)

“Button_level_3” Sprite: [Table 3.1.6.6 Programming Concept of “Button_level_3” Sprite Throughout the Application](#)

“End_tour” Sprite: [Table 3.1.6.9 Programming Concept of “End_tour” Sprite Throughout the Application](#)

“Back” Sprite: [Table 3.1.6.1 Programming Concept of “Back” Sprite Throughout the Application](#)

“Write_back” Sprite: [Table 3.1.6.2 Programming Concept of “Write_back” Sprite Throughout the Application](#)

“Write_click_icon” Sprite: [Table 3.1.6.8 Programming Concept of “Write_click_icon” Sprite Throughout the Application](#)

3.1.5 Level 3 of the Library

Sprite Involved 1: “Button_level_3” Sprite

Table 3.1.5.1 Programming Concept of “Button_level_3” Sprite in Level 3 of the Library

Code	Concept
 A Scratch script for the "Button_level_3" sprite. It consists of the following blocks: <ul style="list-style-type: none">A "when I am clicked" hat block.A "play sound click_2 until done" sound block.A "stop all but this script" control block.A "clear" control block.A "broadcast level 3" control block.A "hide" control block.	When “Button_level_3” sprite is clicked, “Button_level_3” sprite will play the sound “Click2” to allow users to know that the button is clicked. After that, it will stop all other script from running except for this script that is currently running and clear all the drawing that are drawn on the stage previously. Once done, “level_3” will be broadcasted to “Button_level_1” and “Button_level_2” to let the button to disappear in Level 3 of the library.

	library. “level_3” is also broadcasted to the “Stage” so that it can switch the background costume to “Level_3”.
--	--

Sprite Involved 2: “Button_level_1” Sprite

Table 3.1.5.2 Programming Concept of “Button_level_1” Sprite in Level 3 of the Library

Code	Concept
	When the script receives “level_3” broadcasted by “Button_level_3” sprite, “Button_level_1” sprite hides itself to avoid itself from remaining on the stage when the stage switches to the level 3 of the library.

Sprite Involved 3: “Button_level_2” Sprite

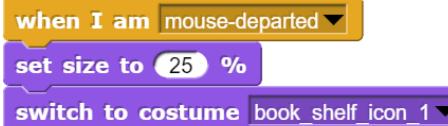
Table 3.1.5.3 Programming Concept of “Button_level_2” Sprite in Level 3 of the Library

Code	Concept
	When the script receives “level_3” broadcasted by “Button_level_3” sprite, “Button_level_2” sprite hides itself to avoid itself from remaining on the stage when the stage switches to the level 3 of the library.

Sprite Involved 4: “Book_shelf_icon” Sprite

Table 3.1.5.4 Programming Concept of “Book_shelf_icon” Sprite in Level 3 of the Library

Code	Concept
	When flag button  is clicked, a variable “book_shelf” created is initialized with the value of 0 and the “Book_shelf_icon” sprite is then hidden. This variable is vital in the script written in the

	“Back” sprite that allow users to be redirected back to the previous scene.
 <pre> when I receive level 3 go to x: -111 y: 48 set size to 25 % switch to costume book_shelf_icon_1 show </pre>	When the second script receives “level_3” broadcasted by the “Button_level_3” sprite, “Book_shelf_icon” sprite will move to the specified location, which is beside the book shelf in level 3 and shrinks in size by setting its size to 25% of the original size. It then switches costume to “book_shelf_icon_1” and show itself on stage.
 <pre> when I am mouse-entered play sound Click set size to 25 % switch to costume book_shelf_icon_2 </pre>	When users’ mouse hover over the “Book_shelf_icon” sprite, it will play “Click” sound. It shrinks in size by setting its size to 25% of the original size and switch costume to “book_shelf_icon_2”, meaning that the book shelf icon is to be selected.
 <pre> when I am mouse-departed set size to 25 % switch to costume book_shelf_icon_1 </pre>	When users’ mouse moves away from the “Book_shelf_icon” sprite, it will shrink in size by setting its size to 25% of the original size and switch back to costume “book_shelf_icon_1”, implying that the book shelf icon is not selected.
 <pre> when I am clicked play sound Click2 hide broadcast book_shelf </pre>	When “Book_shelf_icon” sprite is clicked, it will play sound “Click2” and hide itself. “Book_shelf” is then broadcasted to the next script to switch the stage to the next scene, which is the book shelf.

```

when I receive book_shelf ▶
clear
set book_shelf to 1
wait 3 secs
write_book_shelf
go to x: -134 y: -76
set x to x position
set y to y position
set height_of_info_area to 88
wait 1 secs
draw_info_area
write_book_shelf_info
wait 3 secs
clear
broadcast level_3 to Write_back ◀

```

When the sixth script receives “book_shelf” broadcasted by the previous script, the variable [“book_shelf”](#) is set to 1. “Book_shelf_icon” sprite that is hidden in the previous script will write the word “Book shelf” on the stage at a particular location through the [“write_book_shelf” custom Pen block](#). It then goes to the next specified location; and the x coordinate of that location is assigned to the variable [“x”](#) while the y coordinate of that location is assigned to the variable [“y”](#). Here, two variables which are [“x”](#) and [“y”](#) are created for specifying the appropriate location to position the information box drawn by the [“draw_info_area” custom Looks block](#). On the other hand, the “height_info_area” variable is created to specify the height of the information box drawn by the same [“draw_info_area” custom Looks block](#). Once the information box is drawn at the desired location, the information that are intended to be delivered will be written in that information box drawn via the [“write_book_shelf_info” custom Pen block](#). After 3 seconds, “Book_shelf_icon” will clear the drawings and information written on the stage. “level_3” is broadcasted to “Write_back” sprite so that it can rewrite the word “Back” below the “Back” sprite on the stage.

```

when I receive computer_scene ▶
hide

```

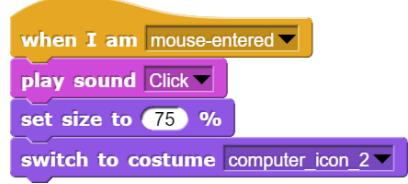
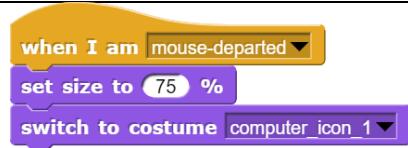
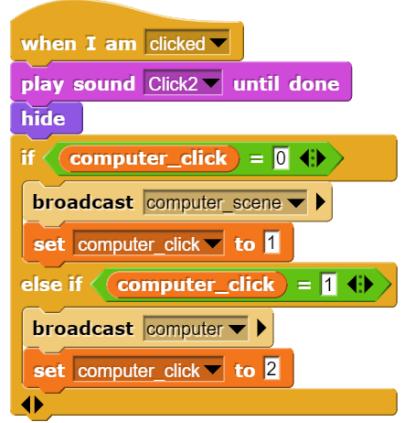
When “Book_shelf_icon” sprite receives “computer_scene” broadcasted by the “Computer_icon” sprite, it will hide itself so that it will not appear on the scene where the computer

	zone is displayed and disrupting the presentation of the computer zone.
<pre>when I receive [3d_library_back v] hide</pre>	When “Book_shelf_icon” sprite receives “3d_library_back” broadcasted by “Back_sprite”, indicating that users are going to be redirected back to view the 3D model of the library to select the next level to tour, it will hide itself on the stage to avoid blocking other elements presented as the 3D model of the library is displayed by the “Library” sprite.

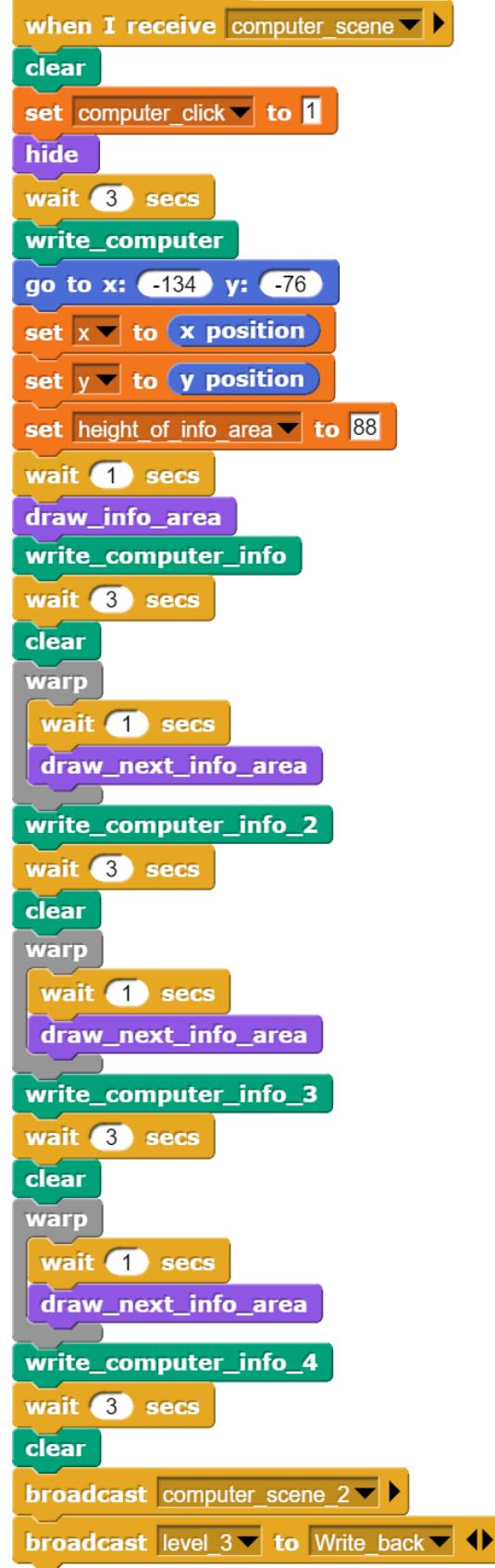
Sprite Involved 5: “Computer_icon” Sprite

Table 3.1.5.5 Programming Concept of “Computer_icon” Sprite in Level 3 of the Library

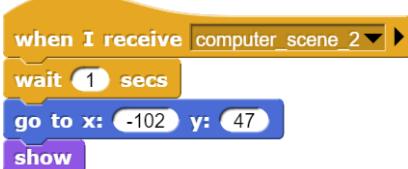
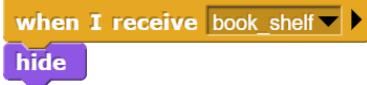
Code	Concept
<pre>when green flag clicked set [computer_click v] to [0] hide</pre>	When flag button is clicked, variable <u>“computer_click”</u> created is initialized with the value of 0 and the “Computer_icon” sprite is then hidden. This variable is vital in the script written in the “Back” sprite that allow users to be redirected back to the previous scene.
<pre>when I receive [level_3 v] set [computer_click v] to [0] go to x: (61) y: (-4) set size to (75) % switch to costume [computer_icon_1 v] show</pre>	When the second script receives “level_3” broadcasted by the “Button_level_3” sprite, the variable <u>“computer_click”</u> is set to 0. “Book_shelf_icon” sprite will move to the specified location, which is above the computer zone in level 3 and shrinks in size by setting its size to 75% of the original size. It then switches costume to “computer_icon_1” and show itself on stage.

	<p>When users' mouse hover over the "Computer_icon" sprite, it will play "Click" sound. It shrinks in size by setting its size to 75% of the original size and switch costume to "computer_icon_2", meaning that the computer icon is to be selected.</p>
	<p>When users' mouse moves away from the "Computer_icon" sprite, it will shrink in size by setting its size to 75% of the original size and switch back to costume "computer_icon_1", implying that the computer icon is not selected.</p>
	<p>When "Computer_icon" sprite is clicked, it will play sound "Click2" and hide itself. The variable "<u>computer_click</u>" plays a major role in ensuring itself can execute the right function which is redirecting the users to the next scene correctly; and ensuring "Back" sprite can produce the desired outcome, which is redirecting the users back to the intended previous scene. To be specific, there are two scenes involved in introducing the computer zone, thus the "Computer_icon" will be clicked at most twice in a row according to order. The first click is to direct users to view the magnified image of the computer zone, while the second click is to direct users to a computer screen that allows users to type anything on it. So if the value of "<u>computer_click</u>" variable remains 0, which means that this is the first click made and "Computer_icon" sprite has not previously been clicked before, "computer_scene" is broadcasted to the "Stage" so that it can change background</p>

costume to “Computer_scene” and is also broadcasted to “Book_shelf_icon” sprite so that it can hide itself on stage. The value of [“computer_click”](#) is set to 1, recording the first click. If the value of [“computer_click”](#) variable is equated to 1, implying that the “Computer_icon” sprite has previously been clicked once, it will broadcast “computer” to “Stage” to switch background costume to “Computer” and also to “Notepad” sprite so that it will show itself on stage. The value of [“computer_click”](#) is set to 2, tracking the second click.

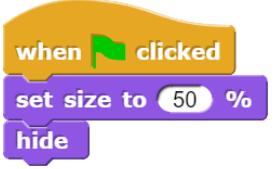


When the sixth script receives “computer_scene” broadcasted by the previous script, the variable “[computer click](#)” is set to 1. “Computer_icon” sprite that is hidden in the previous script will write the word “Computer zone” on the stage at a particular location through the “[write_computer](#)” custom Pen block. It then goes to the next specified location; and the x coordinate of that location is assigned to the variable “[x](#)” while the y coordinate of that location is assigned to the variable “[y](#)”. Here, two variables which are “[x](#)” and “[y](#)” are created for specifying the appropriate location to position the information box drawn by the “[draw_info_area](#)” custom Looks block. On the other hand, the “height_info_area” variable is created to specify the height of the information box drawn by the same “[draw_info_area](#)” custom Looks block. Once the information box is drawn at the desired location, the information that are intended to be delivered will be written in that information box drawn via the “[write_computer_info](#)” custom Pen block. After 3 seconds, it will clear the previously drawn drawings and written information on the stage and immediately redraw the information box shortly via “[draw_next_info_area](#)” custom Looks block. “Computer_icon” sprite then write the following information that is needed to be delivered to the users via the “[write_computer_info_2](#)” custom Pen block. After 3 seconds, the same procedure repeats again for writing the third information via the “[write_computer_info_3](#)” custom Pen block and for

	writing the fourth information via the “ write_computer_info_4 ” custom Pen block. Once all relevant information has been delivered, “computer_scene_2” is broadcasted to the next script so that users can use the computer. “level_3” is also broadcasted but specifically to “Write_back” sprite so that it can rewrite the word “Back” below the “Back” sprite on the stage.
	When the seventh script receives “computer_scene_2” broadcasted by the previous script, “Computer_icon” sprite that is hidden will move to a specific location and show itself to execute the following function that allows users to use the computer when clicked. This means that the “Computer_icon” sprite might be clicked more than once in a row.
	When “Computer_icon” sprite receives “book_shelf” broadcasted by the “Book_shelf_icon” sprite, it will hide itself so that it will not appear on the scene where the bookshelf is displayed and disrupting the presentation of the bookshelf.
	When “Computer_icon” sprite receives “3d_library_back” broadcasted by “Back_sprite”, indicating that users are going to be redirected back to view the 3D model of the library to select the next level to tour, it will hide itself on the stage to avoid blocking other elements presented as the 3D model of the library is displayed by the “Library” sprite.

Sprite Involved 6: “Notepad” Sprite

Table 3.1.5.6 Programming Concept of “Notepad” Sprite in Level 3 of the Library

Code	Concept
	When flag button  is clicked, size of “Notepad” sprite is shrunk to 50% of its original size and hidden on the stage to prevent interfering the opening animation and the following presentation of the application.
	When “Notepad” sprite received “computer” broadcasted by the “Computer_icon” sprite, it will move to the specified location and show itself, which appears as a notepad application on a computer screen on stage.
	When “Notepad” sprite receives “level_3” broadcasted other sprites, particularly by “Back” sprite, which is when users want to return to the computer zone after finish using the computer, it will hide itself on the stage.
	When “Notepad” sprite is clicked, it will clear the stage and broadcast “computer_notepad” to the “Write_notepad” sprite so that the typing of the keyboard can be activated; to the “Close_tab” sprite so that it shows itself on the stage; and to the “Stage” so that it changes background costume to “Computer_2”.

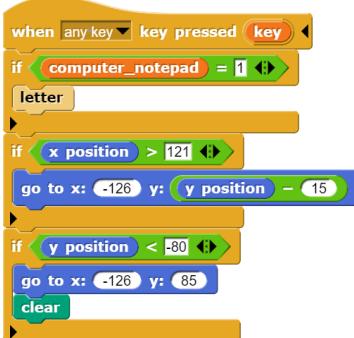
Sprite Involved 7: “Write_notepad” Sprite

Table 3.1.5.7 Programming Concept of “Write_notepad” Sprite in Level 3 of the Library

Code	Concept
	When flag button  is clicked, “Write_notepad” sprite will be hidden on the stage to prevent interfering the opening animation and the following presentation of the application.



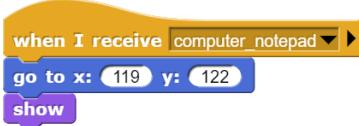
When “Write_notepad” sprite receives “computer_notepad” broadcasted by the “Notepad” sprite, it will move to a specific location which is within the computer tab displayed by “Stage” on stage. A variable “[computer_notepad](#)” is made as a method to identify whether the computer notepad is active or not, by keeping track on whether “computer_notepad” sprite is clicked. This helps prevent the typing of the keyboard to appear on another scene even if the computer notepad is not displayed, and other scene is running. So, when computer notepad is activated by the users by clicking on the “Notepad” sprite and initiated this current script run by “Write_notepad” sprite, it will set the “[computer_notepad](#)” to the value of 1, indicating that the computer notepad is in use and typing function is allowed to be run via the next script.



If the variable “[computer_notepad](#)” is equated to 1, which means that the computer note pad is in use and displayed on the stage, any key pressed on the keyboard will be written on the stage via the “[letter](#)” custom Pen block starting at the position specified in the second script. “x position” and “y position” refers to the coordinate of the current position of “Write_notepad” sprite, and it changes as this sprite write what users type on their keyboard on stage. If “x position” is greater than 121, which is outside the computer tab provided by the “Stage”, it will move to the next line by decreasing the value of “y position” by 15. If “y position” is greater than -80, which is also outside the computer tab, it will go back to the starting point that is the same coordinate specified in the second script and clear the previously written typing.

Sprite Involved 8: “Close_tab” Sprite

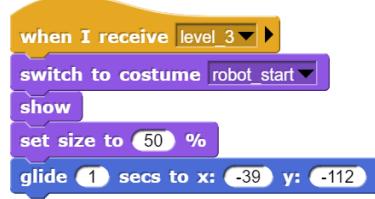
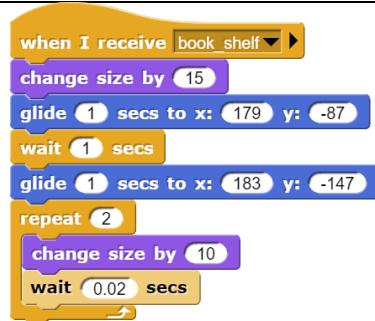
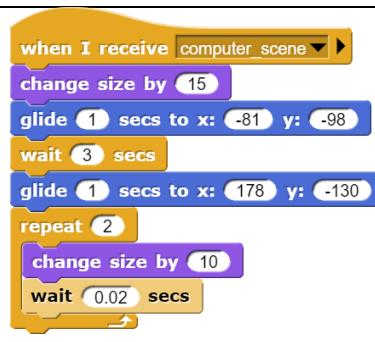
Table 3.1.5.8 Programming Concept of “Close_tab” Sprite in Level 3 of the Library

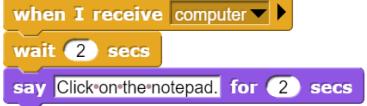
Code	Concept
	When flag button is clicked, size of “Close_tab” sprite is set to 50% of its original size and will be hidden on the stage to prevent interfering the opening animation and the following presentation of the application.
	When “Close_tab” sprite receives “computer_notepad” broadcasted by “Notepad” sprite, it will go to a specified position, which is at the top right of the computer tab displayed by the “Stage”.
	When “Close_tab” sprite is clicked, it will hide itself and set the value of the <u>“computer notepad”</u> to 0, meaning that the computer’s note pad is deactivated and closed. It will then broadcast “computer” to the “Stage” to switch the background costume to “Computer”, which shows a computer home screen without the notepad tab being opened. “level_3” is also broadcasted to the “Back” sprite and “Write_back” sprite so that the back button will be shown again on the stage that allows user to be redirected back to the previous scene, which is the computer zone. “Computer_back” is also broadcasted to the stage so that the music will not play when users click on back from computer Notepad tab to computer, the music will only play when users click on the “Computer_icon” which leads users to the computer screen.

Sprite Involved 9: “Robot” Sprite

Table 3.1.5.9 Programming Concept of “Robot” Sprite in Level 3 of the Library

Code	Concept
------	---------

 <pre> when I receive [level 3 v] switch to costume [robot_start v] show set size to [50 %] glide (1 sec) to x: (-39) y: (-112) </pre>	<p>When the thirteenth script receives “level_3” broadcasted by the “Button_level_3” sprite, the “Robot” sprite switches costume to “robot_start” and shows itself on the stage. It then shrinks in size by setting its size to 50% of the original size and glides 1 second to a certain position to show that the “Robot” sprite is moving and standing to the center of level 3 of the library.</p>
 <pre> when I receive [book shelf v] change (15) size glide (1 sec) to x: (179) y: (-87) wait (1 sec) glide (1 sec) to x: (183) y: (-147) repeat (2) [change (10) size wait (0.02) sec] </pre>	<p>When the fourteenth script receives “book_shelf” broadcasted by the “Book_shelf_icon” sprite, which means that users interact with the “Book_shelf_icon” sprite by clicking on it, the “Robot” sprite will change its size by enlarging itself by 15 and glide 1 second to a certain position. After 1 second, the “Robot” sprite will glide 1 second to another certain position and enlarge its size by 10 two times to show that the “Robot” sprite wants to introduce the book shelf in the library by writing some words from “Book_shelf_icon” sprite, which are “You can find some books here; there are also some books that are relevant to computing”.</p>
 <pre> when I receive [computer scene v] change (15) size glide (1 sec) to x: (-81) y: (-98) wait (3 sec) glide (1 sec) to x: (178) y: (-130) repeat (2) [change (10) size wait (0.02) sec] </pre>	<p>When the fifteenth script receives “computer_scene” broadcasted by the “Computer_icon” sprite, which means that users interact with the “Computer_icon” sprite by clicking on it, the “Robot” sprite will change its size by enlarging itself by 15 and glide 1 second to a certain position. After 3 second, the “Robot” sprite will glide 1 second to another certain position and enlarge its size by 10 two times to show that the “Robot” sprite wants to introduce the computer zone in the library by writing some words from “Computer_icon” sprite, which are “Computers are available for academic usage.”, “Free WIFI connection here guarantees a smooth browsing experience.”, “An ideal studying area for everyone.” and “Click on the computer icon for a brief overview.”.</p>

	<p>When the sixteenth script receives “computer” broadcasted by the “Computer_icon” sprite, which means that users interact with the “Computer_icon” sprite by clicking on it for the second click, the “Robot” sprite will wait for 2 seconds and say “Click on the notepad.” for 2 seconds, which guides users to interact with the “Notepad” sprite.</p>
	<p>When the seventeenth script receives “computer_notepad” broadcasted by the “Notepad” sprite, which means that users interact with the “Notepad” sprite by clicking on it, the “Robot” sprite will wait for 2 seconds and say “Type anything on your keyboard.” for 2 seconds, which asks users to input text by pressing any key on their keyboard. After 1 second, the “Robot” sprite will continue to say, “If you have finished typing, you can click on ‘close’ to close the tab”, which lets the users exit the tab of Notepad by clicking on ‘close’ button when they have completed their typing.</p>

Other Sprites Involved: Library” Sprite, “Level” Sprite, Button Representative Sprites, “End_tour” Sprite, “Back” Sprite, “Write_back” Sprite, “Write_click_icon” Sprite

These sprites will perform similar task throughout the entire application. To acknowledge what these following sprites will perform in this phase, please refer to:

“Library” Sprite: [Table 3.1.6.5 Programming Concept of “Button_level_1” Sprite Throughout the Application](#)

“Level” Sprite: [Table 3.1.6.3 Programming Concept of “Level” Sprite Throughout the Application](#)

“Button_level_1” Sprite: [Table 3.1.6.4 Programming Concept of “Button_level_1” Sprite Throughout the Application](#)

“Button_level_2” Sprite: [Table 3.1.6.5 Programming Concept of “Button_level_2” Sprite Throughout the Application](#)

“Button_level_3” Sprite: [Table 3.1.6.6 Programming Concept of “Button_level_3” Sprite Throughout the Application](#)

“End_tour” Sprite: [Table 3.1.6.9 Programming Concept of “End_tour” Sprite Throughout the Application](#)

“Back” Sprite: [Table 3.1.6.1 Programming Concept of “Back” Sprite Throughout the Application](#)

“Write_back” Sprite: [Table 3.1.6.2 Programming Concept of “Write_back” Sprite Throughout the Application](#)

“Write_click_icon” Sprite: [Table 3.1.6.8 Programming Concept of “Write_click_icon” Sprite Throughout the Application](#)

3.1.6 Throughout the Virtual Tour

Sprite Involved 1: “Back” Sprite

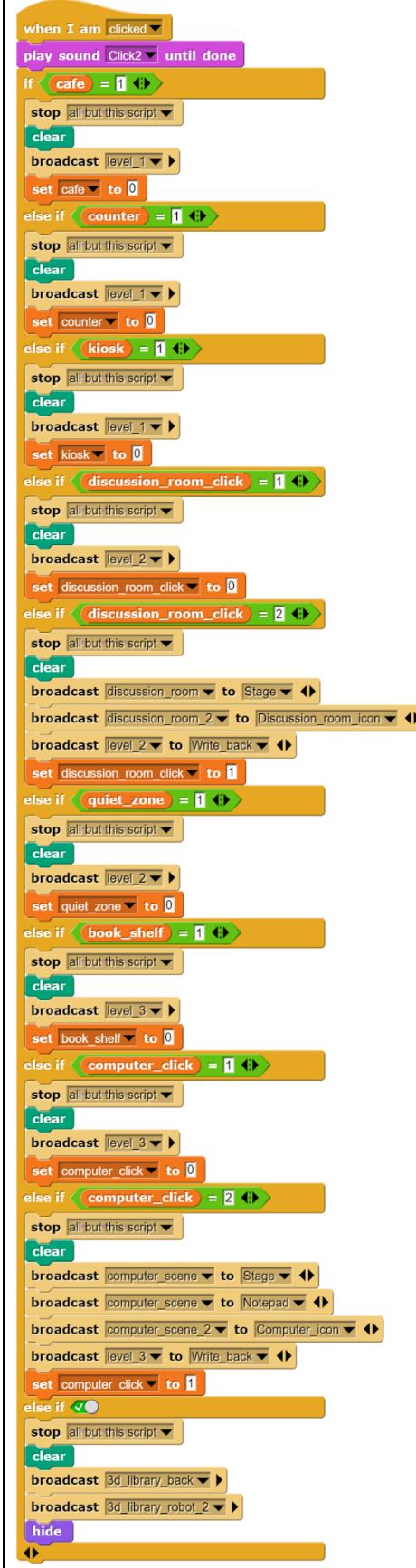
Phase Involved in:

- (a) [3.1.3 Level 1 of the Library](#)
- (b) [3.1.4 Level 2 of the Library](#)
- (c) [3.1.5 Level 3 of the Library](#)

Table 3.1.6.1 Programming Concept of “Back” Sprite Throughout the Application

Code	Concept
	When flag button is clicked, “Back” sprite will switch costume to “back” and will be hidden on the stage to prevent interfering the opening animation and the following presentation of the application.
	When “Back” sprite receive “level_1” broadcasted by “Button_level_1” sprite, it will show itself at a specific location, which is at the bottom left of the stage.

<pre>when I receive [level 2 v] show go to x: [-192] y: [-129]</pre>	<p>When “Back” sprite receive “level_2” broadcasted by “Button_level_2” sprite, it will show itself at a specific location, which is at the bottom left of the stage.</p>
<pre>when I receive [level 3 v] show go to x: [-192] y: [-129]</pre>	<p>When “Back” sprite receive “level_3” broadcasted by “Button_level_3” sprite, it will show itself at a specific location, which is at the bottom left of the stage.</p>
<pre>when I am [mouse-entered v] play sound [Click v] set size to (110%)</pre>	<p>When users’ mouse hovers over the “Back” sprite, it will play the sound “Click” and increase in size by 10%, implying that the back button is to be clicked.</p>
<pre>when I am [mouse-departed v] set size to (100%)</pre>	<p>When users’ mouse moves away from the “Back” sprite, it will return to its original size, indicating that the back button is not clicked.</p>



When “Back” sprite is clicked, it will play the sound “Click2”. In this script, the variables involved in each level including “cafe”, “kiosk”, and “counter” in level 1; “discussion_room_click” and “quiet_zone” in level 2; and “book_shelf” and “computer_click” in level 3 are duty-bound to keep track of the frequency of effective clicking on the respective icons on each level. Thus, “Back” sprite can redirect users back to the previous scene precisely based on the values stored in each variable. Frequency of effective clicking on the icon refers to as user proceed towards the next scene, the value of the relative variable will increase by one; when “Back” sprite is clicked and users are redirected back to the previous scene, the value of the variable will drop by 1.

In all the if and else if statements, the “stop all but this script” command is used so that when the “Back” sprite is clicked, the currently running script which is other than this script will be forced stop, and all the drawings done on the stage will be cleared through the clear command. After that, if and else if statement responsible for scene in level 1 will broadcast “level_1” to sprites involved in level 1 and to the “Stage” so that it changes to the relevant background costume. The if and else if statement involved in both level 2 and level 3 will also do the same.

However, for the variable responsible for discussion room in level 2, which is “discussion_room_click” and the variable responsible for computer zone in level 3, which is “computer_click” variable, are written slightly differently since two scene transitions will be made at the

both of those areas. In terms of discussion room, when the variable “discussion_room_click” equals to 2, means that two click has been made and the users is currently viewing the interior of the discussion room, “discussion_room” will be broadcasted to the “Stage” to switch background costume to “Discussion_room”. “discussion_room_2” is specifically broadcasted to the “Discussion_room_icon” sprite so that it will appear on the stage. “level_2” is broadcasted to the “Write_back” sprite so that it can rewrite the word “Back” below the “Back” sprite on the stage. In terms of the computer zone, when the variable “computer_click” equals to 2, means that two click has been made and the users is currently viewing the computer screen scene that allows users to type on the screen, “computer_scene” will be broadcasted to the “Stage” to switch background costume to “Computer_scene” and “Notepad” so that it is hidden on the stage. “computer_scene_2” is broadcasted to the “Computer_icon” sprite so that it will appear on the stage. “level_3” is broadcasted to the “Write_back” sprite so that it can rewrite the word “Back” below the “Back” sprite on the stage.

For the final else if statement, “3d_library_back” is broadcasted to “Library” so that users can be redirected back to view the 3D model of the library and make the selection of level. “3d_library_robot_2” is broadcasted to the “Robot” sprite so that it will appear at the bottom left of the stage.



When “Back” sprite receives “computer_notepad” broadcasted by the “Notepad” sprite, it will hide itself on

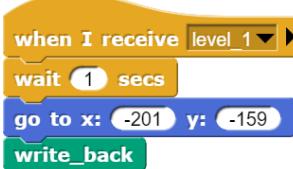
	stage when it switches to the next scene where users are allowed to type on the screen.
--	---

Sprite Involved 2: “Write_back” Sprite

Phase Involved in:

- (a) [3.1.3 Level 1 of the Library](#)
- (b) [3.1.4 Level 2 of the Library](#)
- (c) [3.1.5 Level 3 of the Library](#)

Table 3.1.6.2 Programming Concept of “Write_back” Sprite Throughout the Application

Code	Concept
	When flag button  is clicked, “Write_back” sprite will clear the stage and will be hidden on the stage to prevent interfering the opening animation and the following presentation of the application.
	When “Write_back” sprite receives “level_1” broadcasted by other sprites, particularly “Back” sprite and icon representative sprites in level 1, which are “Cafe_icon”, “Counter_icon”, and “Information_kiosk_icon”. It will write the word “Back” below the “Back” sprite at the bottom left of the stage via the “write_back” custom Pen block .
	When “Write_back” sprite receives “level_2” broadcasted by other sprites, particularly “Back” sprite and icon representative sprites in level 2, which are “Discussion_room_icon” and “Quiet_zone_icon”. It will write the word “Back” below the “Back” sprite at the bottom left of the stage via the “write_back” custom Pen block .

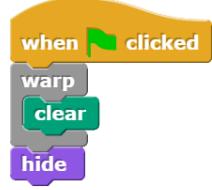
	<p>When “Write_back” sprite receives “level_3” broadcasted by other sprites, particularly “Back” sprite and icon representative sprites in level 3, which are “Book_shelf_icon” and “Computer_icon”. It will write the word “Back” below the “Back” sprite at the bottom left of the stage via the “write_back” custom Pen block.</p>
---	---

Sprite Involved 3: “Library” Sprite

Phase Involved in:

- (a) [3.1.3 Level 1 of the Library](#)
- (b) [3.1.4 Level 2 of the Library](#)
- (c) [3.1.5 Level 3 of the Library](#)

Table 3.1.6.3 Programming Concept of “Library” Sprite Throughout the Application

Code	Concept
	<p>When flag button  is clicked, “Library” sprite will rapidly clear the stage and will be hidden on the stage to prevent interfering the opening animation and the following presentation of the application.</p>



When “Library” sprite receives “3d_library_back” broadcasted by the “Back” sprite, it will swiftly clear the stage and start drawing the 3D model of the library. To start off, pen size is set to 5 and “Library” sprite is moved to a specific location. The “angle” and “size” variable that is also used in “Library_intro” sprite is set to its maximum value in respect with the one drawn by “Library_intro”. So, in this current displaying of 3D model of the library, the magnifying effect of the library is excluded. “Library” sprite will then draw the first cuboid for the first level by applying the values assigned to both the “angle” and “size” variable via the [“render_cuboid” custom Motion block](#). “Library” sprite then moves to the next location on the same x-axis but at a higher y-axis coordinate, which is where it is going to draw the second cuboid for the second level via the [“render_cuboid” custom block](#). “Library_intro” repeat the same thing did to the second level for the third level but at a much higher y-axis coordinate.

Once the 3D model of the library, it will directly move to the respective position and fill the building rapidly with the pen colour set. It will fill the first level of the library with a pen colour of , the second level with a pen colour of , and the third level with a pen colour of .

Sprite Involved 4: “Level” Sprite

Phase Involved in:

- (a) [3.1.2 Starting of the Virtual Tour](#)
- (b) [3.1.3 Level 1 of the Library](#)
- (c) [3.1.4 Level 2 of the Library](#)
- (d) [3.1.5 Level 3 of the Library](#)

Table 3.1.6.4 Programming Concept of “Level” Sprite Throughout the Application

Code	Concept
	When flag button  is clicked, all drawing done on the stage previously will be cleared. Hence, when users click on the flag button  again, the previous drawing drawn will not remain on the stage, disrupting the overall presentation of the opening animation.
	When the second script receives “level_of_library” broadcasted by “Library_intro” sprite, “Level” sprite will move to the given location, which is at the first cuboid drawn by the “Library_intro” sprite. “level” variable is created to be used in the “write_level” custom Pen block to identify which number of levels to be written. Through “write_level” custom Pen block , “Level” sprite will write “Level 1” at the first level of the 3D model of the library with respect to the value assigned to the “level” variable which is 1. “Level” sprite will wait for 0.05 seconds until the “Button_level_1” sprite appears and then finally write the “Level 1” on the 3D model. “level_of_library_2” is broadcasted to the next script.
	When the third script receives “level_of_library_2” broadcasted from the previous script, “Level” sprite will move to the given location, which is at the second cuboid drawn by the “Library_intro” sprite. “level” variable is assigned with the value of 2. Through “write_level” custom Pen block , “Level” sprite will write “Level 2” at the second level of the 3D model of the library with respect to the value assigned to the “level” variable.

	<p>“Level” sprite will wait for 0.05 seconds until the “Button_level_2” sprite appears and then finally write the “Level 2” on the 3D model. “level_of_library_3” is broadcasted to the next script.</p>
 <pre> when I receive level_of_library_3 go to x: -44 y: 83 wait 0.05 secs set level to 3 write_level broadcast click_level </pre>	<p>When the fourth script receives “level_of_library_3” broadcasted from the previous script, “Level” sprite will move to the given location, which is at the third cuboid drawn by the “Library_intro” sprite. “level” variable is assigned with the value of 3. Through “write_level” custom Pen block, “Level” sprite will write “Level 3” at the third level of the 3D model of the library with respect to the value assigned to the “level” variable. “Level” sprite will wait for 0.05 seconds until the “Button_level_3” sprite appears and then finally write the “Level 2” on the 3D model. “click_level” is broadcasted to the “Robot” sprite to switch costume and to the next script.</p>
 <pre> when I receive click_level go to x: -113 y: -165 write_click_level </pre>	<p>When the fifth script receives “click_level” broadcasted from the previous script, “Level” sprite will move to the specified location which is at the bottom of the 3d building drawn by the “Library_intro” sprite and write “Please select the level”. Notifying the users to select the desired level of the library to explore by clicking the button representative sprites,</p>

Sprite Involved 5: “Button_level_1” Sprite

Phase Involved in:

- (a) [3.1.2 Starting of the Virtual Tour](#)
- (b) [3.1.3 Level 1 of the Library](#)
- (c) [3.1.4 Level 2 of the Library](#)
- (d) [3.1.5 Level 3 of the Library](#)

Table 3.1.6.5 Programming Concept of “Button_level_1” Sprite Throughout the Application

Code	Concept
	When flag button  is clicked, “Button_level_1” sprite is hidden and sent to a specific location on the stage.
	When the second script receives “level_of_library” broadcasted by “Library_intro” sprite or “Library” sprite, the “Button_level_1” sprite will switch costume to “unselected” and show itself.
	When users’ mouse hovers over the “Button_level_1” sprite, it will display “Click” sound and switch costume to “selected”, implying that the button for level 1 of the library is to be selected if clicked.
	When users’ mouse moves away from the “Button_level_1” sprite, it will switch costume to “selected”, implying that the button for level 1 of the library is not selected.

Sprite Involved 6: “Button_level_2” Sprite

Phase Involved in:

- (a) [3.1.2 Starting of the Virtual Tour](#)
- (b) [3.1.3 Level 1 of the Library](#)
- (c) [3.1.4 Level 2 of the Library](#)
- (d) [3.1.5 Level 3 of the Library](#)

Table 3.1.6.6 Programming Concept of “Button_level_2” Sprite Throughout the Application

Code	Concept
	When flag button  is clicked, “Button_level_2” sprite is hidden and sent to a specific location on the stage.

	When the second script receives “level_of_library_2” broadcasted by “Level” sprite, the “Button_level_2” sprite will switch costume to “unselected” and show itself.
	When users’ mouse hovers over the “Button_level_2” sprite, it will display “Click” sound and switch costume to “selected”, implying that the button for level 1 of the library is to be selected if clicked.
	When users’ mouse moves away from the “Button_level_2” sprite, it will switch costume to “selected”, implying that the button for level 1 of the library is not selected.

Sprite Involved 7: “Button_level_3” Sprite

Phase Involved in:

- (a) [3.1.2 Starting of the Virtual Tour](#)
- (b) [3.1.3 Level 1 of the Library](#)
- (c) [3.1.4 Level 2 of the Library](#)
- (d) [3.1.5 Level 3 of the Library](#)

Table 3.1.6.7 Programming Concept of “Button_level_3” Sprite Throughout the Application

Code	Concept
	When flag button  is clicked, “Button_level_3” sprite is hidden and sent to a specific location on the stage.
	When the second script receives “level_of_library_3” broadcasted by “Level” sprite, the “Button_level_3” sprite will switch costume to “unselected” and show itself.
	When users’ mouse hovers over the “Button_level_3” sprite, it will display “Click” sound and switch costume to “selected”,

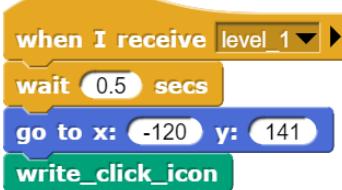
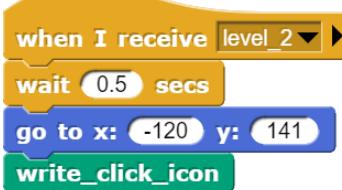
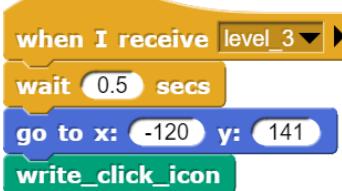
	implying that the button for level 1 of the library is to be selected if clicked.
	When users' mouse moves away from the "Button_level_3" sprite, it will switch costume to "selected", implying that the button for level 1 of the library is not selected.

Sprite Involved 8: "Write_click_icon" Sprite

Phase Involved in:

- (a) [3.1.3 Level 1 of the Library](#)
- (b) [3.1.4 Level 2 of the Library](#)
- (c) [3.1.5 Level 3 of the Library](#)

Table 3.1.6.8 Programming Concept of "Write_click_icon" Sprite Throughout the Application

Code	Concept
	When flag button  is clicked, "Write_click_icon" sprite is hidden.
	When "Write_click_icon" receives "level_1" broadcasted by the "Button_level_1" sprite, it will go to the specified location which is estimated as the center of the room to inform users to click on the respective icon to visit that area.
	When "Write_click_icon" receives "level_2" broadcasted by the "Button_level_2" sprite, it will go to the specified location which is estimated as the center of the room to inform users to click on the respective icon to visit that area.
	When "Write_click_icon" receives "level_3" broadcasted by the "Button_level_3" sprite, it will go to the specified location which is estimated as the center of the room to inform users to click on the respective icon to visit that area.

	When “Write_click_icon” receives any message broadcasted by each levels’ icon representative sprites, it will execute the “stop other scripts in sprite” command which stops all the scripts in this “Write_click_icon” sprite except for this current script running. This stops the “write_click_icon” custom Pen block from continuing writing when the users abruptly clicked on one of the icons presented on the stage even though the word “Please click on the icon” is not shown completely yet. These script act as a precautionary step that prevents the continuation of the script in “write_click_icon” custom Pen block to appear on the following scene, degrading the visualization and presentation of the application.

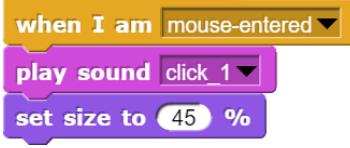
Sprite Involved 9: “End_tour” Sprite

Phase Involved in:

- (a) [3.1.3 Level 1 of the Library](#)
- (b) [3.1.4 Level 2 of the Library](#)
- (c) [3.1.5 Level 3 of the Library](#)

Table 3.1.6.9 Programming Concept of “End_tour” Sprite Throughout the Application

Code	Concept
	When “End_tour” sprite receives “3d_library_back” broadcasted by the “Back” sprite, it will go to the specified location, which is at the top right of the stage and show itself.

	When users' mouse hovers over the "End_tour" sprite, it will increase in size by 5% from the initially set 40% of its original size and play sound "click_1", meaning that end tour button is to be clicked.
	When users' mouse moves away from the "End_tour" sprite, it will set its size back to 40% of its original size, implying that end tour button is not clicked.
	When "End_tour" sprite receives "level_1" broadcasted by the "Button_level_1" sprite, it will hide itself on the stage to avoid disrupting the presentation of the level 1 of the library.
	When "End_tour" sprite receives "level_2" broadcasted by the "Button_level_2" sprite, it will hide itself on the stage to avoid disrupting the presentation of the level 1 of the library.
	When "End_tour" sprite receives "level_3" broadcasted by the "Button_level_3" sprite, it will hide itself on the stage to avoid disrupting the presentation of the level 1 of the library.

3.1.7 Rating of the Application

Sprite Involved 1: "End_tour" Sprite

Table 3.1.7.1 Programming Concept of "End_tour" Sprite in Rating of the Application

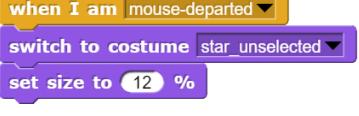
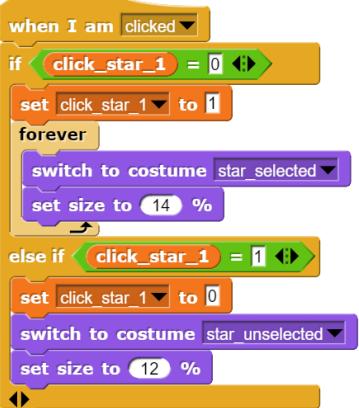
Code	Concept
	When users' mouse hovers over the "End_tour" sprite, it will increase in size by 5% from the initially set 40% of its original size and play sound "click_1".
	When users' mouse moves away from the "End_tour" sprite, it will set its size back to 40% of its original size.

 <pre> when I am clicked play sound [click v2] hide broadcast [end_tour v] </pre>	<p>When “End_tour” sprite is clicked, it will play sound “click_2” and hide itself on the stage. “end_tour” is broadcasted to the “Review” sprite.</p>
--	--

Sprite Involved 2, 3, 4, 5, 6: Rating representative sprites (star_1, star_2, star_3, star_4, star_5)

Table 3.1.7.2 Common Programming Concept for All Rating Representative Sprites

Code	Concept
 <pre> when green flag clicked hide </pre>	<p>When flag button  is clicked, rating representative sprites are hidden.</p>
 <pre> when I receive [show stars v] go to x: [-144] y: [-49] switch to costume [star unselected v] set size to (12 %) set [click_star_1 v] to [0] show </pre>	<p>When rating representative sprites receives “show_stars” broadcasted by “Review” sprite, they will move to their specified location on the same y-axis but different x-axis. They will line up in sequence with increasing x coordinate, from star_1 to star_5. After that, they will switch costume to “star_unselected” and set their size to 12% of their original size. Variables are created for each rating representative sprites, comprising of “<u>click_star_1</u>” variable for “star_1” sprite, “<u>click_star_2</u>” variable for “star_2” sprite, “<u>click_star_3</u>” variable for “star_3” sprite, “<u>click_star_4</u>” variable for “star_4” sprite, and “<u>click_star_5</u>” variable for “star_5” sprite. These variables are applied to record whether the respective star is selected or not, they will majorly affect the accuracy of the overall rating calculated by the “Submit” sprite. These variables are initialized</p>
 <pre> when I receive [show stars v] go to x: [-74] y: [-49] switch to costume [star unselected v] set size to (12 %) set [click_star_2 v] to [0] show </pre>	
 <pre> when I receive [show stars v] go to x: [-4] y: [-49] switch to costume [star unselected v] set size to (12 %) set [click_star_3 v] to [0] show </pre>	

 <pre> when I receive [show stars v] go to x: (136) y: (-49) switch to costume [star unselected v] set size to (12 %) show set [click star 5 v] to (0) broadcast [submit v] </pre>	<p>to the value of 0, meaning that the stars are not selected. Once done, these sprites will reveal themselves on the stage, opening the rating service to users.</p>
 <pre> when I am [mouse-entered v] switch to costume [star selected v] set size to (14 %) </pre>	<p>When mouse hovers over any of the rating representative sprites, that sprite will switch costume to “star_selected” and shrink in size to 14% of the original size, indicating that the star is to be clicked.</p>
 <pre> when I am [mouse-departed v] switch to costume [star unselected v] set size to (12 %) </pre>	<p>When mouse moves away from that sprite, it will switch costume back to “star_unselected”, implying that the star is not clicked.</p>
 <pre> when I am [clicked v] if [click star 1 v] = (0) then set [click star 1 v] to (1) forever switch to costume [star selected v] set size to (14 %) else if [click star 1 v] = (1) then set [click star 1 v] to (0) switch to costume [star unselected v] set size to (12 %) end </pre>	<p>When the rating representative sprites are clicked, what script to run will based on the value of their respective variable through the if and else if statement.</p> <p>If the value of the variable is 0, meaning that the star has not been clicked before, it will assign the variable to the value of 1, means that the first click is made; and switch to costume “star_selected” with size of 14% of the original size forever. This means that the</p>

```

when I am clicked
if click_star_2 = 0
  set click_star_2 to 1
  broadcast two_star_rate
forever
  switch to costume star_selected
  set size to 14 %
else if click_star_2 = 1
  stop all but this script
  set click_star_2 to 0
  broadcast undo_rating
  switch to costume star_unselected
  set size to 12 %

```

```

when I am clicked
if click_star_3 = 0
  set click_star_3 to 1
  broadcast three_star_rate
forever
  switch to costume star_selected
  set size to 14 %
else
  stop all but this script
  set click_star_3 to 0
  broadcast undo_rating
  switch to costume star_unselected
  set size to 12 %

```

```

when I am clicked
if click_star_4 = 0
  set click_star_4 to 1
  broadcast four_star_rate
forever
  switch to costume star_selected
  set size to 14 %
else
  stop all but this script
  set click_star_4 to 0
  broadcast undo_rating
  switch to costume star_unselected
  set size to 12 %

```

star is clicked, and it is highlighted. The forever loop allows the sprite to remain with the same costume, preventing other script like “when mouse entered” and “when mouse departed” script from switching the costume.

For rating representative sprites other than “star_1” sprite, they will broadcast their respective message, mainly to broadcast it to their previous rating representative sprites. The message includes “two_star_rate” from “star_2” sprite, “three_star_rate” from “star_3” sprite, “four_star_rate” sprite from “star_4” sprite, and “five_star_rate” from “star_5” sprite. So, when that sprite is clicked, the sprites in front of the current sprite will also be highlighted. Since “star_1” sprite is the first rating representative sprite, which means there is no more sprite placed in front of it, there is no need for it to broadcast any message as there is nothing to highlight.

Else, if the value of the variable is equated to 1, indicate that the first click is made and this is currently the second click, it will set the value of the variable to 0 and switch to “star_unselected” with its size set to 12% of its original size. This means that the star is unclicked, and it is no longer highlighted.

All rating representative sprites will broadcast “undo_rating”, among each other. So, when that sprite is unclicked, the sprites in front of the current sprite will also no longer be highlighted, or when the sprite at the front is unclicked, all the sprites after it will also no longer be highlighted.

 <pre> when I am clicked if click_star_5 = 0 set click_star_5 to 1 broadcast five_star_rate forever [switch to costume star_selected set size to 14 %] else set click_star_5 to 0 stop all but this script broadcast undo_rating switch to costume star_unselected set size to 12 %] </pre>	
 <pre> when I receive undo rating set click_star_1 to 0 switch to costume star_unselected set size to 12 % </pre>  <pre> when I receive undo rating stop all but this script set click_star_2 to 0 switch to costume star_unselected set size to 12 % </pre>	<p>When rating representative sprites receive the “undo_rating” broadcasted from one of the rating representative sprites, their respective variables which are “click_star_1” variable for “star_1” sprite, “click_star_2” variable for “star_2” sprite, “click_star_3” variable for “star_3” sprite, “click_star_4” variable for “star_4” sprite, and “click_star_5” variable for “star_5” sprite will all be set back to the value of 0. This means that rating made is undone and no rating is currently made. After that, they will switch costume back to “star_unselected”, meaning that they had returned to the state where the users have not click them.</p>
 <pre> when I receive undo rating stop all but this script set click_star_3 to 0 switch to costume star_unselected set size to 12 % </pre>  <pre> when I receive undo rating stop all but this script set click_star_4 to 0 switch to costume star_unselected set size to 12 % </pre>	
 <pre> when I receive undo rating stop all but this script set click_star_5 to 0 broadcast undo_rating switch to costume star_unselected set size to 12 % </pre>	

	When the rating representative sprites receive “overall” rating, they will hide themselves on the stage so it will disrupt the presentation of the calculated overall rating.
--	---

Table 3.1.7.3 Receiving Broadcasted Message for Specific Rating Representative Sprites

“two_star_rate” Message	
Code	Concept
	When the “star_1” sprite receives “two_star_rate”, it set its “click_star_1” variable to the value of 1 and switch to costume “star_selected” with size 14% of their original size in a forever loop. This means that the first star is also clicked and highlighted as the second star is clicked and highlighted.
“three_star_rate” Message	
	When the “star_1” and “star_2” sprites receive “three_star_rate”, they set their respective variables which are “click_star_1” variable for “star_1” sprite and “click_star_2” variable for “star_2” sprite to the value of 1 and switch to costume “star_selected” with size 14% of their original size in a forever loop. This means that the first and second stars are also clicked and highlighted as the third star is clicked and highlighted.
“four_star_rate” Message	
Code	Concept
“star_1” Sprite	

```

when I receive [four_star_rate v]
set [click_star_1 v] to [1]
forever
  switch to costume [star_selected v]
  set size to [14 %]

```

“star_2” Sprite

```

when I receive [four_star_rate v]
set [click_star_2 v] to [1]
forever
  switch to costume [star_selected v]
  set size to [14 %]

```

“star_3” Sprite

```

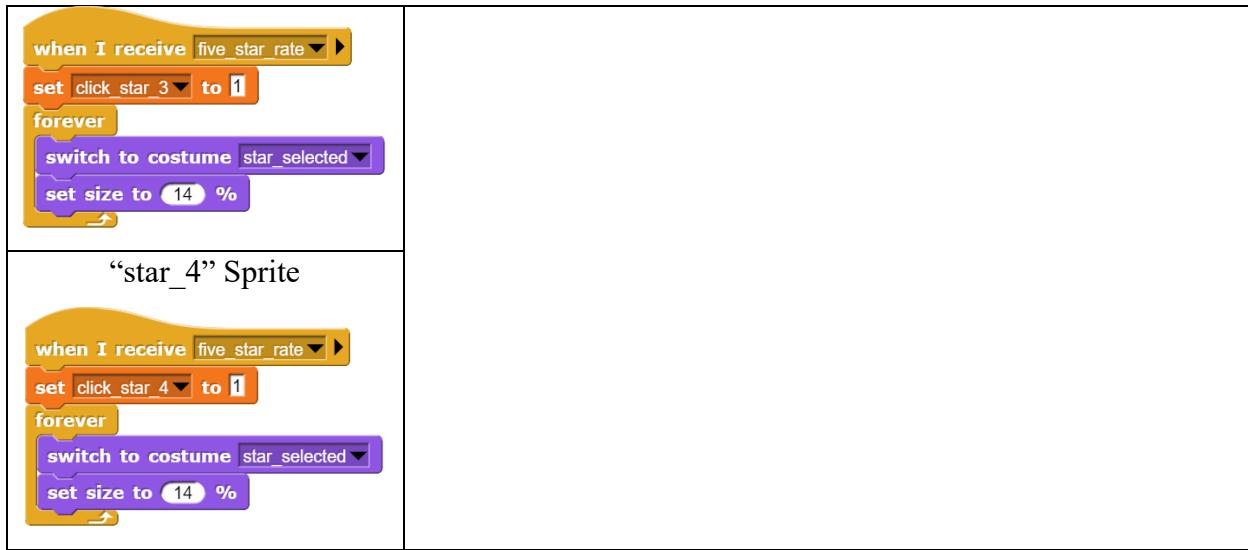
when I receive [four_star_rate v]
set [click_star_3 v] to [1]
forever
  switch to costume [star_selected v]
  set size to [14 %]

```

When the “star_1”, “star_2”, and “star_3” sprites receive “four_star_rate”, they set their respective variables which are [“click_star_1”](#) variable for “star_1” sprite, [“click_star_2”](#) variable for “star_2” sprite, and [“click_star_3”](#) variable for “star_3” sprite to the value of 1 and switch to costume “star_selected” with size 14% of their original size in a forever loop. This means that the first, second, and third stars are also clicked and highlighted as the fourth star is clicked and highlighted.

“five_star_rate” Message

Code	Concept
“star_1” Sprite	When the “star_1”, “star_2”, “star_3”, and “star_4” sprites receive “five_star_rate”, they set their respective variables which are “click_star_1” variable for “star_1” sprite, “click_star_2” variable for “star_2” sprite, “click_star_3” variable for “star_3” sprite, and “click_star_4” variable for “star_4” sprite to the value of 1 and switch to costume “star_selected” with size 14% of their original size in a forever loop. This means that the first, second, third, and fourth stars are also clicked and highlighted as the fifth star is clicked and highlighted.
“star_2” Sprite	
“star_3” Sprite	



Sprite Involved 6: “Submit” Sprite

Table 3.1.7.4 Programming Concept of “Submit” Sprite in Rating of the Application

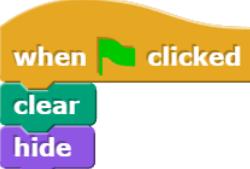
Code	Concept
<pre> when green flag clicked hide </pre>	When flag button is clicked, “Submit” sprite will hide itself on the stage preventing from interrupting the presentation of the opening animation and the following scenes.
<pre> when I receive [end_tour v] set [rating_list v] to [list v] set [review v] to [0] hide </pre>	When “Submit” sprite receives “end_tour” broadcasted by “End_tour” sprite, a rating list will be created that will be used to store all the ratings given for the three review questions. A <u>“review”</u> variable is also created to note down the number of review questions given out. It then stays hidden on the stage.
<pre> when I receive [show_stars v] switch to costume [submit_rating v] set size to [40 %] wait [2 secs] show </pre>	When “Submit” sprite receives “show_stars” broadcasted by the “Review” sprite, it will switch costume to “submit_rating” with size set to 40% of its original size. It then shows itself on stage.
<pre> when I am mouse-entered switch to costume [submit_rating v] set size to [45 %] </pre>	When users’ mouse hovers over “Submit” sprite, it switches costume to “submit_rating” with a 5% increase in size from the

	previously set 40% size, implying that the “Submit” sprite is to be clicked.
<pre> when I am mouse-departed switch to costume submit_rating set size to (40 %) </pre>	When users’ mouse moves away from the “Submit” sprite, it will switch costume back to “submit_rating” with a size of 40% of its original size, meaning that the “Submit” sprite is not clicked.
<pre> when I am clicked repeat if (click_star_5) = (1) then add (5) to (rating_list) else if (click_star_4) = (1) then add (4) to (rating_list) else if (click_star_3) = (1) then add (3) to (rating_list) else if (click_star_2) = (1) then add (2) to (rating_list) else if (click_star_1) = (1) then add (1) to (rating_list) else if (green flag is set) then add (0) to (rating_list) end change (review) by (1) if (review) = (1) then broadcast (review_1) else if (review) = (2) then broadcast (review_2) else if (review) = (3) then broadcast (review_3) end set (overall rating) to (round (sum (rating_list)) / 3) broadcast (overall_rating) end </pre>	<p>When “Submit” sprite is clicked, the value stored in each rating representative sprites’ variable, which are “click_star_1”, “click_star_2”, “click_star_3”, “click_star_4”, and “click_star_5” variables will be evaluated. The values of each variable will be evaluated in descending order. For instance, if they are evaluated in descending order, and “click_star_3” does not equal to 1, then rating will not be 3 stars and above, it will either be 1 star or 2 stars if “click_star_1” or both “click_star_2” and “click_star_1” variables equal to 1. However, if they are evaluated in ascending order, and “click_star_3” equals to 1, the condition of the else if statement is satisfied and will run the script even though the rating can be 3 stars or more than that. So, the script evaluates whether the highest rating is achieved and satisfies the if statement or not and then gradually decrease the rating until the lowest to identify the exact rating.</p> <p>If “click_star_5” equals to 1, 5 is added to the “rating_list” list; if “click_star_4” equals to 1, 4 is added to the “rating_list” list; if “click_star_3” equals to 1, 3 is added to the “rating_list” list; if “click_star_2” equals to 1, 2 is added to the “rating_list” list; if “click_star_1” equals to 1, 1 is added to the “rating_list” list. Else, 0 is added to the “rating_list” list.</p> <p>After that, the value of the variable “review” will increase by 1, meaning that one review question is given and one rating is stored in the “rating_list” list.</p>

	<p>If the value of “<u>review</u>” variable is equated to 1, means that one review question is given out, it will broadcast “review_2” to “Review” sprite to ask the second review question. If the value of “<u>review</u>” variable is equated to 2, means that two review questions are given out, it will broadcast “review_3” to “Review” sprite to ask the last review question. Else, which normally refers to when value of “<u>review</u>” variable equals to 3, when all three review questions have been given out, it creates another variable “<u>overall_rating</u>” to store the average value of the rating given by the users. Then, it broadcast “overall_rating” to the next script and to the “Review” sprite to display the overall rating on the stage.</p>
	<p>When “Submit” sprite receives “overall_rating” broadcasted by from the final else if statement in the previous script, it will hide itself on the stage.</p>

Sprite Involved 7: “Review” Sprite

Table 3.1.7.5 Programming Concept of “Review” Sprite in Rating of the Application

Code	Concept
	<p>When flag button  is clicked, “Review” sprite clears the stage and is hidden to avoid interfering the presentation of the opening animation and the following scenes.</p>

```

when I receive end_tour
stop all but this script
clear
wait [2 secs]
switch to costume [star v]
set size to [20 %]
stop all but this script
hide
review_question_1
broadcast show_stars

```

When “Review” sprite receives “end_tour” broadcasted by the “End_tour” sprite, it will initiate the rating section by stopping other running script that might be running which usually happens when users abruptly click on the “End_tour” sprite even though the other sprites might be still halfway running their script to showcase the 3D model of the library. This prevents the presentation of the 3D model of the library to continue in the rating section. After that command is executed, it will clear the stage and switch costume to “star” with size of 20% of its original size. Finally, it hides itself and write the first review question on the stage via the [“review_question_1” custom Pen block](#). “show_stars” is broadcasted to the “Submit” sprite and the rating representative sprites, which are “star_1”, “star_2”, “star_3”, “star_4”, and “star_5” so that they will be displayed on stage.

```

when I receive review_2
clear
stop all but this script
broadcast show_stars
hide
review_question_2

```

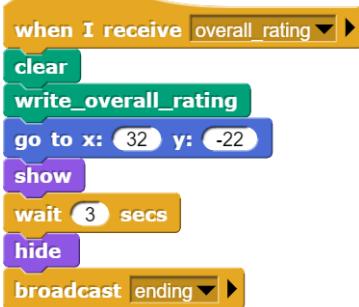
When “Review” sprite receives “review_2” broadcasted by the “Submit” sprite, it will clear the stage and stop all currently running script except for this script. This helps to stop the forever loop that might be executed by the rating representative sprites when they are clicked to give the rating for the previous review question. “show_stars” is broadcasted to the “Submit” sprite and the rating representative sprites, which are “star_1”, “star_2”, “star_3”, “star_4”, and “star_5” so that they will be displayed on stage. It will then hide itself and write the second review question on the stage via the [“review_question_2” custom Pen block](#).

```

when I receive review_3
clear
stop all but this script
broadcast show_stars
hide
review_question_2

```

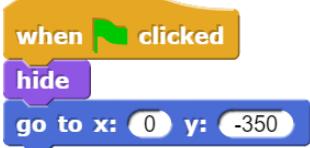
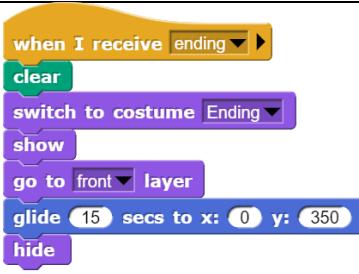
When “Review” sprite receives “review_3” broadcasted by the “Submit” sprite, it will clear the stage and stop all currently running script except for this script. This helps to stop the forever loop that might be executed by the rating representative sprites when they are clicked to give the rating for the previous review question. “show_stars” is broadcasted to the “Submit” sprite and

	<p>the rating representative sprites, which are “star_1”, “star_2”, “star_3”, “star_4”, and “star_5” so that they will be displayed on stage. It will then hide itself and write the second review question on the stage via the “review_question_3” custom Pen block.</p>
 <pre> when I receive [overall_rating v] clear write_overall_rating go to x: 32 y: -22 show wait [3] secs hide broadcast [ending v] </pre>	<p>When “Review” sprite receives “overall_rating” broadcasted by the “Submit” sprite, it will clear the stage and write the overall rating calculated by the “Submit” sprite at the middle of the stage via the “write_overall_rating” custom Pen block. Later, it will show itself on the stage, beside the value of the rating calculated. After 3 seconds, it will hide itself and broadcast “ending” to the “Ending” sprite to start the ending animation.</p>

3.1.8 Ending Animation

Sprite Involved 1: “Ending” Sprite

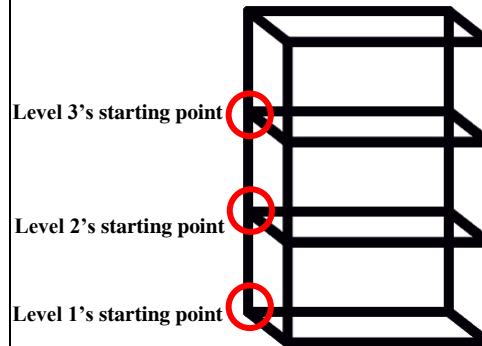
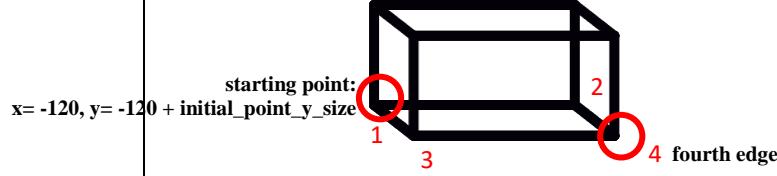
Table 3.1.8.1 Programming Concept of “Ending” Sprite in Ending Animation

Code	Concept
 <pre> when green flag clicked hide go to x: 0 y: -350 </pre>	<p>When flag button  is clicked, “Ending” sprite hide itself and move to its specified position on the stage.</p>
 <pre> when I receive [ending v] clear switch costume to [Ending v] show go to [front v] layer glide [15] secs to x: 0 y: 350 hide </pre>	<p>When “Ending” sprite receives “ending” broadcasted by the “Review” sprite, it will clear the stage and switch costume to “Ending”. It then shows itself and go to the front layer of the stage, gliding for 15 seconds from the bottom to the top. Once done, it will hide itself on the stage.</p>

3.1.9 Custom Block Used Throughout the Application

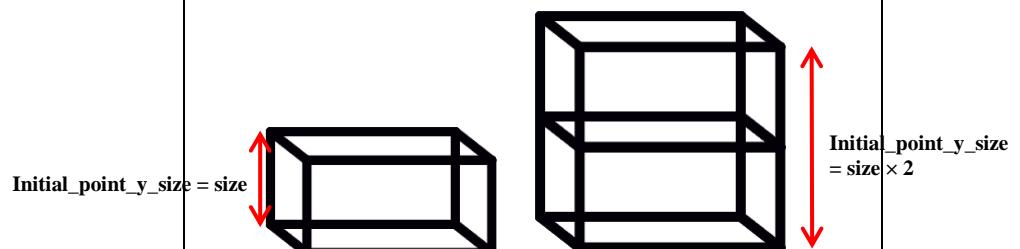
Table 3.1.9.1 Motion Custom Block

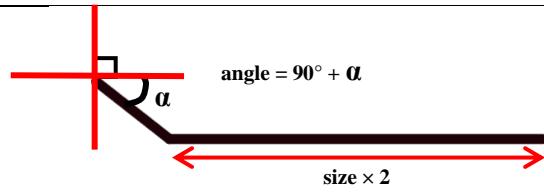
Code	Concept
<pre>greetings [+ greetings +] switch to costume robot_start set volume to (65) % play sound Robot_talking_voice until done switch to costume robot_wave wait (0.5) secs switch to costume robot_wave_2 wait (0.05) secs say Hi there! for (2) secs switch to costume robot_wave wait (0.05) secs switch to costume robot_start ask What's your name? and wait say [join Welcome to our library's grand opening day. answer for (2) secs wait (1) secs say Let's have a tour around the library! for (2) secs wait (1) secs</pre>	<p>This block will switch to the costume “robot_start” which shows the users that the “Robot” sprite is on and have a 65% volume of sound shows that the “Robot” sprite is talking to users. When “Robot” sprite is talking, it will change the costumes from “robot_wave” to “robot_wave_2” and back to “robot_wave” and “robot_start” after the “Robot” sprite say, “Hi there!” to the users. After that, the “Robot” sprite will ask users for name and welcome the users with users’ own name. Lastly, the “Robot” sprite will invite users to have a virtual tour around the library.</p>
<pre>+ determine_max_x + go to x: -120 y: -120 + initial_point_y_size point in direction angle move (size / 2) steps change x by (size * 2) set [max x] to [x position]</pre> <p>The image shows a Scratch script for the “determine_max_x” custom block. It starts with a yellow hat block labeled “+ determine_max_x +”. Inside, it has a “go to x: -120 y: -120 + initial_point_y_size” block, followed by a “point in direction angle” block, a “move (size / 2) steps” block, a “change x by (size * 2)” block, and finally a “set [max x] to [x position]” block. A red box highlights the “change x by (size * 2)” block. A yellow arrow points from the end of this block to a 3D model of a cuboid on the stage, indicating the path of the script's movement.</p>	<p>This block is created to determine the maximum x-axis of the 3D model of the library and assign that value identified to the variable “max_x”.</p> <p>To do so, the script follows the drawing path of the cuboid in “render_cuboid” custom Looks block. It first moves to the starting coordinate where the “render_cuboid” custom Looks block starts drawing the cuboid. The y-coordinate of the starting point for each level varies. So, “initial_point_y_size” variable is created to specify the y-coordinate of the starting point for each level, and thus it can move to that starting coordinate by referring to the value assigned to it. The x-coordinate of the starting point is fixed to -120.</p>



From the starting point, it will move to the fourth edge of the cuboid, where its x-coordinate is the maximum x-axis of the cuboid. Once it reaches that point, the x-coordinate will be stored in the “[max_x](#)” variable.

For level 1, value of “[initial_point_y_size](#)” variable will be 0 since it’s the first level. For level 2, value of “[initial_point_y_size](#)” will equals to the “[size](#)” variable used to define the size of the cuboid, which is the height of one cuboid. For level 3, the value of “[initial_point_y_size](#)” will be twice the value of “[size](#)” variable, which is twice the height of one cuboid.





To go to the fourth edge, the sprite will point in direction at a degree of the value of the “[angle](#)” variable used to define the angle to draw the cuboid; then move a few steps about half the value of the “[size](#)” variable. After that, it changes its x-coordinate by twice the value of the “[size](#)” variable, which is generally the length of the cuboid, and then it reached the fourth edge.

Table 3.1.9.2 Looks Custom Block

Code	Concept
<p>Glitch</p> <pre> + Glitch + repeat (3) switch to costume [glitch_1 v] wait (0.05) secs repeat (6) next costume wait (0.05) secs end </pre>	<p>This block is created to show the effect of glitching in the opening scene by repeating a change in the costume to the next costume.</p>
<p>render_cuboid</p>	<p>This block is used to draw the cuboid, where one cuboid represents one level of the library. In this block, the “size” and “angle” variable are used. The “size” variable is used to specify the length of each line to draw, while the “angle” variable is used to position the sprite in the correct orientation especially when drawing a slanted line.</p>

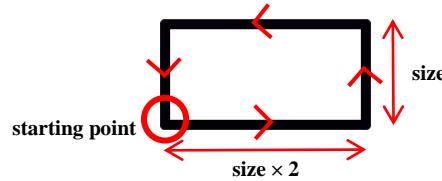
```

+ render_cuboid +
pen up
set pen size to 5
set pen color to black
pen down
change x by size × 2
change y by size
change x by size × -2
change y by size × -1
point in direction angle
move size / 2 steps
change x by size × 2
point in direction 180 + angle
move size / 2 steps
point in direction angle
move size / 2 steps
change x by size × -2
change y by size
point in direction 180 + angle
move size / 2 steps
point in direction angle
move size / 2 steps
change x by size × 2
change y by size × -1
change y by size × 1
point in direction 180 + angle
move size / 2 steps
pen up

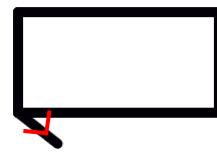
```

To start drawing the cuboid, the pen is brought up and its size is set to 5 with a pen colour of black. After that, the pen is brought down and start moving.

For the cuboid, the length is twice the value of the “size” variable, and the height is equal to the value of the “size” variable. To start, it will first draw a rectangle. From the starting point, it increases the x-coordinate by its length, increases the y-coordinate by its height, decreases the x-coordinate by its length, and decreases the y-coordinate by its height.

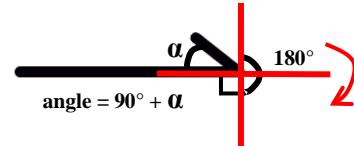
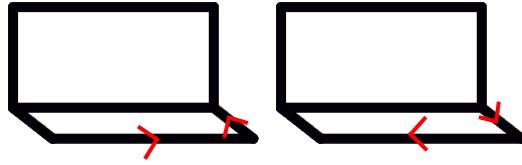


After the first rectangle is drawn, it will draw the base. It starts with drawing a slanted line by pointing in the direction of the “angle” variable and move a few steps about half the value of the “size” variable.

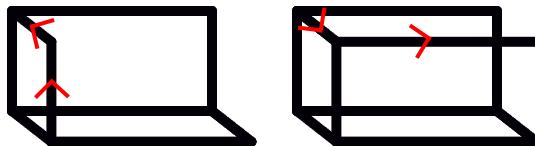


After that, it increases the x-coordinate by its length; and will draw another slanted line by pointing in the direction of the angle of 180° added with the value of the “angle” variable ($180^\circ + \text{angle}$ variable), and then move a few steps about half the value of the “size” variable. It then returns to the previous position using the same path by pointing in the direction of the “angle” variable and move a few

steps about half the value of the “size” variable; and decreases the x-coordinate by its length.



Once the base is drawn, it will draw the top surface. It increases the y-coordinate by its height, draw another slanted line by pointing in the direction of the angle of 180° added with the value of the “angle” variable ($180^\circ + \text{“angle”}$ variable), and then move a few steps about half the value of the “size” variable. It then returns by pointing in the direction of the “angle” variable and move a few steps about half the value of the “size” variable and increase the x-coordinate by its length.



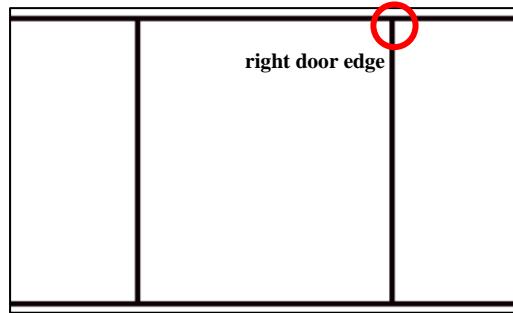
Subsequently, it will decrease its y-coordinate by its height and increase it by its height again. Finally, it draws the final slanted line by pointing in direction of the angle of 180° added with the value of the “angle” variable ($180^\circ + \text{“angle”}$ variable), and then move a few steps about half the value of the “size” variable.

draw_outline	<p>This block is used to draw the outline of the wall. The pen is lifted and go to the specified location, which is at the first edge. Pen size is set to 5 with pen colour of black. Pen is down and increase its x-coordinate by 480 to draw the upper border of the wall. Pen is lifted and move to the specified location, which is the third edge. Pen is down and increase its coordinate by 480. Pen is up and go to the specified location, which is at the left door edge. Pen is down and decrease its y-coordinate by 270. Pen is up and move to the specified location, which is at the right door edge. Pen is down and decrease its y-coordinate by 270. The outline of the door is drawn, and pen is lifted.</p> <pre>+ draw_outline+ pen up go to x: -240 y: 135 set pen size to 5 set pen color to black pen down change x by 480 pen up go to x: -240 y: -135 pen down change x by 480 pen up go to x: -120 y: 135 pen down change y by -270 pen up go to x: 120 y: 135 pen down change y by -270 pen up</pre>

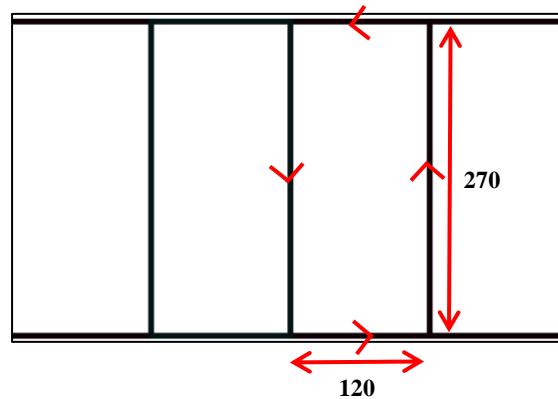
draw_door_right

```
+ draw_door_right +  
go to x: 120 y: 135  
pen up  
set pen size to 5  
set pen color to black  
pen down  
point in direction angle × -1  
move 120 steps  
change y by -270  
point in direction 180 + angle × -1  
move 120 steps  
change y by 270  
warp  
set pen color to brown  
pen up  
go to x: 120 y: 135  
pen down  
point in direction angle × -1  
move 120 steps  
repeat (270)  
change y by -1  
point in direction 180 + angle × -1  
move 120 steps  
move -120 steps  
pen up  
go to x: 120 y: 135  
set pen size to 5  
set pen color to black  
pen down  
point in direction angle × -1  
move 120 steps  
change y by -270  
point in direction 180 + angle × -1  
move 120 steps  
change y by 270  
pen up
```

This block is used to draw the outline of the right door. For starters, it goes to the edge of the right door, lifted the pen, and set the size to 5 with pen colour of black.



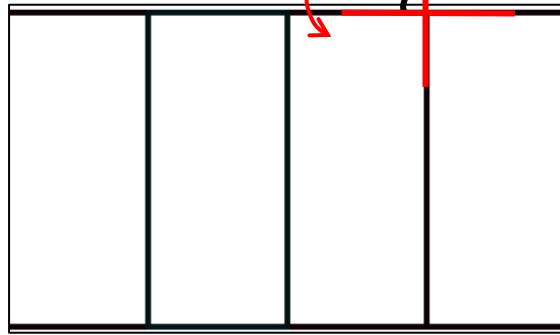
After that, it starts to draw a rectangular outline. It points in the direction of the value of the “angle” variable set before in the opposite direction and move 120 steps, that is the length of the door. It decreases its y-coordinate by 270, which is the height of the wall and the door. Then, it points in the opposite direction of the angle of 180° added with the value of the “angle” variable ($180^\circ + \text{angle}$) and move 120 steps. It increases its y-coordinate by 270, and the outline of the right door is drawn.



if angle = 90°

opposite direction, thus $\alpha = \text{angle} \times -1$

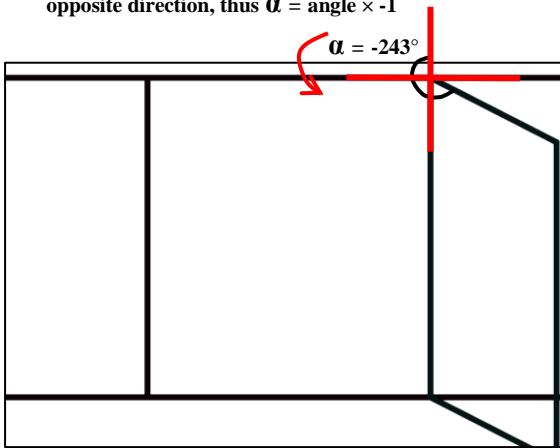
$$\alpha = -90^\circ$$



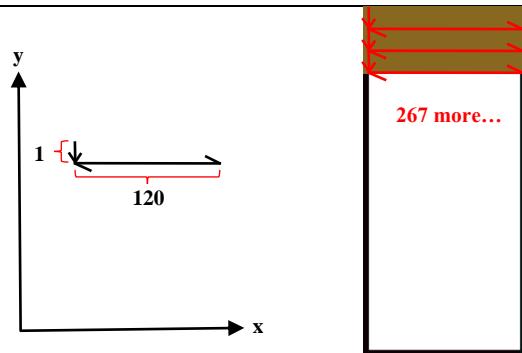
if angle = 243°

opposite direction, thus $\alpha = \text{angle} \times -1$

$$\alpha = -243^\circ$$

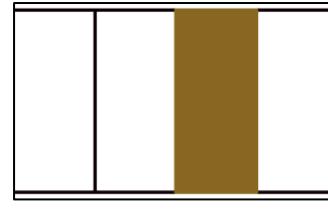


To colour the right door, the pen colour is set to the colour . The pen is lifted, and it moves to the edge of the right door. The pen is down, and it points in the opposite direction of the value of the “[angle](#)” variable and move 120 steps. A loop is created that iterates 270 times. In each loop, it decreases its y-coordinate by 1, point in the opposite direction of the angle of 180° added with the value of the “[angle](#)” variable ($180^\circ + \text{“angle”}$ variable), move 120 steps, and then move 120 steps back.



motion of sprite in each loop

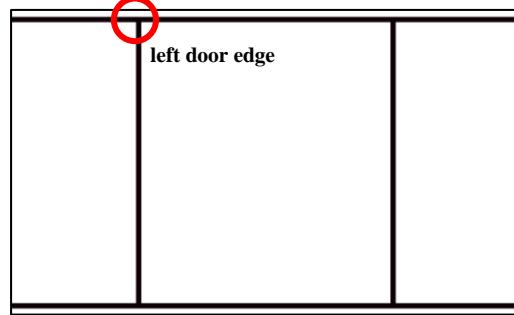
To cover the colourations done on the outline of the right door, the drawing of the outline of the right door is repeated. Once done, the pen is lifted and a coloured right door is drawn.



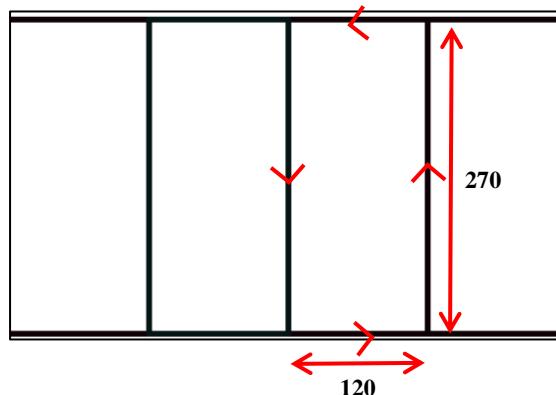
draw_door_left

```
+ draw_door_left +  
go to x: -120 y: 135  
pen up  
set pen size to 5  
set pen color to black  
pen down  
point in direction angle  
move 120 steps  
change y by -270  
point in direction 180 + angle  
move 120 steps  
change y by 270  
warp  
set pen color to brown  
pen up  
go to x: -120 y: 135  
pen down  
point in direction angle  
move 120 steps  
move -120 steps  
repeat (270)  
  change y by -1  
  point in direction angle  
  move 120 steps  
  move -120 steps  
pen up  
go to x: -120 y: 135  
set pen size to 5  
set pen color to black  
pen down  
point in direction angle  
move 120 steps  
change y by -270  
point in direction 180 + angle  
move 120 steps  
change y by 270  
pen up
```

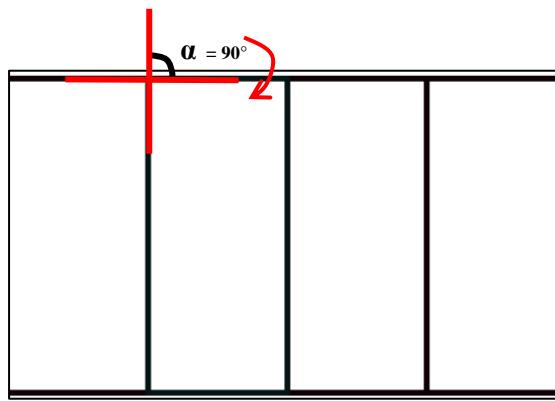
This block is used to draw the outline of the left door. For starters, it goes to the edge of the left door, lifted the pen, and set the size to 5 with pen colour of black.



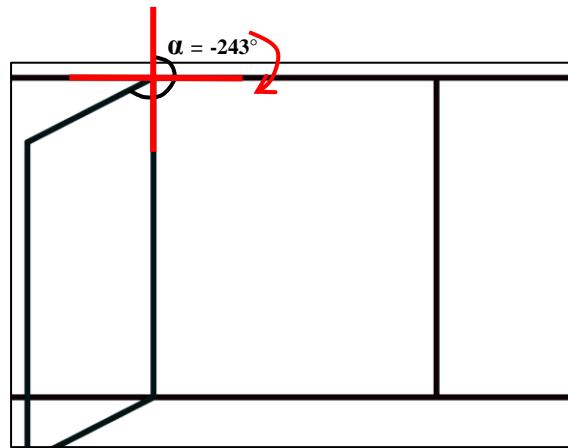
After that, it starts to draw a rectangular outline. It points in the direction of the value of the "angle" variable set before and move 120 steps, that is the length of the door. It decreases its y-coordinate by 270, which is the height of the wall and the door. Then, it points in the direction of the angle of 180° added with the value of the "angle" variable ($180^\circ + \text{angle}$) and move 120 steps. It increases its y-coordinate by 270, and the outline of the left door is drawn.



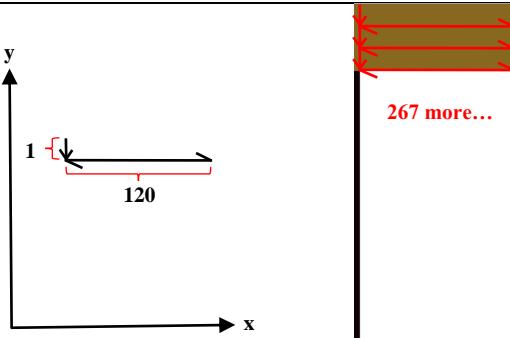
if angle = 90°



if angle = 243°

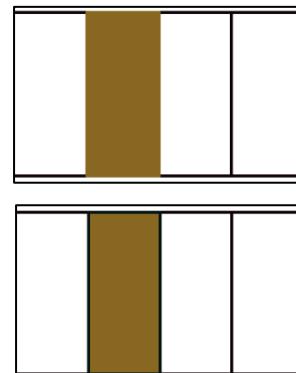


To colour the left door, the pen colour is set to the colour █. The pen is lifted, and it moves to the edge of the left door. The pen is down, and it points in the direction of the value of the “angle” variable and move 120 steps. A loop is created that iterates 270 times. In each loop, it decreases its y-coordinate by 1, point in the direction of the angle of 180° added with the value of the “angle” variable ($180^\circ + \text{“angle”}$ variable), move 120 steps, and then move 120 steps back.



motion of sprite in each loop

To cover the colourations done on the outline of the left door, the drawing of the outline of the left door is repeated. Once done, the pen is lifted and a coloured left door is drawn.



draw_info_area

This is used to draw the information box that act as a platform to present the information about a particular area. In this block, “x” and “y” variables are created to define the starting coordinate to draw the information box; while “height_of_info_area” variable is created to defined how long in terms of y-axis will the information box be based on how much information is needed to be given. The length of the information box is fixed to 302.

To start, it moves to the starting coordinate. Pen is lifted, setting pen colour to and pen size to 3. Pen

```

draw_info_area
pen up
go to x: x y: y
set pen color to □
set pen size to 3
pen down
go to x: x + 302 y: y
go to x: x + 302 y: y + height_of_info_area × -1
go to x: x y: y + height_of_info_area × -1
go to x: x y: y
pen up
warp
set pen color to □
go to x: 29 y: -133
fill
pen up
go to x: x y: y
set pen color to □
set pen size to 3
pen down
glide 1 secs to x: x + 302 y: y
glide 1 secs to x: x + 302 y: y
y + height_of_info_area × -1
glide 1 secs to x: x y: y + height_of_info_area × -1
glide 1 secs to x: x y: y
pen up

```

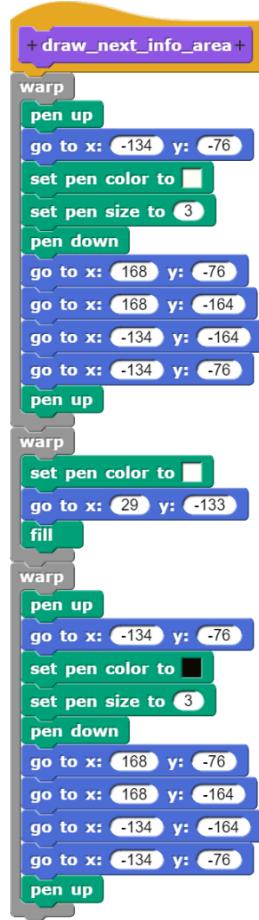
is put down and increase its x-coordinate by 302. It decreases its y-coordinate by the value of the “[height of info area](#)”, decreases its x-coordinate by 302, and return to the starting point. The pen is lifted. A rectangle is drawn. After that, the rectangle is filled with the colour □.



Successively, the border of the information box will be drawn. Instead of moving to one point, it glides, presenting a gradually appearing line bordering the information box. The pen is lifted and move to the starting coordinate. It glides to the point with its x-coordinate increased by 302 and glides to the second point with its y-coordinate decreased by the value of the “[height of info area](#)”. It glides to the third point by decreasing its x-coordinate by 302 and return to the starting point. The pen is lifted. A rectangular border is drawn.



draw_next_info_area

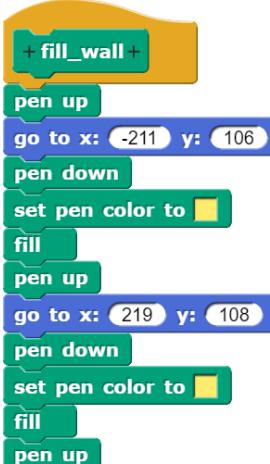
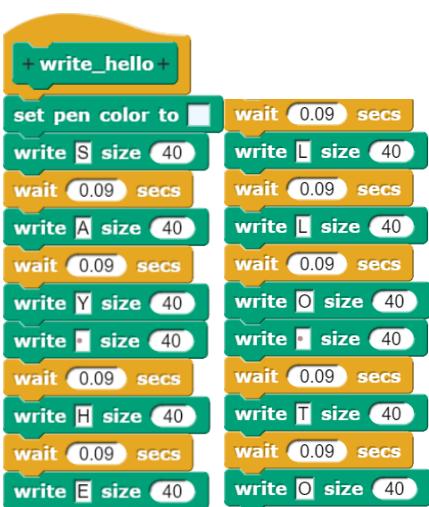
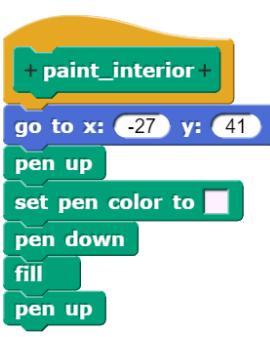


This block is a successive block of the "["draw_info_area"](#) custom Pen block", a slightly simplified version of it in terms of drawing the border of the information box. Instead of gliding from one point to another, it moves quickly to from one point to the next.

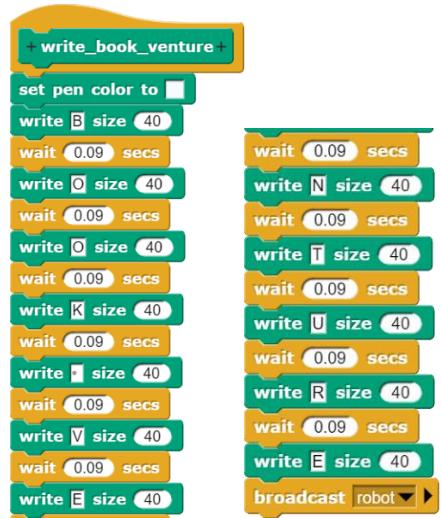
Firstly, the white area which is the information area is drawn, similar to what is done in the "["draw_info_area"](#) custom Pen block". It moves to the starting coordinate. Pen is lifted, setting pen colour to white and pen size to 3. Pen is put down and increase its x-coordinate by 302. It decreases its y-coordinate by the 88, decreases its x-coordinate by 302, and return to the starting point. The pen is lifted. A rectangle is drawn. After that, the rectangle is filled with the colour white.

Following up is the border of the information box. It repeats what is basically done to draw outline of the information area. It moves to the starting coordinate. Pen is lifted, setting pen colour to black and pen size to 3. Pen is put down and increase its x-coordinate by 302. It decreases its y-coordinate by the 88, decreases its x-coordinate by 302, and return to the starting point. The pen is lifted. A rectangular border is drawn.

Table 3.1.9.3 Pen Custom Block

Code	Concept
fill_wall  <pre data-bbox="204 382 474 846"> + fill_wall + pen up go to x: -211 y: 106 pen down set pen color to yellow fill pen up go to x: 219 y: 108 pen down set pen color to yellow fill pen up </pre>	This block uses the pen to colour the walls with the colour yellow in the second scene when the doors open, and the “Robot” sprite greets the user.
write_hello  <pre data-bbox="204 920 633 1427"> + write_hello + set pen color to black write S size 40 wait 0.09 secs write L size 40 wait 0.09 secs write A size 40 wait 0.09 secs write L size 40 wait 0.09 secs write Y size 40 wait 0.09 secs write O size 40 wait 0.09 secs write E size 40 wait 0.09 secs write T size 40 wait 0.09 secs write O size 40 </pre>	This block writes the word “SAY HELLO TO” in a certain position with a size of 40 and a pen color of black.
paint_interior  <pre data-bbox="204 1491 474 1828"> + paint_interior + go to x: -27 y: 41 pen up set pen color to pink pen down fill pen up </pre>	This block uses the pen to colour the background with the colour pink, in the second scene after the doors open, behind the “Robot” sprite

write_book_venture



This block writes the word “BOOK VENTURE” in a certain position with a size of 40 and a pen color of black.

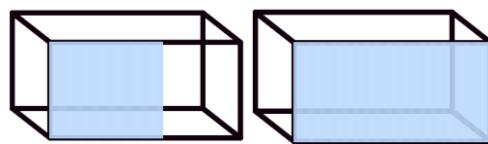
write_click_robot



This block writes the word “Please click the robot” in a certain position with a size of 20 and a pen color of blue.

fill_front_space_building

This block is used to fill the front surface of the cuboid, which represents one level of the building with colours integrated with a gliding effect.

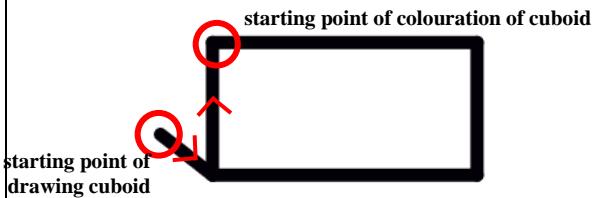


```

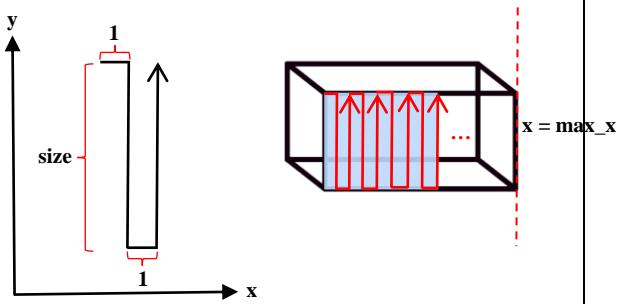
+ fill_front_space_building +
go to x: -120 y: -120 + initial_point_y_size ⏪
point in direction angle
move size / 2 steps
pen down
change y by size
repeat until x position > max_x ⏪
  change x by 1
  change y by size * -1 ⏪
  change x by 1
  change y by size
pen up

```

To start, it first goes to its starting point of drawing the cuboid, and then move to the point where it starts to colour the front surface of the cuboid.

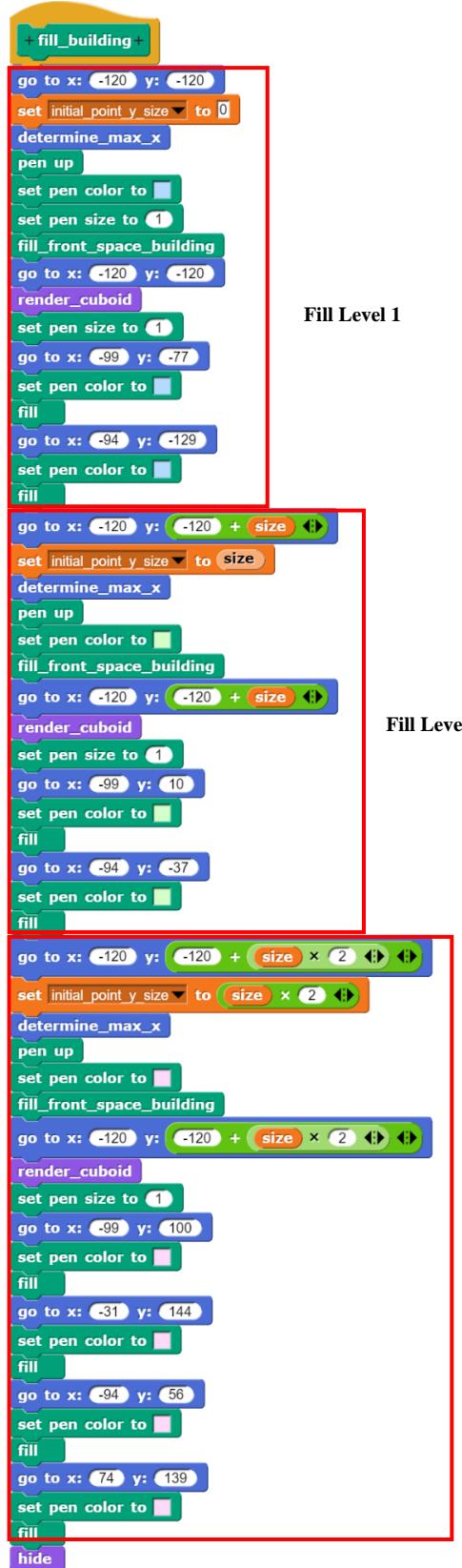


To colour the front surface, a loop is created. For each loop, it increases its x-coordinate by 1, decreases its y-coordinate by the value of the “size” variable. Then, it increases its x-coordinate by 1 again and increases its y-coordinate by 1. This loop will iterate until it exceeds the maximum x-axis value stored in the “max_x” variable defined in the [“determine_max_x” custom Motion block](#). The pen is lifted. The colouration of the front surface of the cuboid is done.



motion of sprite in each loop

fill_building

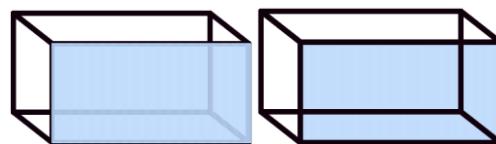


The Scratch script for "fill_building" consists of three nested loops, each labeled "Fill Level 1", "Fill Level 2", and "Fill Level 3". Each level loop contains the following sequence of blocks:

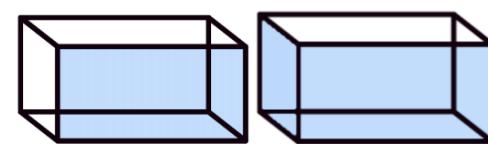
- Custom Motion block: "+ fill_building +"
- Motion block: "go to x: -120 y: -120"
- Control block: "set [initial_point_y_size v] to 0"
- Custom Motion block: "determine_max_x"
- Pen control block: "pen up"
- Pen control block: "set pen color to blue"
- Pen control block: "set pen size to 1"
- Custom Pen block: "fill_front_space_building"
- Motion block: "go to x: -120 y: -120"
- Custom Motion block: "render_cuboid"
- Pen control block: "set pen size to 1"
- Motion block: "go to x: -99 y: -77"
- Pen control block: "set pen color to blue"
- Action block: "fill"
- Motion block: "go to x: -94 y: -129"
- Pen control block: "set pen color to blue"
- Action block: "fill"
- Motion block: "go to x: (-120) y: (-120) + [size v]" (with a green arrow pointing right)
- Control block: "set [initial_point_y_size v] to [size v]" (with a green arrow pointing right)
- Custom Motion block: "determine_max_x"
- Pen control block: "pen up"
- Pen control block: "set pen color to green"
- Custom Pen block: "fill_front_space_building"
- Motion block: "go to x: (-120) y: (-120) + [size v]" (with a green arrow pointing right)
- Custom Motion block: "render_cuboid"
- Pen control block: "set pen size to 1"
- Motion block: "go to x: -99 y: 10"
- Pen control block: "set pen color to green"
- Action block: "fill"
- Motion block: "go to x: -94 y: -37"
- Pen control block: "set pen color to green"
- Action block: "fill"
- Motion block: "go to x: (-120) y: (-120) + [size v] × 2" (with a green arrow pointing right)
- Control block: "set [initial_point_y_size v] to [size v] × 2" (with a green arrow pointing right)
- Custom Motion block: "determine_max_x"
- Pen control block: "pen up"
- Pen control block: "set pen color to blue"
- Custom Pen block: "fill_front_space_building"
- Motion block: "go to x: (-120) y: (-120) + [size v] × 2" (with a green arrow pointing right)
- Custom Motion block: "render_cuboid"
- Pen control block: "set pen size to 1"
- Motion block: "go to x: -99 y: 100"
- Pen control block: "set pen color to blue"
- Action block: "fill"
- Motion block: "go to x: -31 y: 144"
- Pen control block: "set pen color to blue"
- Action block: "fill"
- Motion block: "go to x: -94 y: 56"
- Pen control block: "set pen color to blue"
- Action block: "fill"
- Motion block: "go to x: 74 y: 139"
- Pen control block: "set pen color to blue"
- Action block: "fill"
- Action block: "hide"

This block is used to highlight and colour the three cuboids of the building. In this block, three custom blocks are used, including "["determine_max_x"](#)", "["fill_front_space_building"](#)", and "["render_cuboid"](#)".

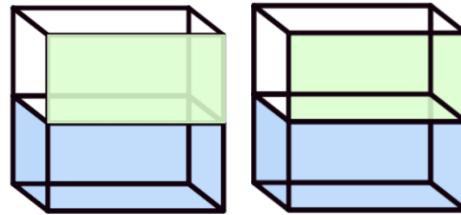
To initiate, it first goes to the starting point of drawing the cuboid and set the "["initial_point_y_size"](#)" to 0, indicating that it is at the starting point of level 1's cuboid. The maximum x-axis is determined through the "["determine_max_x"](#) custom Motion block". Pen is lifted and set to pen colour of blue and size of 1. The pen is placed down and the front surface of the level 1's cuboid is highlighted with the colour blue via the "["fill_front_space_building"](#) custom Pen block". After that, it returns to the starting point to redraw the outline of the cuboid to cover the colouration made at the outline.



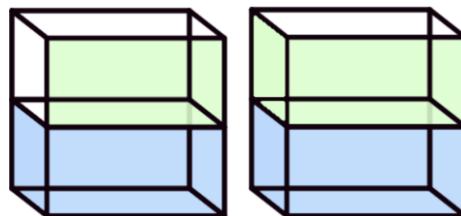
Lastly, pen size is set to 1 and it will move to the relevant coordinates to fill the empty surfaces of the level 1's cuboid with the same colour blue.



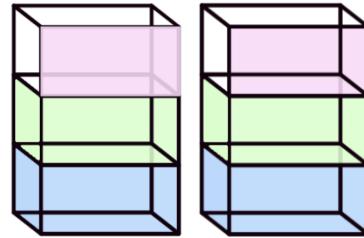
For the second cuboid of level 2, it does exactly what it is done for the first cuboid of level 1, just that the starting point is different, thus the “[initial_point_y_size](#)” variable is set to the value of the “[size](#)” variable. The maximum x-axis is determined through the “[determine_max_x](#)” custom Motion block. Pen is lifted and set to pen colour of  and size of 1. The pen is placed down and the front surface of the level 1’s cuboid is highlighted with the colour  via the “[fill_front_space_building](#)” custom Pen block. After that, it returns to the starting point to redraw the outline of the cuboid to cover the colouration made at the outline.



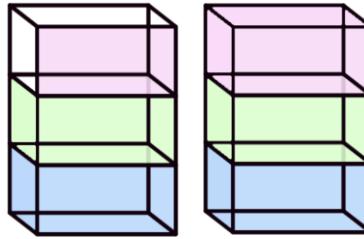
Lastly, pen size is set to 1 and it will move to the relevant coordinates to fill the empty surfaces of the level 2’s cuboid with the same colour .



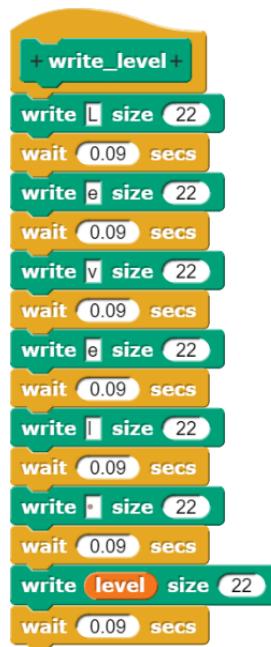
For the third and final cuboid of level 3, it does exactly what it is done for the first cuboid of level 1 as well, just that the starting point is different, thus the “[initial_point_y_size](#)” variable is set to twice the value of the “[size](#)” variable. The maximum x-axis is determined through the “[determine_max_x](#)” custom [Motion block](#). Pen is lifted and set to pen colour of  and size of 1. The pen is placed down and the front surface of the level 1’s cuboid is highlighted with the colour  via the “[fill_front_space_building](#)” custom [Pen block](#). After that, it returns to the starting point to redraw the outline of the cuboid to cover the colouration made at the outline.



Lastly, pen size is set to 1 and it will move to the relevant coordinates to fill the empty surfaces of the level 3’s cuboid with the same .

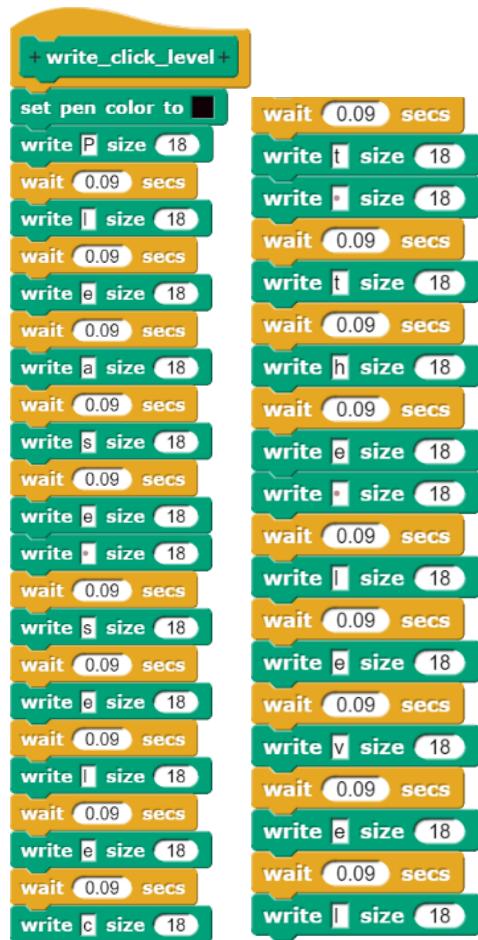


write_level



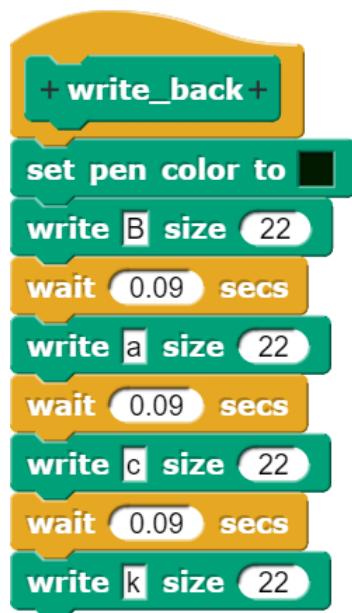
This block writes the word “Level” in a certain position with a size of 22.

write_click_level



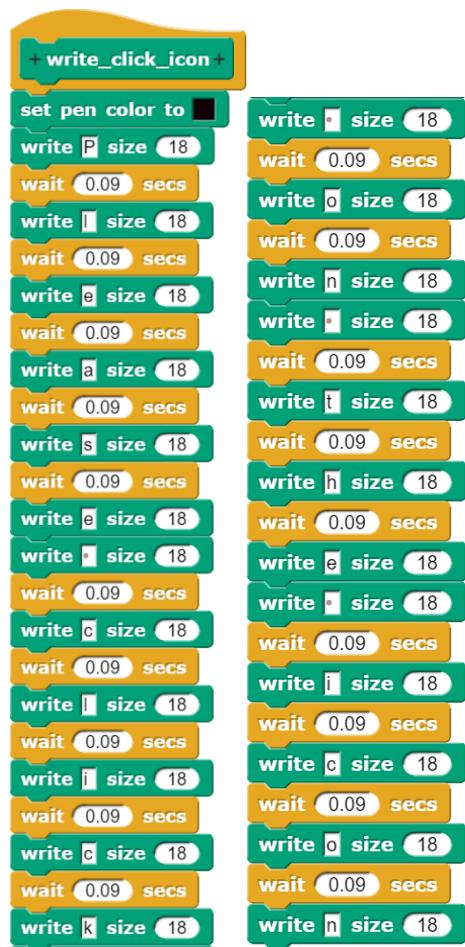
This block writes the word “Please select the level” in a certain position with a size of 18 and a pen color of black.

write_back



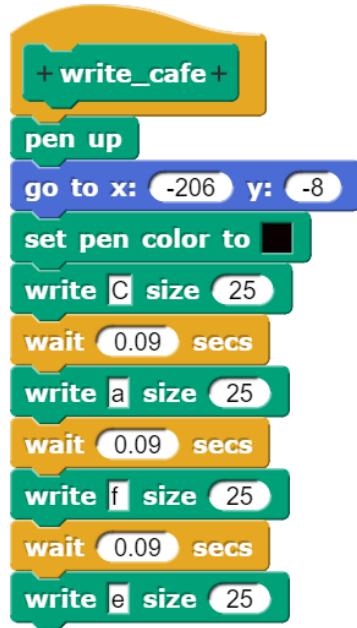
This block writes the word “Back” in a certain position with a size of 22 and a pen color of black.

write_click_icon



This block writes the word “Please click on the icon” in a certain position with a size of 18 and a pen color of black.

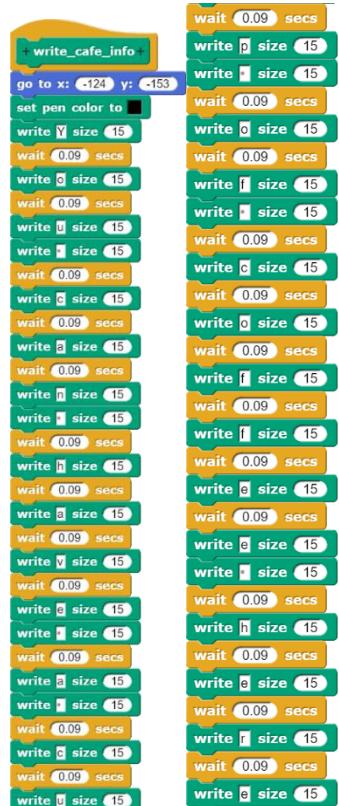
write_cafe



```
+ write_cafe +
pen up
go to x: -206 y: -8
set pen color to black
write C size 25
wait 0.09 secs
write a size 25
wait 0.09 secs
write f size 25
wait 0.09 secs
write e size 25
```

This block writes the word “Cafe” in a certain position with a size of 25 and a pen color of black.

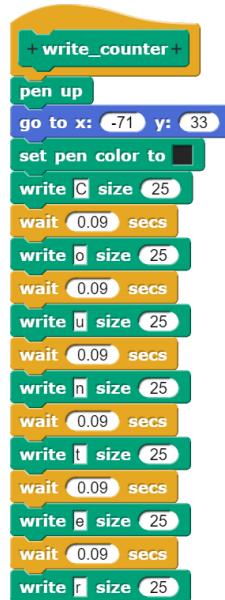
write_cafe_info



```
write_cafe_info
go to x: -124 y: -153
set pen color to black
write Y size 15
wait 0.09 secs
write o size 15
wait 0.09 secs
write u size 15
wait 0.09 secs
write n size 15
wait 0.09 secs
write v size 15
wait 0.09 secs
write e size 15
wait 0.09 secs
write a size 15
wait 0.09 secs
write i size 15
wait 0.09 secs
write c size 15
wait 0.09 secs
write o size 15
wait 0.09 secs
write h size 15
wait 0.09 secs
write e size 15
wait 0.09 secs
write r size 15
wait 0.09 secs
write e size 15
```

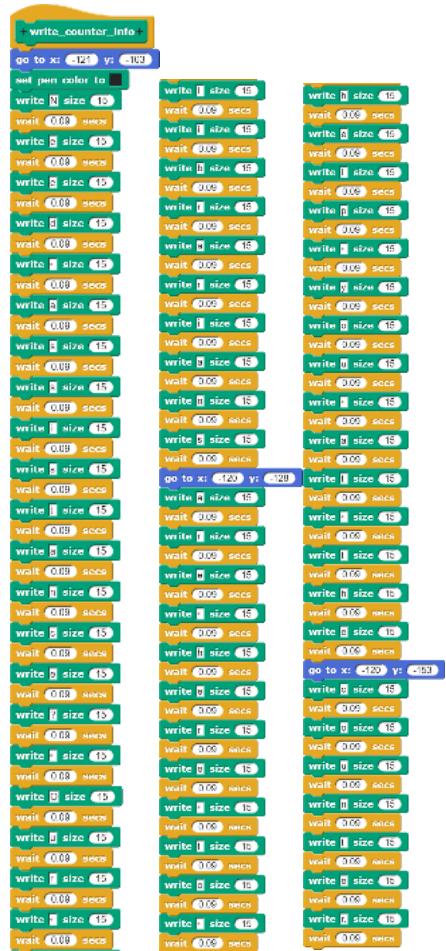
This block writes the word “You can have a cup of coffee here” in a certain position with a size of 15 and a pen color of black.

write_counter



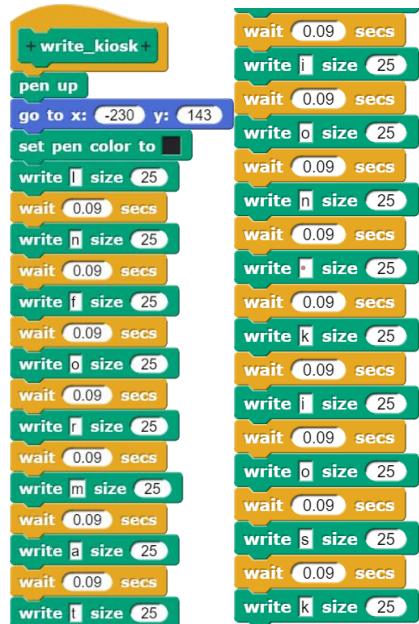
This block writes the word “Counter” in a certain position with a size of 25 and a pen color of **black**.

write_counter_info



This block writes the word “Need assistance? Our librarians are here to help you at the counter.” in a certain position with a size of 15 and a pen color of **black**.

write_kiosk



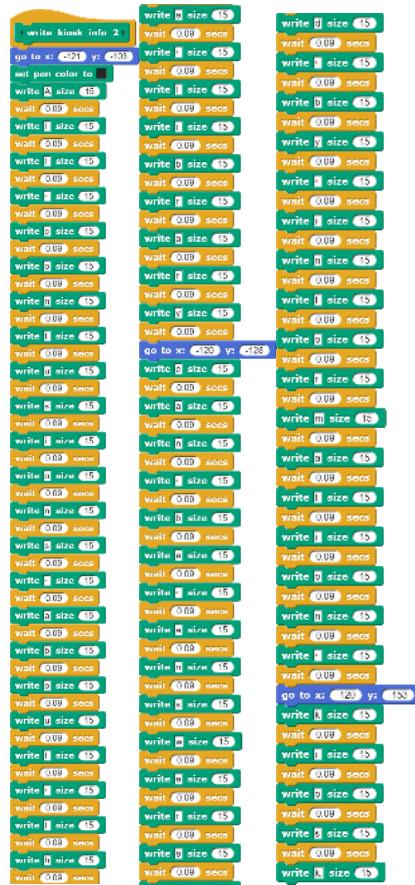
This block writes the word “Information kiosk” in a certain position with a size of 25 and a pen color of black.

write_kiosk_info



This block writes the word “Need help? Information kiosk is here to guide you.” in a certain position with a size of 15 and a pen color of black.

write_kiosk_info_2



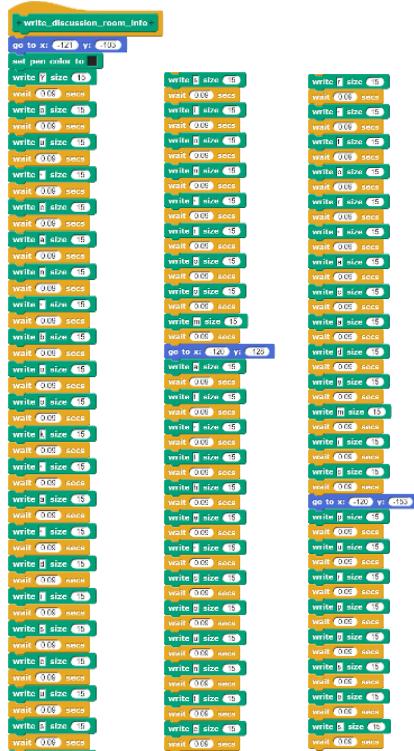
This block writes the word “All confusions about the library can be answered by information kiosk.” in a certain position with a size of 15 and a pen color of **black**.

write_discussion_room



This block writes the word “Discussion room” in a certain position with a size of 25 and a pen color of **black**.

write_discussion_room_info



This block writes the word “You can book a discussion room at the counter for academic purposes.” in a certain position with a size of 15 and a pen color of black.

write_discussion_room_info_2



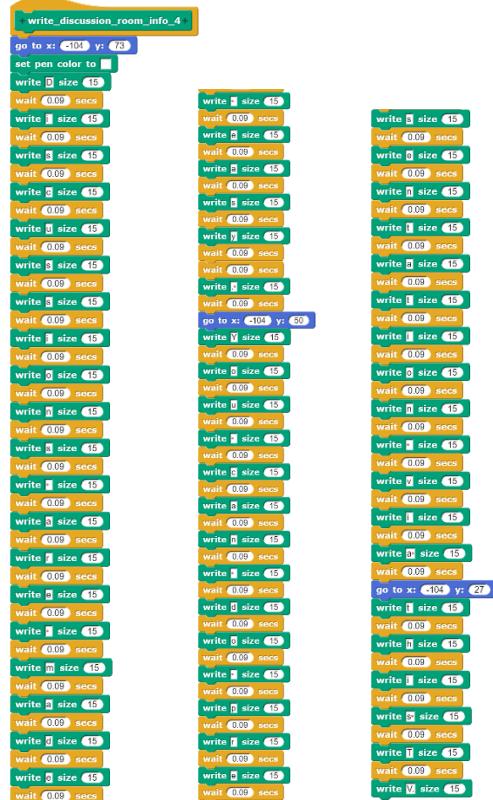
This block writes the word “For more details, please visit the counter.” in a certain position with a size of 15 and a pen color of black.

write_discussion_room_info_3



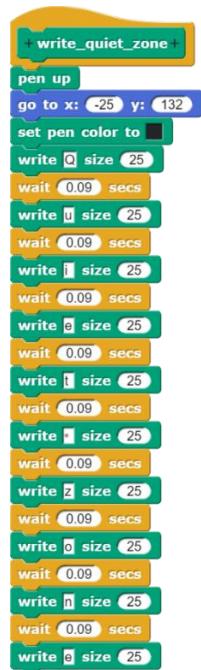
This block writes the word “Click on the icon to enter the discussion room.” in a certain position with a size of 15 and a pen color of black.

write_discussion_room_info_4



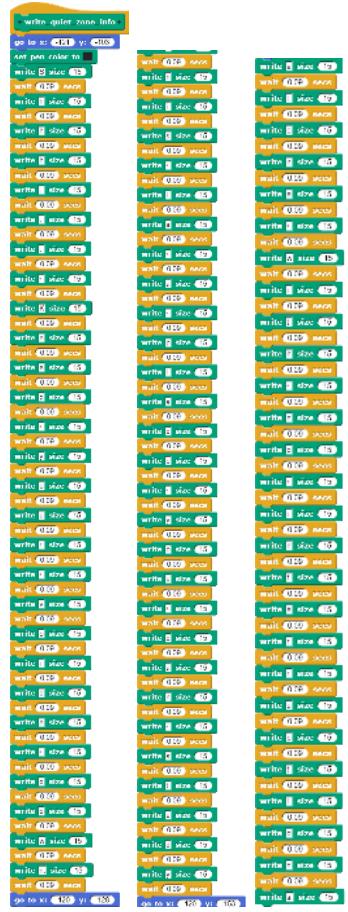
This block writes the word “Discussions are made easy. You can do presentation via this TV.” in a certain position with a size of 15 and a pen color of black.

write_quiet_zone



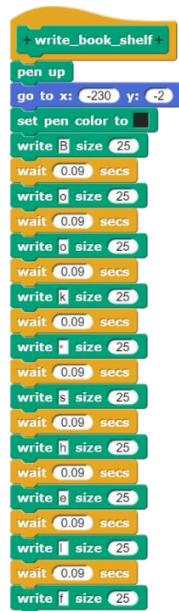
This block writes the word “Quiet zone” in a certain position with a size of 25 and a pen color of black.

write_quiet_zone_info



This block writes the word “Shhh... Keep your voice down. This is a place you can study alone with no interruptions.” in a certain position with a size of 15 and a pen color of black.

write_book_shelf



This block writes the word “Book shelf” in a certain position with a size of 25 and a pen color of ■.

write_book_shelf_info



This block writes the word “You can find some books here; there are also some books that are relevant to computing.” in a certain position with a size of 15 and a pen color of ■.

write_computer



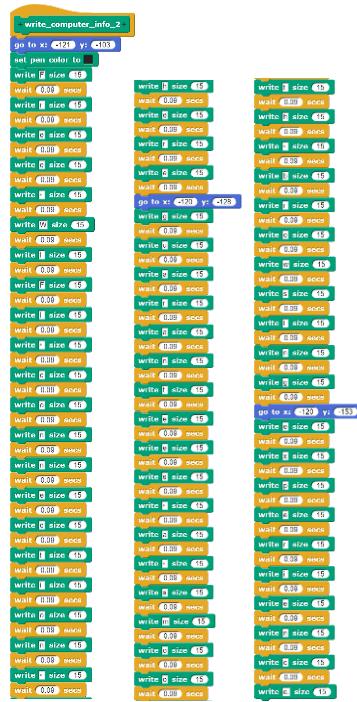
This block writes the word “Computer zone” in a certain position with a size of 25 and a pen color of black.

write_computer_info



This block writes the word “Computers are available for academic usage.” in a certain position with a size of 15 and a pen color of black.

write_computer_info_2



This block writes the word “Free WiFi connection here guarantees a smooth browsing experience.” in a certain position with a size of 15 and a pen color of black.

write_computer_info_3



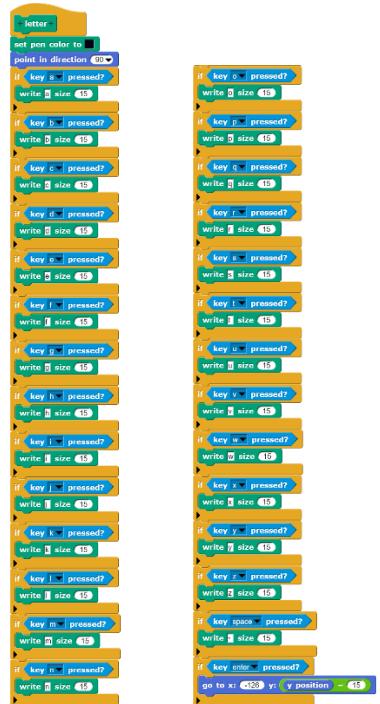
This block writes the word “An ideal studying area for everyone.” in a certain position with a size of 15 and a pen color of black.

write_computer_info_4



This block writes the word “Click on the computer icon for a brief overview.” in a certain position with a size of 15 and a pen color of black.

letter



When this block receives broadcast “Computer notepad” from “Notepad” sprite, it will point in a direction of 90 degrees and write any letter on “Computer 2 background” using “Write_notepad” sprite in a certain position with a size of 15 and a pen color of black.

review_question_1

This block writes the word “How’s the new library “BOOK VENTURE”? in a certain

```

+review_question_1+
go to x: -182 y: 25
set pen color to black
write H size 25
wait 0.09 secs
write O size 25
wait 0.09 secs
write W size 25
wait 0.09 secs
write A size 25
wait 0.09 secs
write I size 25
wait 0.09 secs
write E size 25
wait 0.09 secs
write B size 25
wait 0.09 secs
write R size 25
wait 0.09 secs
write S size 25
wait 0.09 secs
write T size 25
wait 0.09 secs
write U size 25
wait 0.09 secs
write D size 25
wait 0.09 secs
write V size 25
wait 0.09 secs
write F size 25
wait 0.09 secs
write G size 25
wait 0.09 secs
go to x: -58 y: -14
write N size 25
wait 0.09 secs
write L size 25
wait 0.09 secs
write M size 25
wait 0.09 secs
write K size 25
wait 0.09 secs
write P size 25
wait 0.09 secs

```

review_question_2

```

+review_question_2+
go to x: -158 y: -12
set pen color to black
write H size 25
wait 0.09 secs
write O size 25
wait 0.09 secs
write W size 25
wait 0.09 secs
write A size 25
wait 0.09 secs
write I size 25
wait 0.09 secs
write E size 25
wait 0.09 secs
write B size 25
wait 0.09 secs
write R size 25
wait 0.09 secs
write S size 25
wait 0.09 secs
write T size 25
wait 0.09 secs
write U size 25
wait 0.09 secs
write D size 25
wait 0.09 secs
write V size 25
wait 0.09 secs
write F size 25
wait 0.09 secs
write G size 25
wait 0.09 secs

```

position with a size of 25 and a pen color of black.

This block writes the word “How’s the virtual tour?” in a certain position with a size of 25 and a pen color of black.

review_question_3



This block writes the word “How’s the facilities?” in a certain position with a size of 25 and a pen color of black.

write_overall_rating



This block writes the word “Overall Rating” in a certain position with a size of 25 and a pen color of black, and it also displays the average of the total of the user’s rating.

3.1.10 Variables Used Throughout the Application

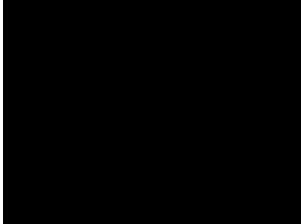
Variable	Explanation
	This variable is used to define the value of the angle. This variable is essential in blocks such as “rendering cuboid” custom Looks block , “draw door right” , and “draw door left” custom Looks block.
	This variable is used to track the frequency of clicking. When the “Robot” sprite is clicked by the users, the value of “greet” variable is set to 1, which means that the “Robot” sprite has been clicked once. Thus, when users click on the “Robot” sprite for a second times, the “Robot” sprite will not greet the users again.
	This variable is used to set the size of the cuboid for each level of the library.
	This variable is used to assign the y-coordinate of the initial or starting point where the “render cuboid” custom Looks block starts to draw the cuboid.
	This variable is used to store the value of the maximum x-axis determined through the “determine max x” custom Motion block . This variable is crucial in the “fill_front_space_building” and “fill building” custom Pen block .
	This variable is used to measure the level of library.
	This variable is used to track the coordinate of x that is to be used in the “draw info area” custom Looks block .
	This variable is used to track the coordinate of y that is to be used in the “draw info area” custom Looks block .
	This variable is used to set the height of information box based on how much information is needed to be delivered in a particular area of the library.
	This variable is used to track the frequency of effective clicking. When the “Cafe_icon” sprite is clicked by the users, the value of

	<p>“cafe” variable is set to 1, which means that the icon has been clicked once. If the “Back” sprite is clicked, the value of “cafe” variable is set to 0, which means that the icon is considered as it has been unclicked.</p>
	<p>This variable is used to track the frequency of effective clicking. When the “Counter_icon” sprite is clicked by the users, the value of “counter” variable is set to 1, which means that the icon has been clicked once. If the “Back” sprite is clicked, the value of “counter” variable is set to 0, which means that the icon is considered as it has been unclicked.</p>
	<p>This variable is used to track the frequency of effective clicking. When the “Information_kiosk_icon” sprite is clicked by the users, the value of “kiosk” variable is set to 1, which means that the icon has been clicked once. If the “Back” sprite is clicked, the value of “kiosk” variable is set to 0, which means that the icon is considered as it has been unclicked.</p>
	<p>This variable is used to track the frequency of effective clicking. When the “Discussion_room_icon” sprite is clicked by the users, the value of “discussion_room_click” variable is set to 1, which means that the icon has been clicked once. If the “Discussion_room_icon” sprite is clicked by the users again, the value of “discussion_room_click” variable is set to 2, which means that the icon has been clicked twice. If the “Back” sprite is clicked, the value of “discussion_room_click” variable is set to 1, which means that the icon is considered as it has been unclicked once. If the “Back” sprite is clicked again, the value of “discussion_room_click” variable is set to 0, which means that the icon is considered as it has been unclicked twice.</p>
	<p>This variable is used to track the frequency of effective clicking. When the “Quiet_zone_icon” sprite is clicked by the users, the value of “quiet_zone” variable is set to 1, which means that the</p>

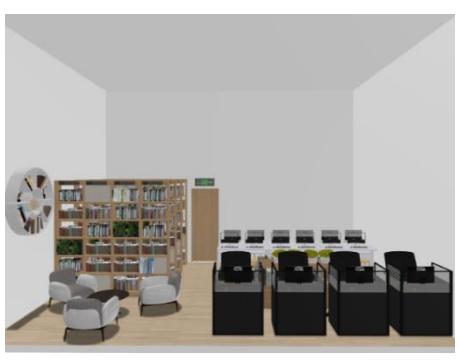
	icon has been clicked once. If the “Back” sprite is clicked, the value of “quiet_zone” variable is set to 0, which means that the icon is considered as it has been unclicked.
book_shelf	This variable is used to track the frequency of effective clicking. When the “Book_shelf_icon” sprite is clicked by the users, the value of “ book_shelf ” variable is set to 1, which means that the icon has been clicked once. If the “Back” sprite is clicked, the value of “ book_shelf ” variable is set to 0, which means that the icon is considered as it has been unclicked.
computer_click	This variable is used to track the frequency of effective clicking. When the “Computer_icon” sprite is clicked by the users, the value of “ computer_click ” variable is set to 1, which means that the icon has been clicked once. If the “Computer_icon” sprite is clicked by the users again, the value of “ computer_click ” variable is set to 2, which means that the icon has been clicked twice. If the “Back” sprite is clicked, the value of “ computer_click ” variable is set to 1, which means that the icon is considered as it has been unclicked once. If the “Back” sprite is clicked again, the value of “ computer_click ” variable is set to 0, which means that the icon is considered as it has been unclicked twice.
computer_notepad	This variable is used to determine the status of the notepad. When the “Notepad” sprite is clicked by the users, the value of “ computer_notepad ” variable is set to 1, which means that the sprite has been clicked once and notepad is in use. If the “Close_tab” sprite is clicked again, the value of “ computer_notepad ” variable is set to 0, which means that the icon is considered as it has been unclicked and the notepad is not in use.
rating_list	This variable is created to make a list that stores the rating given by the users for all the three review questions asked.
review	This variable is used to keep track on how many review questions has been displayed.

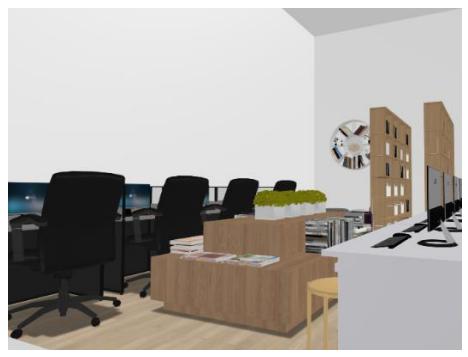
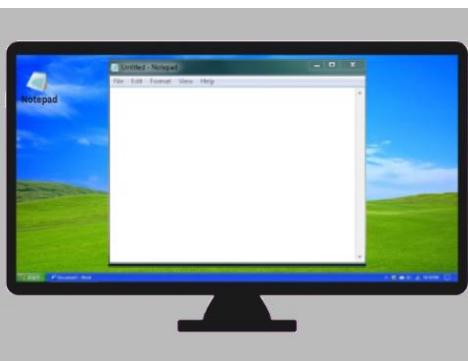
click_star_1	This variable is used to keep track on whether “star_1” sprite is clicked and highlighted. If it is clicked, the value of this variable will be set to 1. Else, the value of this variable equals to 0.
click_star_2	This variable is used to keep track on whether “star_2” sprite is clicked and highlighted. If it is clicked, the value of this variable will be set to 1. Else, the value of this variable equals to 0.
click_star_3	This variable is used to keep track on whether “star_3” sprite is clicked and highlighted. If it is clicked, the value of this variable will be set to 1. Else, the value of this variable equals to 0.
click_star_4	This variable is used to keep track on whether “star_4” sprite is clicked and highlighted. If it is clicked, the value of this variable will be set to 1. Else, the value of this variable equals to 0.
click_star_5	This variable is used to keep track on whether “star_5” sprite is clicked and highlighted. If it is clicked, the value of this variable will be set to 1. Else, the value of this variable equals to 0.
overall_rating	This variable is used to store the average rating given by the users that is calculated by the “Submit” sprite.

3.2 Backdrops functionality used in the application

Backdrops	Functionality
Black 	This backdrop serves as the background for both “Opening” and “Ending” sprites. By entering this backdrop, background music plays for both opening and ending, thereby enhancing users’ experience.
Grey_black_border 	This backdrop serves as the background for 3D model of the building that is drawn by “Library_intro” sprite and also serves as the background for the users’ review that is written by “Review” sprite. By entering this backdrop, background music plays for both the 3D model of the building and the review, thereby enhancing users’ experience.
Grey 	This backdrop serves as the background for the door that is drawn by “Door” sprite. By entering this backdrop, a background music, thereby enhancing users’ experience.
Level 1 of the Library in the Application	
Level_1 	This backdrop is used to display Level 1 of the library to users, which provides an overall view of the environment to users.

<p>Cafe</p> 	<p>This backdrop is used to display a magnified and zoomed in picture of the cafe in Level 1 of the library to users when they interact with the “Cafe_icon” sprite. By entering this backdrop, a background music with a cafe theme plays, thereby enhancing users’ experience.</p>
<p>Kiosk</p> 	<p>This backdrop is used to display a magnified and zoomed in picture of the information kiosk in Level 1 of the library to users when they interact with the “Information_kiosk_icon” sprite. By entering this backdrop, a background music plays, thereby enhancing users’ experience.</p>
<p>Counter</p> 	<p>This backdrop is used to display a magnified and zoomed in picture of the counter in Level 1 of the library to users when they interact with the “Counter_icon” sprite. By entering this backdrop, a background music plays, thereby enhancing users’ experience.</p>
<p>Level 2 of the Library in the Application</p>	
<p>Level_2</p> 	<p>This backdrop is used to display Level 2 of the library to users, which provides an overall view of the environment to users.</p>

<p>Discussion_room</p> 	<p>This backdrop is used to display a magnified and zoomed in picture of the hallway facing the discussion room in Level 2 of the library to users when they interact with the “Discussion_room_icon” sprite. By entering this backdrop, a background music plays, thereby enhancing users’ experience.</p>
<p>Discussion_room_2</p> 	<p>This backdrop is used to display the interior of the discussion room in Level 2 of the library to users when they interact with the “Discussion_room_icon” sprite.</p>
<p>Quite_zone</p> 	<p>This backdrop is used to display a magnified and zoomed in picture of the quiet zone in Level 2 of the library to users when they interact with the “Quite_zone_icon” sprite. By entering this backdrop, a sound effect “shhh” plays, which warn users to be quiet when they are in the quiet zone, thereby enhancing users’ experience.</p>
<p>Level 3 of the Library in the Application</p>	
<p>Level_3</p> 	<p>This backdrop is used to display Level 3 of the library to users, which provides an overall view of the environment to users.</p>

<p>Book_shelf</p> 	<p>This backdrop is used to display a magnified and zoomed in picture of the bookshelf in Level 3 of the library to users when they interact with the “Book_shelf_icon” sprite. By entering this backdrop, a background music plays, thereby enhancing users’ experience.</p>
<p>Computer_scene</p> 	<p>This backdrop is used to display a magnified and zoomed in picture of the computer zone in Level 3 of the library to users when they interact with the “Computer_icon” sprite. By entering this backdrop, a background music plays, thereby enhancing users’ experience.</p>
<p>Computer</p> 	<p>This backdrop is displayed when users interact with the “Computer_icon” sprite, which shows users the computer home page with the “Notepad” sprite. By entering this backdrop, an effect of computer startup sound plays, thereby enhancing users’ experience.</p>
<p>Computer_2</p> 	<p>This backdrop is displayed when users interact with the “Notepad” sprite, which allows users to have a visually immersive experience as they open the notepad tab.</p>

4.0 Stage 4: Implementation

The implementation stage describes the way how the team put the codes and algorithms together which are required to create a functional Virtual Tour application for the new library of under School of Computing. The coding process on Snap! has include the stacking of multiple blocks on each other to create specific commands or orders. Therefore, this stage plays an important role to ensure that the application operates smoothly without any bugs or errors.

4.1 Application user manual

A user manual is truly necessary for guiding the users on how to use this application to gain a decent virtual tour experience of the new library under The School of Computing. The team has given comprehensive instruction in every scenario to allow users to have a better understanding about the functionality of the application. Furthermore, the user can just follow the instructions that have been given for a better user experience. An example is provided in Figure 4.1.1 below. This implies that the requirement of users having the basic knowledge of using Snap! is not mandatory. In terms of initiating the application, user can start the application by simply clicking the flag button  on Snap! and the “Robot” sprite. In terms of conducting the virtual tour, user can do so by following the instructions given in certain scenarios which will guide the user.

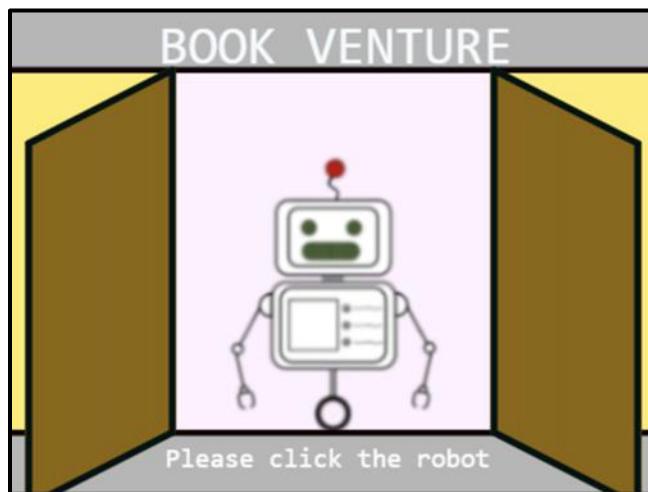


Figure 4.1.1 The “Robot” sprite waiting to be clicked.

In the beginning of the Virtual Tour application, the user would be asked to enter their name. This will allow the “Robot” sprite’s greeting to the user to be personalized. After that, the user would

be given an instruction to select the level that they are curious about and wanted to have a look at. The example is provided in Figure 4.1.2 below.

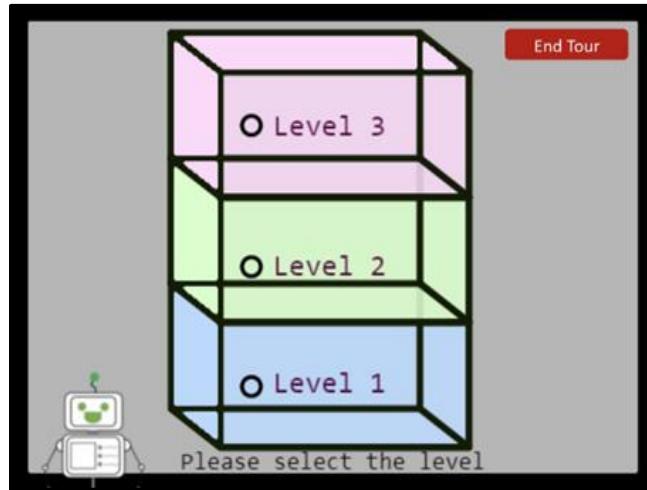


Figure 4.1.2 The scenario of level selection.

After the selection has been made by the user and clicked the level sprite, the user will be directed to the specific level. In every level, there would be some icon that represent the rooms or the facilities on the level. When the icon is clicked by the user, the “Robot” sprite would give information and explain the purpose of the room or facilities. An example has been given in Figure 4.1.3 below.



Figure 4.1.3 Level 1 which has icon.

When all the levels have been visited by the user, the user can click on the end tour button to end the Virtual Tour. Besides that, a star rating session will be provided to the user for evaluation of the performance of the application of Virtual Tour application. The example has been provided in Figure 4.1.4 below.



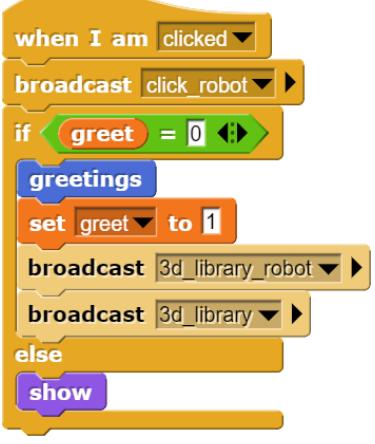
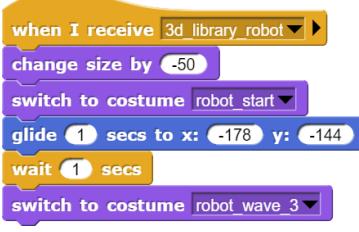
Figure 4.1.4 One of the Question at Rating Session.

4.2 Screenshots of sample input/output and explanation

As the application is programmed on Snap! which is a visual programming language, the input and output refer to the flow of data between the application and user.

4.2.1 Input in the Application

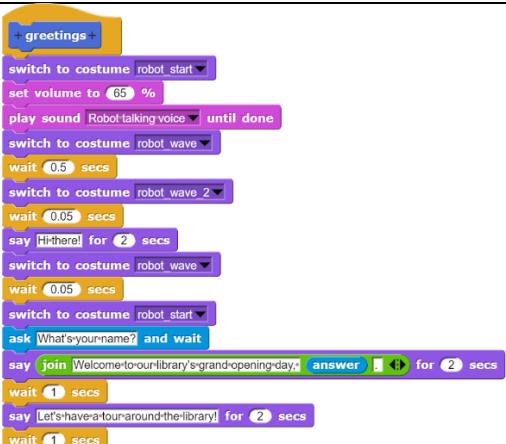
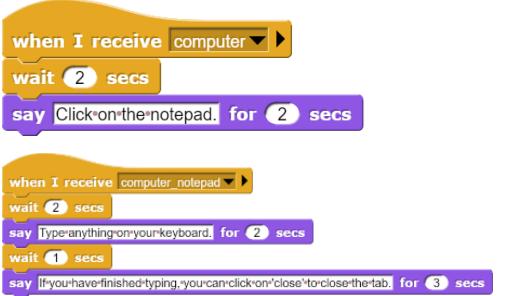
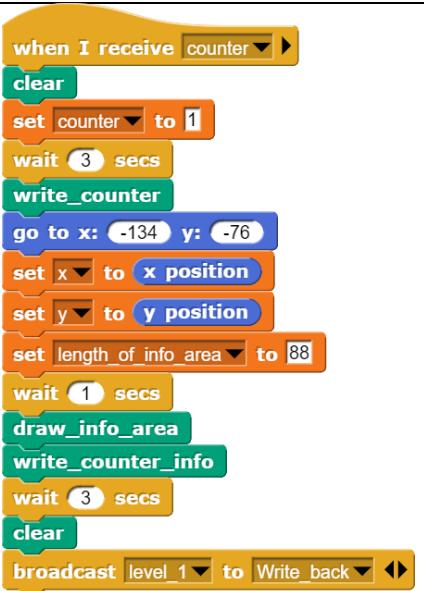
Table 4.2.1 Input explanation in the Application

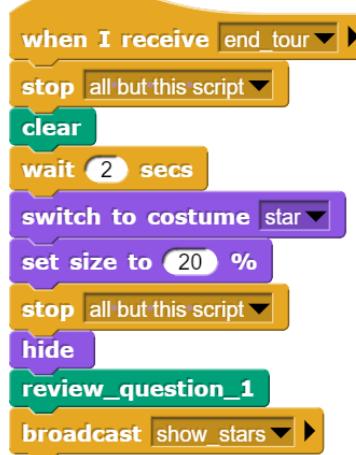
Input	Screenshot	Explanation
When green flag icon is clicked block.		The flag icon means that the Virtual Tour application had been started. In this case, the flag icon brings the user to the greeting session.
		The sprite would not appear when the green flag icon is clicked.
When I am clicked block.		The sprite will continue the next part of the Virtual Tour application once the user clicked the sprite.
When I receive block.		The function of this block is to receive the message broadcasted to display the specific scenario. Every scenario has its own specific broadcast messages.

When alphabet key pressed block	<pre> when [letter] key pressed set pen color to [black v] point in direction [90 ° v] if [key a v] pressed? write [a v] size [15 v] end if [key b v] pressed? write [b v] size [15 v] end if [key c v] pressed? write [c v] size [15 v] end if [key d v] pressed? write [d v] size [15 v] end if [key e v] pressed? write [e v] size [15 v] end </pre>	<p>The function of this block is to allow users to type some word on their keyboard by pressing alphabet keys when they are in the stage where the Notepad tab is shown.</p>
Ask and wait block	<pre> greetings switch to costume [robot_start v] set volume to [65 %] play sound [Robot talking voice v] until done switch to costume [robot_wave v] wait [0.5 sec] switch to costume [robot_wave_2 v] wait [0.05 sec] say [Hi there!] for [2] secs switch to costume [robot_start v] ask [What's your name?] and wait say [join Welcome to our library's grand opening day!] answer [.] for [2] secs wait [1 sec] say [Let's have a tour around the library!] for [2] secs wait [1 sec] </pre>	<p>The user would have to enter their name that would be used for the purpose of greeting to them in the starting of the Virtual Tour.</p>

4.2.2 Output in the Application

Table 4.2.2 Output explanation in the Virtual Tour

Output	Screenshot	Explanation
Say block	 <pre> when green flag clicked say [greetings+] switch to costume robot_start set volume to 65 % play sound Robot-talking voice until done switch to costume robot_wave wait 0.5 secs switch to costume robot_wave_2 wait 0.05 secs say [Hi there!] for 2 secs switch to costume robot_wave wait 0.05 secs switch to costume robot_start ask [What's your name?] and wait say [join [Welcome-to-our-library/sgrand-opening-day.], [answer]] for 2 secs wait 1 secs say [Let's have a tour around the library!] for 2 secs wait 1 secs </pre>	Greeting dialogue When the user clicks on the “Robot” sprite, the sprite would welcome the user to the Virtual Tour.
	 <pre> when I receive [computer v] wait 2 secs say [Click on the notepad.] for 2 secs when I receive [computer notepad v] wait 2 secs say [Type anything on your keyboard.] for 2 secs wait 1 secs say [If you have finished typing, you can click on "close" to close the tab.] for 3 secs </pre>	Notepad dialogue When the user clicks on the “Computer_icon” sprite, the “Robot” sprite would ask the user to click on the “Notepad” sprite and can type on the notepad.
Broadcast block	 <pre> when I receive [counter v] clear set [counter v] to 1 wait 3 secs write_counter go to x: [-134] y: [-76] set [x v] to [x position] set [y v] to [y position] set [length_of_info_area v] to 88 wait 1 secs draw_info_area write_counter_info wait 3 secs clear broadcast [level_1 v] to [Write_back v] </pre>	Broadcast room or facilities When the user chooses the rooms or facilities icon, the stage would broadcast to the specific stage or scenario.



A Scratch script consisting of the following blocks:

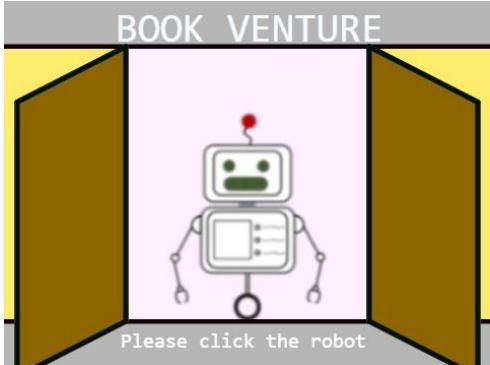
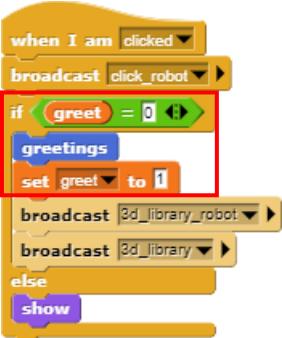
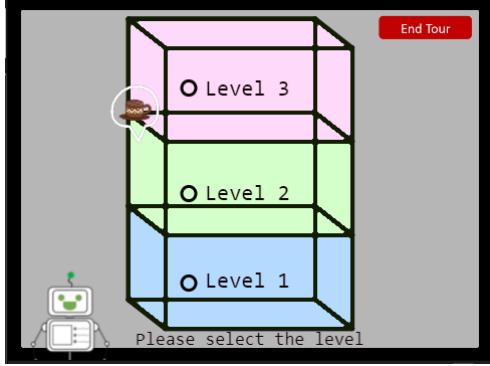
- when I receive [end_tour v]
- stop [all but this script v]
- clear
- wait (2 secs)
- switch to costume [star v]
- set size to (20 %)
- stop [all but this script v]
- hide
- review_question_1
- broadcast [show_stars v]

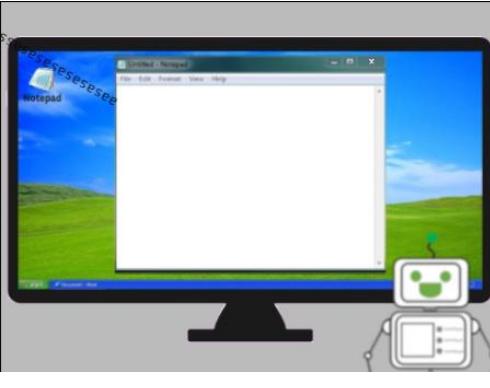
Broadcast review

The user would take a rating session to review the performance of the application on Virtual Tour by clicking on the end tour sprite.

4.3 Testing results

Initial test on 12/12/2023

Issue	Solution
<p>When click on the “Robot” sprite, “Robot” sprite will greet to the users. After that, when users accidentally clicked on the “robot” sprite, it will greet again, and it will return back to the previous script which causes a crash to the application.</p> 	<p>Set the variable “<u>greet</u>” to 1, if the value of variable “<u>greet</u>” is 1, the greetings from the “Robot” sprite to the users will occur. If the value of variable “<u>greet</u>” returns to 0, the “Robot” sprite will not greet the users for the second time.</p> 
<p>When click back, the “Cafe_icon” sprite is still appearing in the 3d building scene.</p> 	<p>Hide the “Cafe_icon” sprite when the backdrop is changed from “Level_1” to “Grey_black_border” when receive “3d_library_back” broadcasted by “Back” sprite.</p> 
<p>When clicking the “Button_level_2” sprite, it brings users to the “Level_3” backdrop but not the “Level_2” backdrop.</p>	<p>When the “Button_level_2” sprite is clicked, it will broadcast to the “level_2” message and not the “level_3” message. Thus, when the stage receives “level_2” broadcasted by</p>

 <p>Please click on the icon</p>	<p>“Button_level_2” sprite, it will change into “Level_2” backdrop.</p> <pre>when I am clicked play sound Click2 stop all but this script clear broadcast level_2 hide</pre>
<p>When typing on the keyboard in the “Computer” stage using the “Write_notepad” sprite, the costume changes its direction and writes away from the Notepad tab.</p> 	<p>Adjust the direction of the “Write_notepad” sprite to a constant direction, which is point in the direction of 90 degrees.</p> <pre>letter set pen color to black point in direction (90) if key a pressed? write a size 15 if key b pressed? write b size 15 if key c pressed? write c size 15 if key d pressed?</pre>

Final test on 28/12/2023

Issue	Solution
<p>When click back, the “Arrow” sprite is still there when its script is force stop.</p> 	<p>Hide the “Arrow” sprite when the backdrop is changed from “Kiosk” to “Level_1” when receive “level_1” broadcasted by “Button_level_1” sprite.</p> <pre>when I receive level_1 hide</pre>

When any key is pressed at any point of the application, the "write_notepad" sprite will still be able to function and write on the screen.



The value of variable "computer notepad" is set to 0 when the backdrop is changed from "Computer_2" to "Computer" so that the sprite "Write_notepad" can just only function and write when the value of variable "computer notepad" is 1.

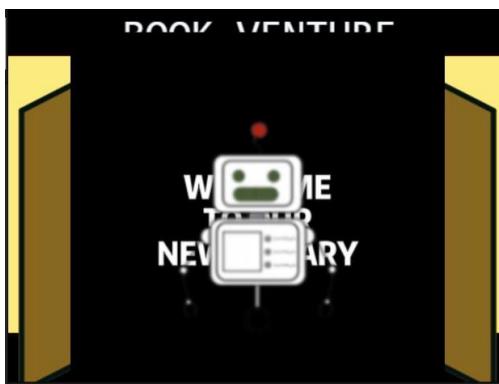
```

when I receive [computer notepad v]
go to x: -126 y: 85
set [computer notepad v] to [1]

when any key is pressed
if [computer notepad v] = [1]
letter
if [x position v] > [121]
go to x: -126 y: [y position v] - [15]
if [y position v] < [-80]
go to x: -126 y: 85
clear

```

When the script in the current sprite or other sprite is running, it continues to run even when the flag is clicked.



"stop all but this script" command in the script in "Opening" sprite.

```

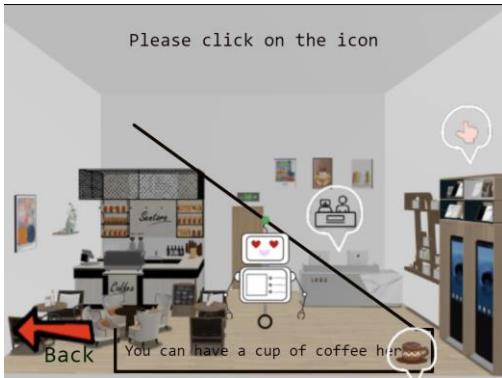
when green flag is clicked
stop [all but this script]
hide
go to x: 0 y: -250
set [ghost effect v] to [0]
broadcast [opening]

```

When the users abruptly click the "Back" sprite even though the scene is not finish displaying, the current script that is still running will continue to run, which will disrupt the presentation of the previous scene when users are redirected back to the previous scene.

The sprite needed to use the command of "stop all but this script".

The script that should be running at that scene will continue to run on the previous scene.



```

when I am clicked
play sound Click2 until done
if cafe = 1
stop [all but this script]
clear
broadcast level_1
set cafe to 0

```

When we stop the script halfway, the pen in the [“draw_info_area” custom Motion block](#) remains down, so the lines will appear.



“pen up” block command must be put at the beginning of the first block command that is using pen before moving to its position.

```

when I receive cafe
clear
set cafe to 1
wait 3 secs
write_cafe
go to x: -133 y: -128
set x to x position
set y to y position
set height of info area to 40
wait 1 secs
draw_info_area
write_cafe_info
wait 3 secs
clear
broadcast level 1 to Write back

```

“write_cafe” is the first command in the script to use pen, so “pen up” block command need to be included in its script.

```

+ write_cafe +
pen up
go to x: -206 y: -8
set pen color to black
write C size 25
wait 0.09 secs
write a size 25
wait 0.09 secs
write f size 25
wait 0.09 secs
write e size 25

```

5.0 Conclusion

In conclusion, this application was designed to introduce the new 3-tier library to both students and lecturers, and to assist them in understanding the amenities provided and the facilities that are available as well as where everything is located across the library. With the help of a robot, users can interact with the application and the robot will direct the user to where each facility or amenity is located as well as provide a short explanation of how the user may fully utilize each of the services provided.

Our group however faced some issues and difficulties with using the Snap! programming, as it does have certain limitations.

- One limitation is that the Snap! software doesn't allow a user to import a video. While we were able to import photos to use as a background or as a sprite, we were unable to import a video, even though having a video of the interior of the library would have enhanced the virtual tour experience.
- Our group also experienced issues with the sound feature in Snap!. Although the code was written correctly and should have allowed the sound and program to run accordingly, the sound feature seems to have its own technical issues. Sometimes, the sound would play correctly as it was set to, and other times it just doesn't work, even though there aren't any errors in the code.
- We also struggled to import images into Snap! as the resizing of the images in Snap! tends to create a glitch which occasionally causes the images to become blur while we are trying to enlarge or shrink the image. To avoid this problem, we have to apply the "set size" command into the script instead of using the resizing function.
- Snap! also has a limitation when creating broadcasts. After we had created a broadcast, we found that we were unable to rename the broadcast. Any editing of the name of a broadcast was extremely time-consuming as we would be required to create a new broadcast, then manually change each command to begin when the new broadcast was received.
- Another issue we faced is that the size of the image is restricted to 4:3 ratio. It is small, and this restriction causes the presentation of the application to be less enjoyable and engaging for users, as Snap cannot be displayed in full screen mode, and is restricted to 4:3 ratio.

- Additionally, Snap! cloud storage is only able to store files up to 10 MB in size. So while we were working on this application, we were only able to use cloud storage for the initial stages of the development. Once the file exceeded 10 MB, we would have to download it and send it to our group members each time we made any adjustments or updates.

Overall, we learned a lot while creating this interactive application that was made to introduce, explain and be a virtual tour for users of our new library. Seeing the fun and interactive visual effects we were able to create using Snap! as well as working and learning together was indeed a satisfying and enriching experience.

6.0 References

23,053 Audio Icon On Off Images, Stock Photos, 3D objects, & Vectors. (n.d.). Shutterstock.

Retrieved December 31, 2023, from <https://www.shutterstock.com/search/audio-icon-on-off>

34,531 Meeting Room Icon Images, Stock Photos, 3D objects, & Vectors / Shutterstock. (n.d.).

Shutterstock. <https://www.shutterstock.com/search/meeting-room-icon>

Ceebee. (2018). Glitch. <https://scratch.mit.edu/projects/205602351/editor/>

Desktop computer icon computer #AD , #AFFILIATE, #paid, #icon, #computer, #Desktop | Computer icon, Computer, Monitor. (n.d.). Pinterest. Retrieved December 31, 2023, from <https://www.pinterest.com/pin/desktop-computer-icon-computer-ad-affiliate-paid-icon-computer-desktop--713609503445425488/>

Gdainti. (2019). *Pixel art design 8 bit retro icon - opened and closed book*. Dreamstime.

<https://www.dreamstime.com/pixel-art-design-bit-retro-icon-opened-closed-book-dark-background-image157429343>

Goncharuk, T. (2022). *Download simple vector drawing in doodle style. robot. cute robot hand drawn with lines. funny illustration for kids for free*. Vecteezy.

<https://www.vecteezy.com/vector-art/10403265-simple-vector-drawing-in-doodle-style-robot-cute-robot-hand-drawn-with-lines-funny-illustration-for-kids>

Grass and Sky Wallpapers (50+ images inside). (n.d.). <https://wallpapercosmos.com/grass-and-sky-wallpapers>

How do I change default background windows color? (n.d.). Super User.

<https://superuser.com/questions/253624/how-do-i-change-default-background-windows-color>

Meeting room, conference, board line icon. Office desk,. (n.d.). DesignBundles.net.

<https://designbundles.net/cooshstore/1051594-meeting-room-conference-board-line-icon-office-des>

Scratch Tutorial: 3D Cube! (easy and fast). (n.d.). Www.youtube.com. Retrieved December 28, 2023, from <https://youtu.be/ycaPMZTq7tw?si=nHnaSj2v4X7t1Qdp>

Surur, S. (2022). *Download the People icon vector with cashier table. cashier, Payment, service. line icon style. Simple design. . .* Vecteezy. <https://www.vecteezy.com/vector-art/8025176-people-icon-vector-with-cashier-table-cashier-payment-service-line-icon-style-simple-design-illustration-editable>

Wiki, C. T. H. & C. (n.d.). *NotePad.* HTML & CSS Wiki.

<https://htmlcss.fandom.com/wiki/Notepad>