

Engineering Design - GNG 1103 [C]

Design Project User and Product Manual

Waste Management

Eh, Are You Recycling?

Professor M. Majeed

/SLASH\
Section C03 – Group 11

Team Members

Sam Barrett	300109196
Laura Chin	300105534
Hongdi Fan	300109379
Seonaid Hair	300114581
Adam Tilley	300108974

11/04/2021

University of Ottawa

Abstract

This product is a cost-effective, easy, and reliable source for users to use in their homes to improve recycling and promote good habits. This deliverable records the entire process of the creation of the app and website ranging from the instructions to build the product to how to use the product most efficiently. This document could be used to completely recreate the project or build off of the existing product.

Table of Contents

Abstract	2
List of Figures	5
List of Tables	9
List of Acronyms and Glossary	9
1. Introduction.....	10
2. Overview.....	10
2.1. App.....	11
2.2. Website.....	12
3. Getting Started	12
3.1. App.....	12
3.1.1. Set-Up Considerations	12
3.1.2. Preparing the App Scenes for Use	21
3.1.3. User Access Considerations.....	26
3.1.4. Accessing the System	28
3.1.5. System Organization & Navigation	28
3.1.6. Exiting the System	29
3.2. Website.....	29
3.2.1. Set-Up and User Access.....	29
3.2.2. System Organization & Navigation	29
4. Using the System	31
4.1. App.....	31
4.1.1. UI	31
4.1.2. Sorting.....	33
4.1.3. Scanning.....	39
4.1.4. Learning	43
4.2. Website.....	43
4.2.1. Manual Sorting.....	43
4.2.2. Maps.....	44
4.2.3. Games	46
4.2.4. Education and News	47
4.2.5. Discussion Forum	47

4.2.6.	Members	48
5.	Troubleshooting & Support	48
5.1.	Error Messages or Behaviors	48
5.2.	Maintenance	50
5.3.	Support	50
6.	Product Documentation	50
6.1.	App.....	50
6.1.1.	Bill of Materials of App.....	50
6.1.2.	Equipment List.....	51
6.1.3.	Instructions.....	51
6.1.4.	Testing & Validation.....	106
6.2.	Website.....	108
6.2.1.	Bill of Materials	108
6.2.2.	Equipment List.....	108
6.2.3.	Instructions.....	108
6.2.4.	Testing & Validation.....	114
7.	Conclusions and Recommendations for Future Work	114
7.1.	App.....	114
7.2.	Website.....	115

List of Figures

Figure 1: App Home Page.....	11
Figure 2: Website Home Page	12
Figure 3: Unity Website to download Unity Hub.....	13
Figure 4: Pop up menu to choose where to download the Unity Hub installer	13
Figure 5: Pop up when the Unity Hub Installer is opened.....	14
Figure 6: Pop up when Unity Hub is ready to begin installing.....	15
Figure 7: Opening page of Unity Hub	15
Figure 8: Install page of Unity Hub	16
Figure 9: Webpage to access older versions of Unity.....	17
Figure 10: Webpage to download the 2019.4.22 version of Unity	17
Figure 11: Pop-up that opens once the download button is pressed.....	18
Figure 12: Pop-up menu in Unity Hub to install the version of Unity	18
Figure 13: User agreement menus that appear before install can occur	19
Figure 14: GitHub Repository - Main Branch	19
Figure 15: Code Button Dropdown - Download Zip.....	20
Figure 16: UnityHub Add Button	20
Figure 17: UnityHub - Downloaded Project.....	21
Figure 18: Opening screen when the Unity project is opened	21
Figure 19: Unity Project save 2 page while opening the Build Settings	22
Figure 20: Build Settings Menu with Android support	22
Figure 21: Location of Package Manager in Unity.....	23
Figure 22: Package Manager menu while searching for the Vuforia Engine	24
Figure 23: Inspector window for ARCamera.....	24
Figure 24: Vuforia configuration menu with inputted code	25
Figure 25: Unity with file open to locate the Build and Run button.....	26
Figure 26: Bottom of Setting menu on Android phone	27
Figure 27: About phone menu	28
Figure 28: Log In Button on Home Page.....	29
Figure 29: Navigation Bar on Website	30
Figure 30: Button Animation Colour Change.....	30
Figure 31: Example of Instructions on Website	30
Figure 32: Search Bar with Dropdown Results	31
Figure 33: Original Home Screen	32
Figure 34: Original Search Screen.....	32
Figure 35: Manual sorting page	33
Figure 36. Specific item category sorting page	34
Figure 37: Plastics sub-section search page.....	35
Figure 38: Example of item search in search bar.....	36
Figure 39: Item typed into search bar and ready to be searched.....	37
Figure 40: Item location scene for searched item in database	38
Figure 41. Error message when an incorrect item is searched.....	39
Figure 42: Accessing the Scanning Function.....	40

Figure 43: Utilizing the Scanning Function.....	41
Figure 44: Exiting the Scanning Function	42
Figure 45: Learning Page.....	43
Figure 46: Manual Sorting Page	44
Figure 47: Metal Sorting Page Breakdown.....	44
Figure 48: Special Recycling Locations Map.....	45
Figure 49: Waste Bins on Campus Map	45
Figure 50: Games Page	46
Figure 51: Sort the Recycling Game.....	46
Figure 52: Education and News Page	47
Figure 53: Discussion Forum.....	47
Figure 54: Members Page When Not Logged In.....	48
Figure 55: Members Page When Logged In.....	48
Figure 56: Error message when an incorrect item is typed into the search bar	50
Figure 57. Switching Platforms	52
Figure 58. Changing Aspect Ratio.....	52
Figure 59. Inserting a Canvas	53
Figure 60. Inserting a Blank Image.....	53
Figure 61. Changing the Colour.....	54
Figure 62. Opening the Asset Store	55
Figure 63. Downloading and Importing Lean GUI	55
Figure 64. Lean GUI Assets Location	56
Figure 65. Adding 01 Button to Button Canvas	56
Figure 66. Removing Grid and Main Camera	57
Figure 67. Description Error Prompt	57
Figure 68. Moving Description out of View.....	58
Figure 69. Deleting Blue Canvas and Description.....	58
Figure 70. Moving Buttons to Foreground	59
Figure 71. Rescaling the Touch Button	59
Figure 72. Changing Font Size	60
Figure 73. Formatting the Buttons	60
Figure 74. Inserting a New Button.....	61
Figure 75. Modifications to New Button	61
Figure 76. Changing Fonts.....	62
Figure 77. Downloading the Title.....	62
Figure 78. Adding Raw Image.....	63
Figure 79. Importing Title Asset.....	63
Figure 80. Attaching the Asset to the Raw Image	64
Figure 81. Final Product of Title Screen UI	64
Figure 82. Creating a new Scene	65
Figure 83. Adding a canvas to the scene.....	65
Figure 84. Change the Canvas to Scale with Screen Size.....	66
Figure 85. Adding a raw image to the canvas.....	66

Figure 86. Using the Rect tool to resize the raw image	67
Figure 87. Adding a button to the canvas	67
Figure 88. Using the move tool to change the location of the button	68
Figure 89. Changing the text and size of the text in the button	68
Figure 90. Adding an Input Field to the canvas	69
Figure 91. Changing the size of the placeholder text.....	70
Figure 92. Adding text to the canvas	70
Figure 93. Adding a C# script to Unity.....	71
Figure 94. InputToText code	73
Figure 95. Recycleable code	74
Figure 96. RecycleableDatabase code	75
Figure 97. RecycleableItem code.....	80
Figure 98. Create and Empty GameObject.....	81
Figure 99. Dragging and dropping the “InputToText” script to the empty GameObject	81
Figure 100. Adding a Runtime Function	82
Figure 101. Picking the correct Function for the Runtime Function added to the button	83
Figure 102. Adding a function to the text box	83
Figure 103. Adding category buttons to the manual sorting page	84
Figure 104. ChangeScene code.....	87
Figure 105. Adding functionality to Category buttons	87
Figure 106. Adding a return button to the type sections.....	88
Figure 107. ReturnButton code.....	88
Figure 108. Finished Metal page	89
Figure 109: Blue Bin Scene	90
Figure 110: Black Bin Scene	90
Figure 111: Green Bin Scene.....	91
Figure 112: Garbage Bin Scene	91
Figure 113: Opening Package Manager.....	92
Figure 114: Installing Vuforia Engine AR Package	93
Figure 115: Adding AR Camera.....	93
Figure 116: Tagging AR Camera as Main Camera	94
Figure 117: Getting License Key	94
Figure 118: Adding License Key	95
Figure 119: Opening Vuforia Engine Configuration	96
Figure 120: Vuforia License Key	96
Figure 121: Unity Vuforia Engine Configuration.....	97
Figure 122: Adding Target Database.....	97
Figure 123: Creating Database.....	98
Figure 124:Add Image Target.....	98
Figure 125: Creating Image Target.....	99
Figure 126:Showing Distinguishing Features.....	100
Figure 127: Appropriate Image Target Features.....	100
Figure 128: Downloading the Target Database	101

Figure 129: Importing Custom Package	102
Figure 130: Creating Image Target in Unity Scene	103
Figure 131: Target Image Inspector Tab	104
Figure 132: Adding a Canvas	105
Figure 133: Adding an Image to Canvas	105
Figure 134: Setting a Source Image	106
Figure 135: Left Side Editor Navigation Bar	109
Figure 136: Button Linked to Another Page	110
Figure 137: Zee Maps Embedding Process	110
Figure 138: Embed Feature in Left Navigation Bar	111
Figure 139: HTML Code for Special Recycling Locations Map	111
Figure 140: Creating the Sort the Recycling Game	112
Figure 141: Adding Buttons	112
Figure 142: Linking the Button to External Websites	113
Figure 143: App Manager - Wix Forum	113

List of Tables

Table 1: Acronyms.....	9
Table 2: Glossary	9
Table 3: Bill of Materials of App.....	51
Table 4: Equipment List of App	51
Table 5: Bill of Materials of Website	108
Table 6: Equipment List of Website	108

List of Acronyms and Glossary

Table 1: Acronyms

Acronym	Definition
AR	Augmented Reality
UI	User Interface

Table 2: Glossary

Term	Acronym	Definition
Hypertext Markup Language	HTML Code	Type of code used to structure web page content

1. Introduction

The purpose of this product is to help and encourage people to recycle, especially in the Ottawa region. This product contains an app and a website. Both the app and website are complex and include many parts. The process and thought process behind the design is also complex and very detailed. Producing a manual to describe the list of designs and parts would be useful to understand how to use the product and how to replicate or modify the product for further improvement.

The purpose of the user manual is to list all of the components of the product which includes the app and website. The components include the design, design process, thought process behind the design and simply the parts of the app and website itself. Along with the listing of the components, there are explanations of every part of the app and the website. The explanations include how to use the app and website, how to replicate the designs, how the designs were built, the reasoning behind the design and the design process. Other details such as the test plan, bill of materials, list of equipment, troubleshooting, support and most importantly the instructions on how to use the app and website are also part of the user manual. In addition, the user manual features many figures from the app and website which will assist the user in navigating, using and understanding the product.

2. Overview

Recycling is a global initiative that is needed to reduce the impact of waste and help preserve the planet. The product designed was created to facilitate recycling for homeowners to ensure that each item is placed in the correct bin and recyclables do not end up in the landfill. This product is expected to be simple, efficient, accurate and available to everyone. The product 'Eh, Are You Recycling?' has many different aspects that increases user friendliness and expands the availability of the app to as many people as possible because it is a phone application as well as a website that can be access on any computer, tablet, or phone.

2.1. App

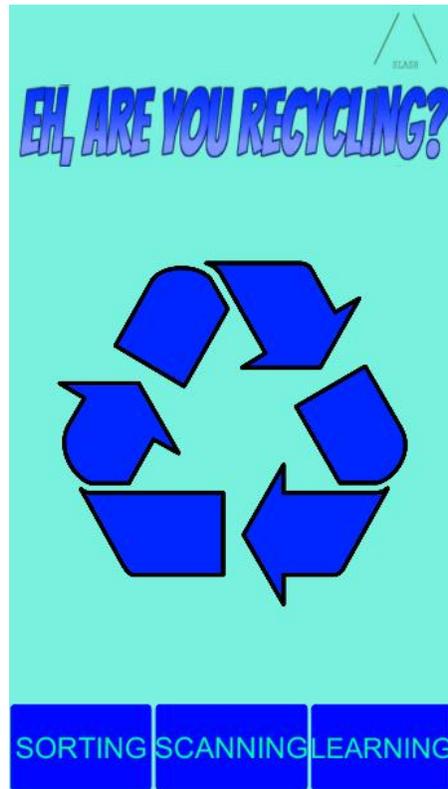


Figure 1: App Home Page

The app opens to a clear and easy to navigate home page with the title of the app placed squarely in the center of the design. Our company logo is displayed right above it. Along the bottom of the home screen, three buttons take the user to three separate pages: manual searching, camera search, and education. The manual search allows the user to locate where their item should go through button prompts with the section or the user can type the name of the item into a search bar. The camera search allows users to scan bar codes to and will pop up an image of which bin the item should go into. The learning page provides the user with a list of links that go to recycling centric websites like the Ottawa recycling page.

2.2. Website

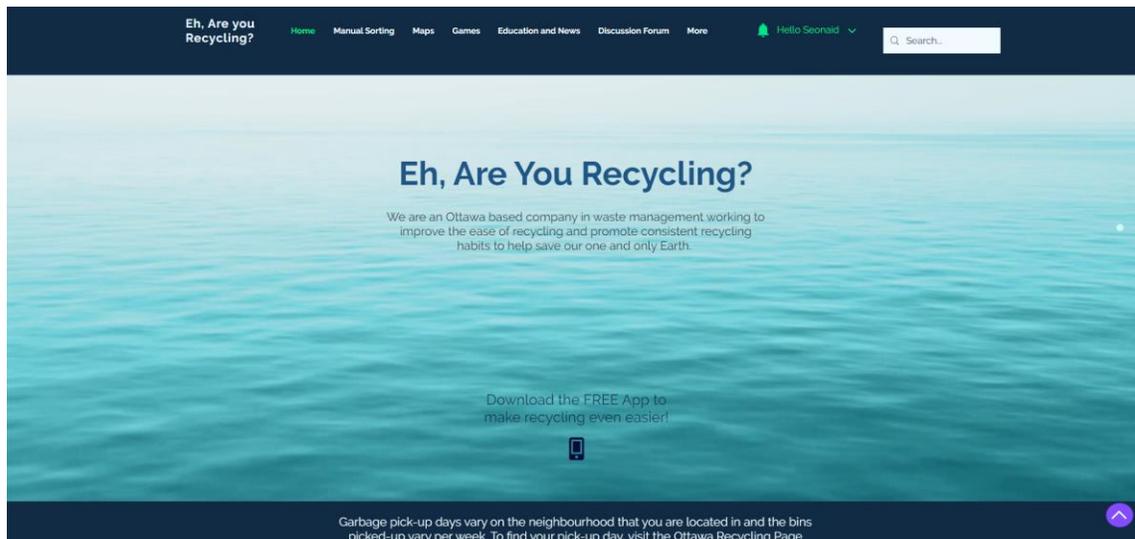


Figure 2: Website Home Page

The website is easy to navigate with the navigation bar at the top of the page. The key features of the website are the manual sorting page and search bar. These two functions allow users to find any item by searching or by following the step-by-step breakdown of their item to find the correct bin. Along with these two functions, there are also two maps that show locations where special recyclables can be brought and locations of waste bins on the University of Ottawa's campus. There is also an educational page, a waste sorting game, and a discussion forum for members to post on.

3. Getting Started

3.1. App

The app is set up through a multitude of steps as the app must first be downloaded to Unity before set onto the phone.

3.1.1. Set-Up Considerations

Before the app can be started on the user's phone, they will need to download the app into Unity and from Unity download it onto their phones after setting up the Unity application.

3.1.1.1. Installing Unity

Since the app is currently still in development it is not widely accessible. In the future, the app will be put on the app store and the users will be able to easily find the app into the app store and download it onto their phones.

To access the app, the user currently must have access to Unity 2019.4.22f. The user can download Unity 2019.22 through Unity hub which is the recommended method. Unity hub can be downloaded from the Unity website: <https://unity3d.com/get-unity/download> (figure 3).

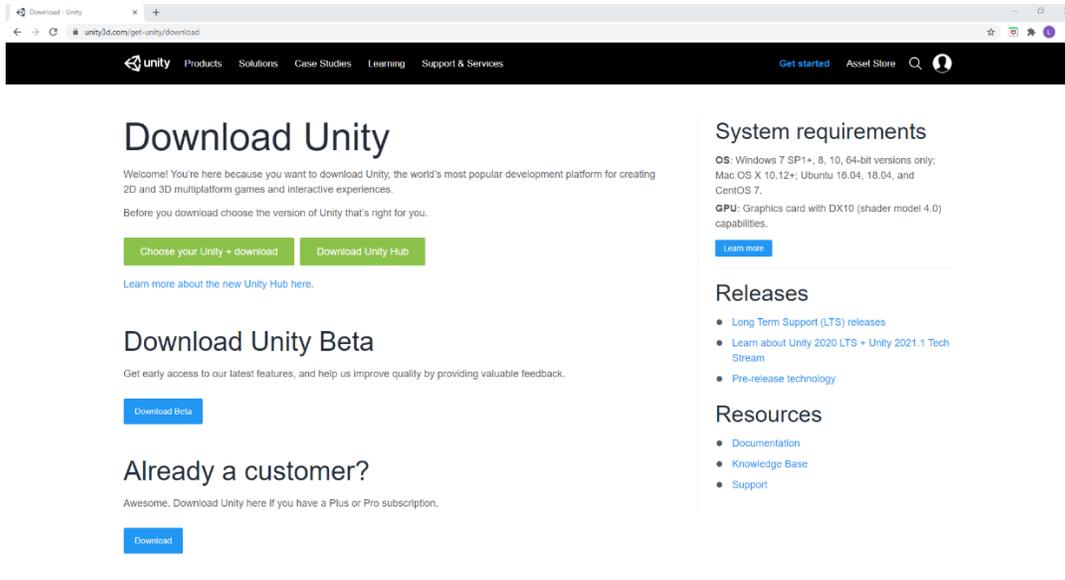


Figure 3: Unity Website to download Unity Hub

From the Download Unity page, the option to download Unity hub will appear at the top of the screen. To start the download, click on the green Download Unity Hub and a menu will open for the user to choose where to download Unity hub installer on the user's computer (figure 4).

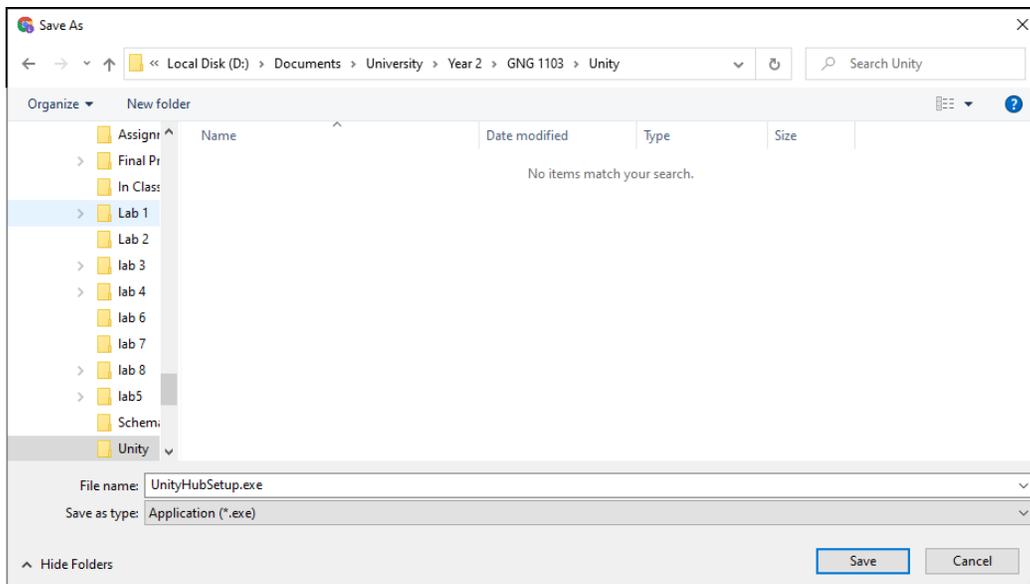


Figure 4: Pop up menu to choose where to download the Unity Hub installer

The user will choose a location and click the save button. Once the installer has been downloaded, the user should open the installer, the computer will ask if the client would like to run the program and the yes option should be chosen. Then, the unity installer will open a license

agreement menu for the user to read through before agreeing to install Unity hub by clicking the “I Agree” button (figure 5).

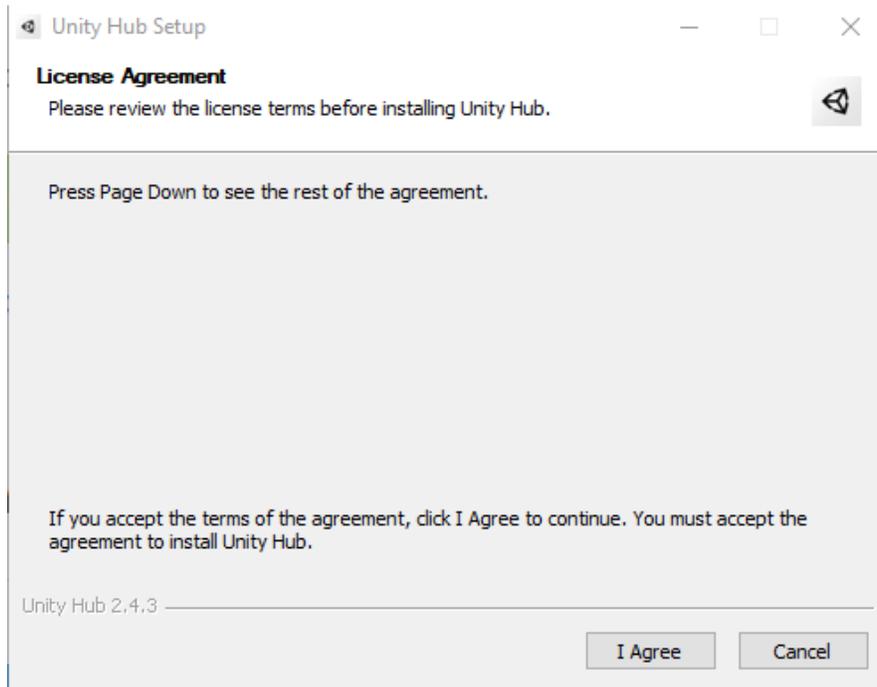


Figure 5: Pop up when the Unity Hub Installer is opened

This will launch the installer and install Unity hub onto the user’s machine. A prompt may open about choosing a location to install Unity hub. Chose a suitable location and press download. Once the download has finished, the installer will open a page to inform the user that Unity hub has been successfully downloaded. From here, the user will check off a box that says, “Run Unity Hub” and then press the “Finish” button (figure 6).

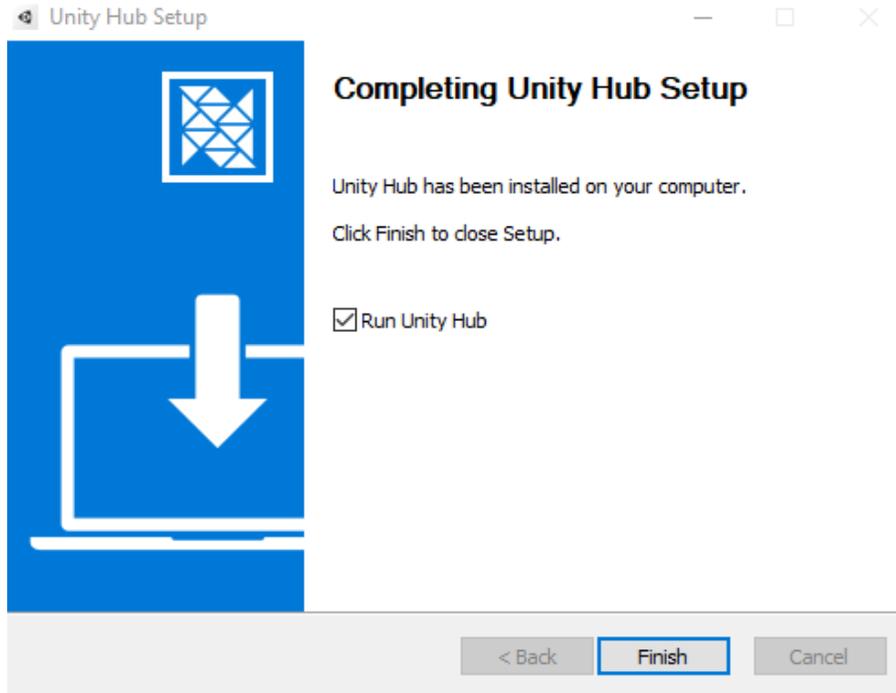


Figure 6: Pop up when Unity Hub is ready to begin installing

Unity hub will then open to the projects page (figure 7) which will show the user the projects they are currently working on. From here, the user should press the “Installs” button on the left side of the page (figure 8).

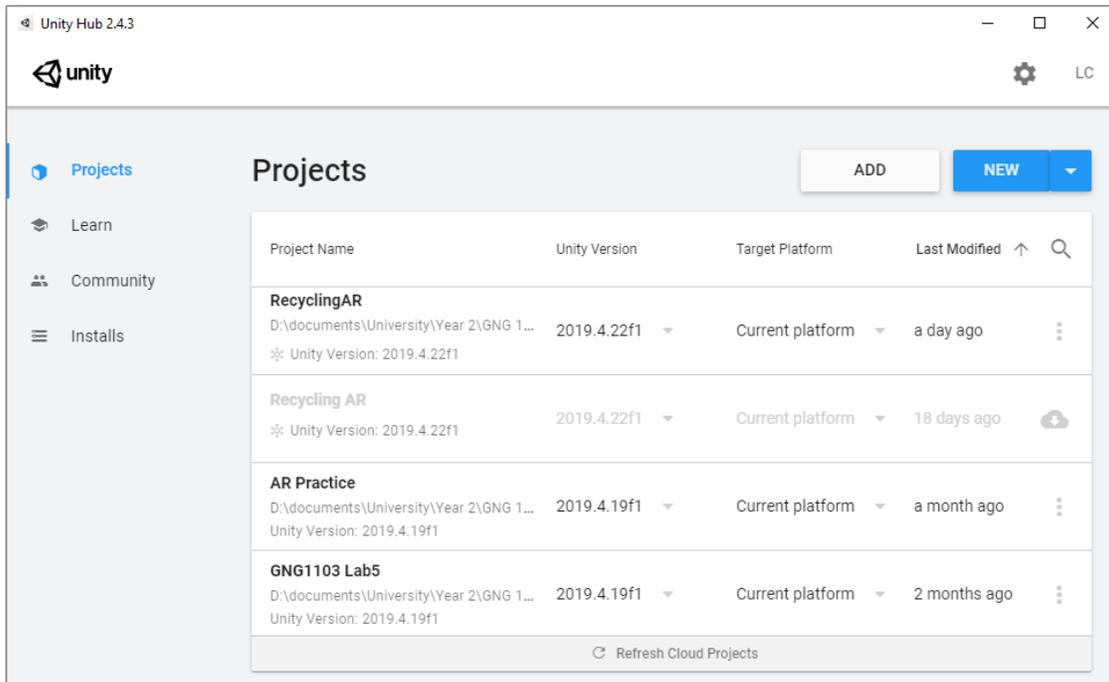


Figure 7: Opening page of Unity Hub

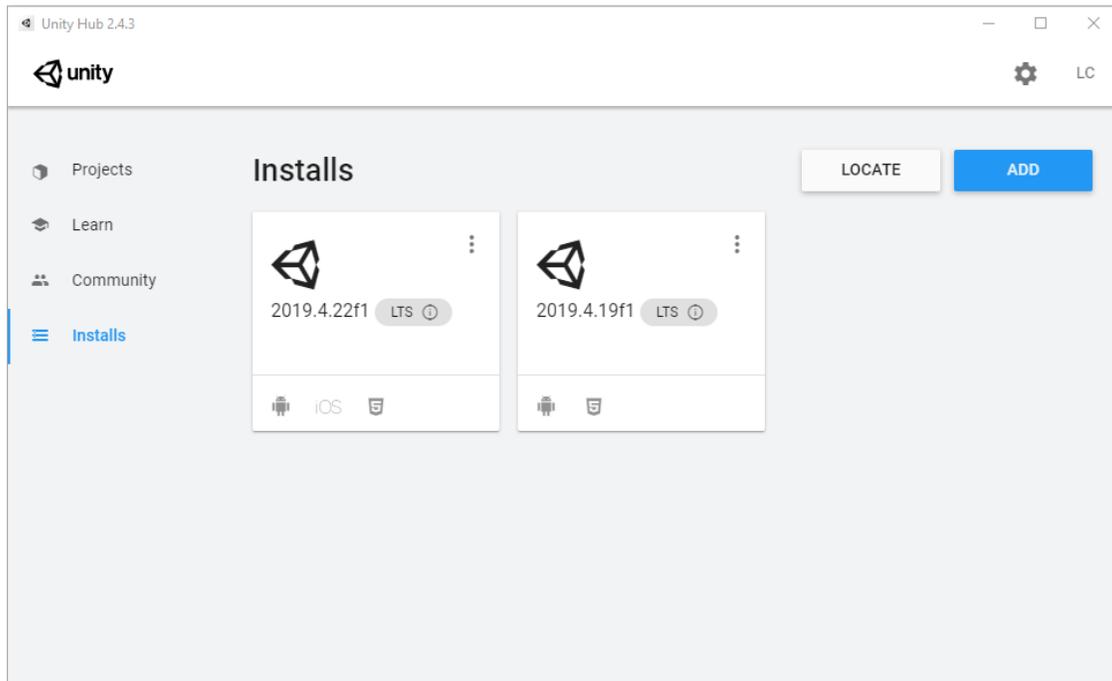


Figure 8: Install page of Unity Hub

The installs page shows which version of Unity have been downloaded to the user's computer. To add Unity 2019.22, the user must click the blue add button. However, Unity 2019.22 does not show up on the quick add menu, so the user must click on the link to visit the download archive: <https://unity3d.com/get-unity/download/archive> (figure 9).

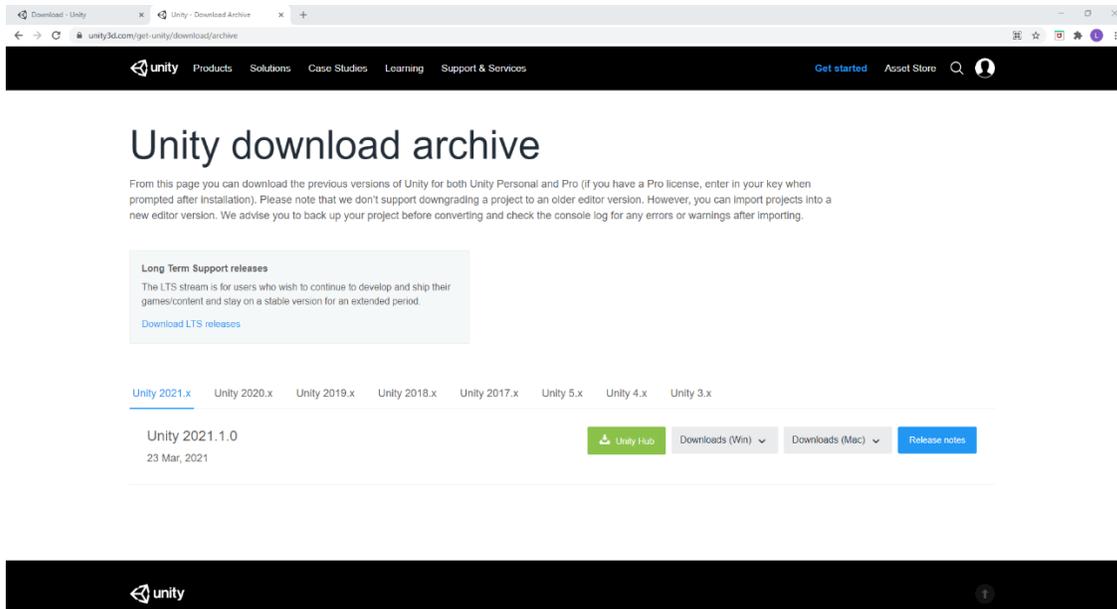


Figure 9: Webpage to access older versions of Unity

From the website, the user must click on the “Unity 2019.x” tab to find the Unity 2019 versions. Once the user has opened the Unity 2019.x tab, they can scroll down to the page to find the Unity 2019.22 version and click on the green Unity Hub button (figure 10).

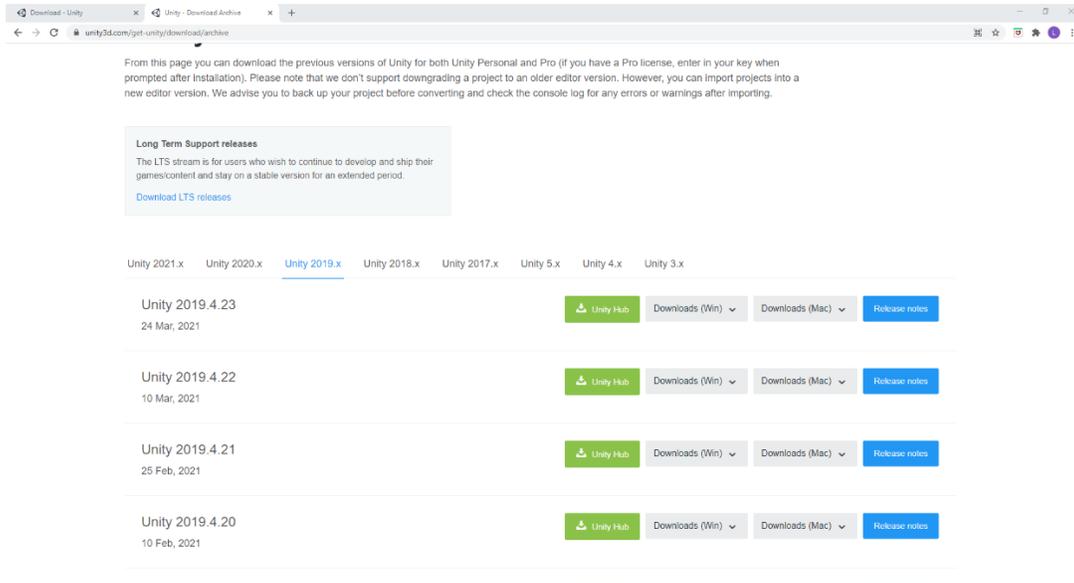


Figure 10: Webpage to download the 2019.4.22 version of Unity

Clicking on the green Unity Hub button will open a page that asks the user if they would like to open Unity Hub, so the user should click the “Open Unity Hub” button (figure 11).

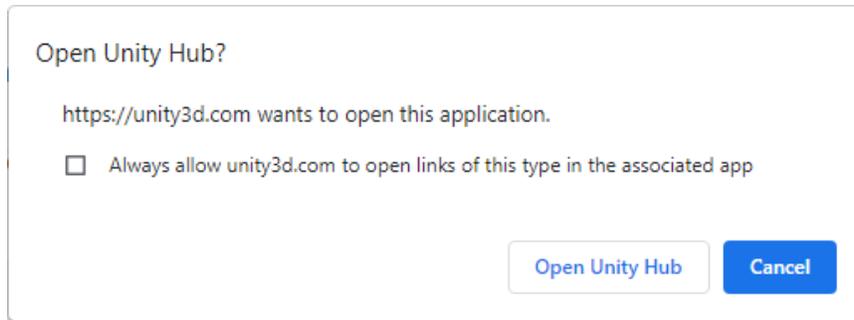


Figure 11: Pop-up that opens once the download button is pressed

Unity Hub will now open and reveal an “Add Unity Version” page that will require the user to choose which portions of the Unity Version they would like to add. Ensure that the box under “Dev tools” is checked off and also check off the box for either “Android Build Support” or “iOS Build Support”. The user should also open the drop down menu to ensure that all box under the heading are also checked off (figure 12).

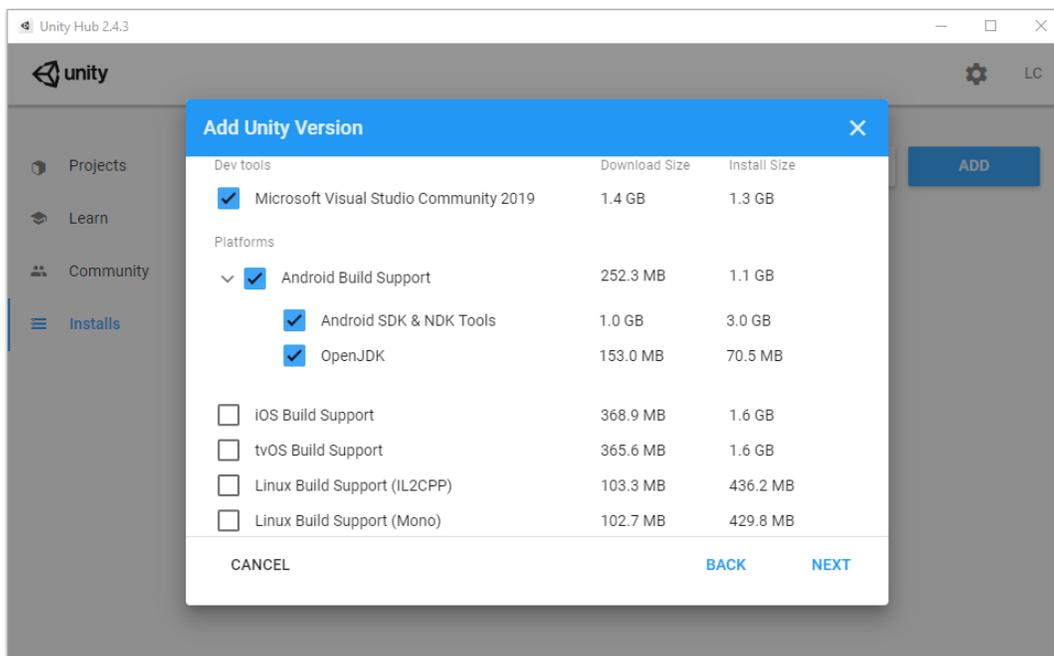


Figure 12: Pop-up menu in Unity Hub to install the version of Unity

Then click the “Next” button. This will prompt the user to read through multiple license agreement. The user should read through the terms before clicking the agreement box and clicking next (figure 13). When the user has reached the last agreement box, “Next” button will be replaced with a “Done” button. Clicking on the “Done” button will begin the installation of the Unity version onto the user’s computer.

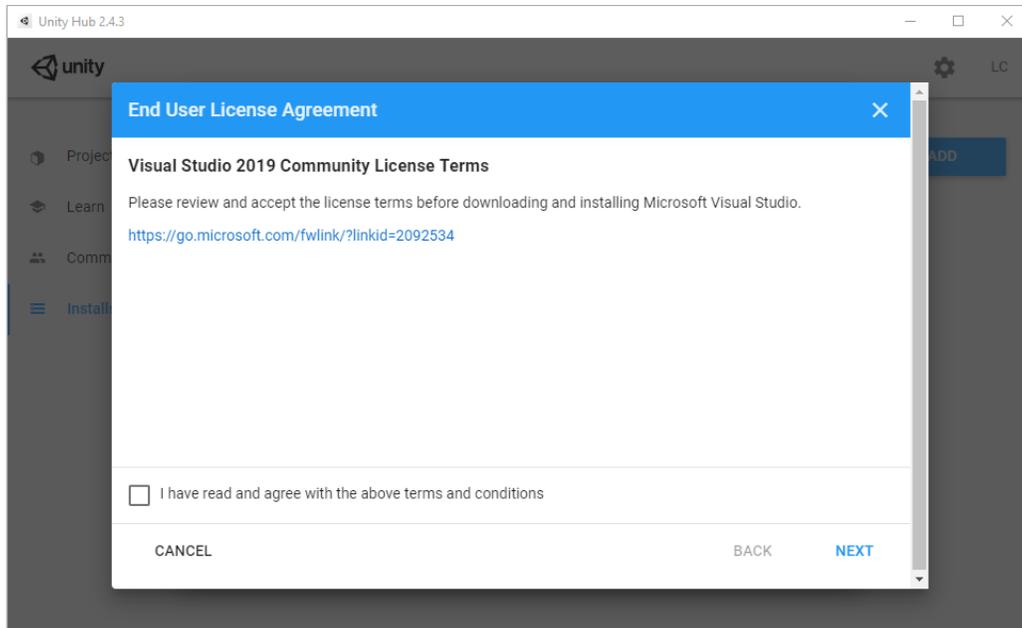


Figure 13: User agreement menus that appear before install can occur

3.1.1.2. Installing the App to Unity

The Tools UnityHub and GitHub must be used to install the app created onto Unity through a repository that is shared. To download the app, the user must go onto the main branch of the repository (figure 14) on the GitHub browser and press a green button in the top right corner that says “Code” and select “Download Zip” (figure 15). The file cannot be downloaded onto a OneDrive/Cloud location on the computer or else Unity will not be able to find a correct path for installation. Once this file has been downloaded, extract the file and open UnityHub,

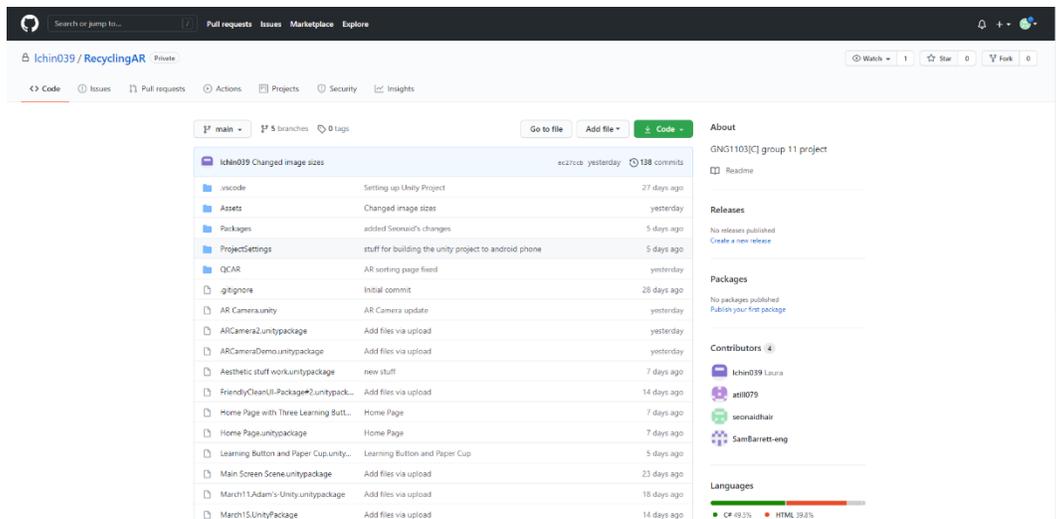


Figure 14: GitHub Repository - Main Branch

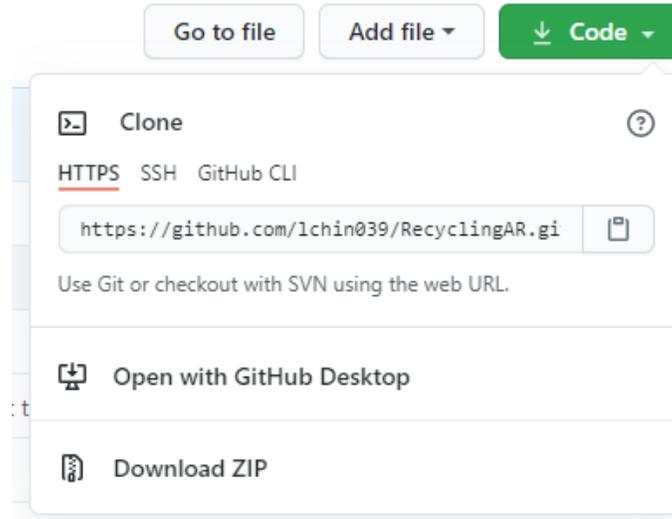


Figure 15: Code Button Dropdown - Download Zip

Select “Add” in the top right corner (figure 16) of UnityHub, and select the file from the computer. Version 4.19.22 of Unity must be used to create a project with this file as this was the version that it was created on and is not compatible with other versions. See section 3.1.1.1 on instructions to install a new version of Unity. Once it has been uploaded (figure 17) and a new project has been created using the file downloaded from GitHub, the project can be opened and edited.

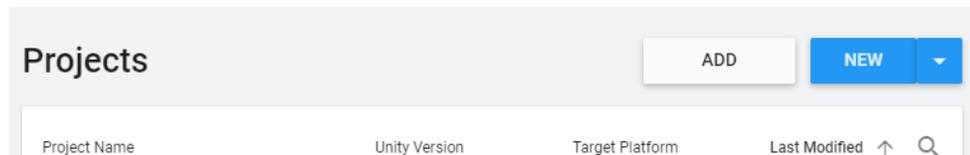


Figure 16: UnityHub Add Button

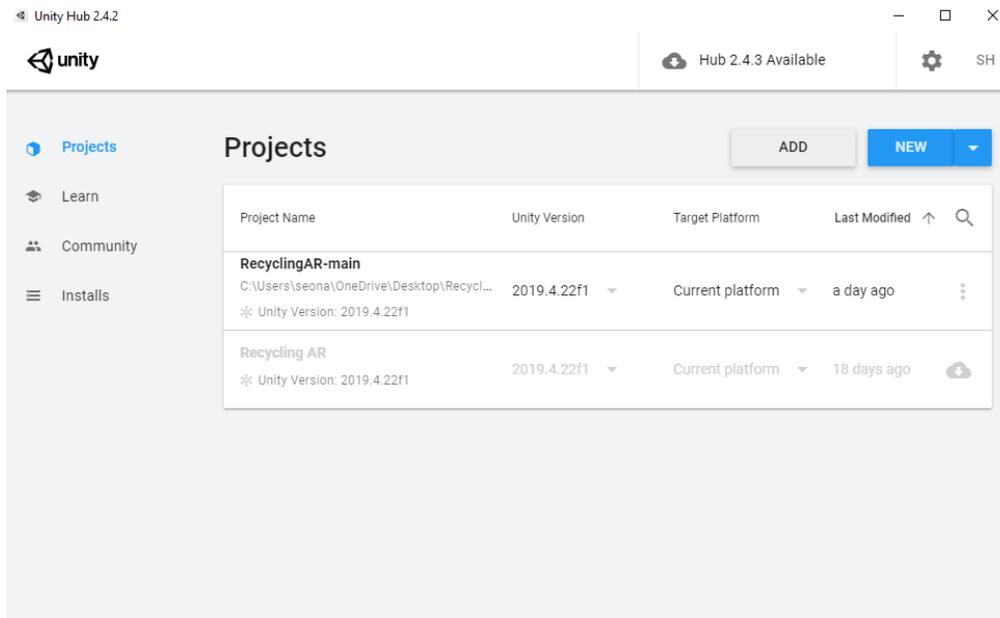


Figure 17: UnityHub - Downloaded Project

3.1.2. Preparing the App Scenes for Use

Once the app has been installed to Unity and has been opened, the app is still not ready to work. The user will need look at their project window at the bottom of the screen (figure 18) and click on the folder titled “Group 11 Work”.



Figure 18: Opening screen when the Unity project is opened

Open the folder and then click on the folder named “App Scenes”. Here the user will find a row of all the scenes used in the app. The user should first click on the scene named “Unity Project save 2” and then open the file window. From the file window click on the option “Build Settings” (figure 19).

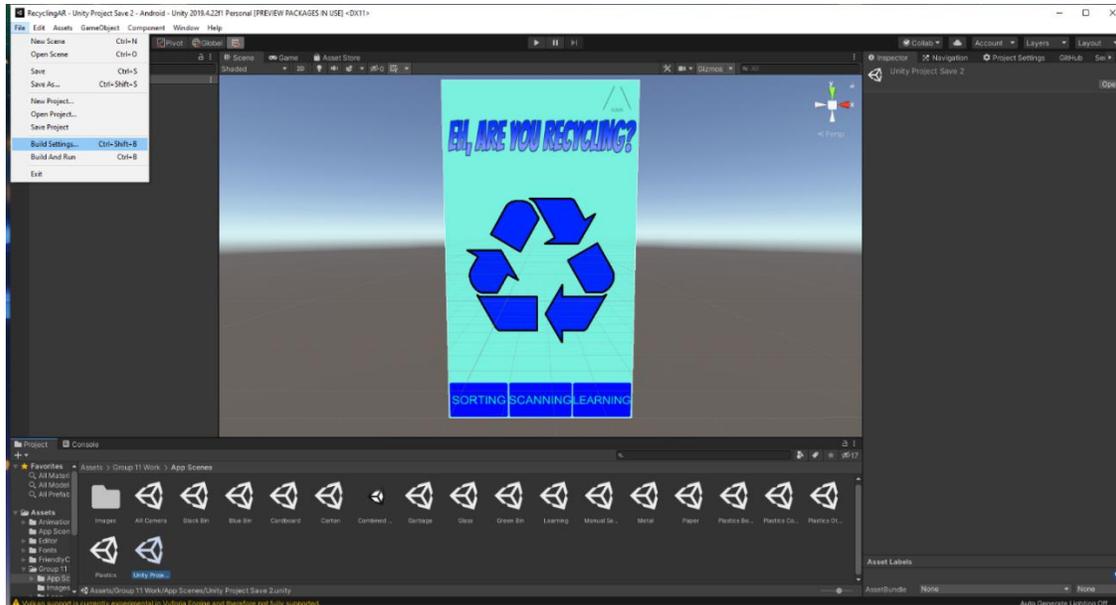


Figure 19: Unity Project save 2 page while opening the Build Settings

From the build settings menu the user will need to add the app scene to the “Scenes In Build” list at the top of the opened menu. To add the scene to the list, the user will need to click the “Add Open Scenes” button at the bottom right of the menu (figure 20). Each Scene will need to be added one at a time, so the user will need to click on every scene in the “App Scenes” menu from the Project window. This will allow that buttons to transfer the scene from one page to another.

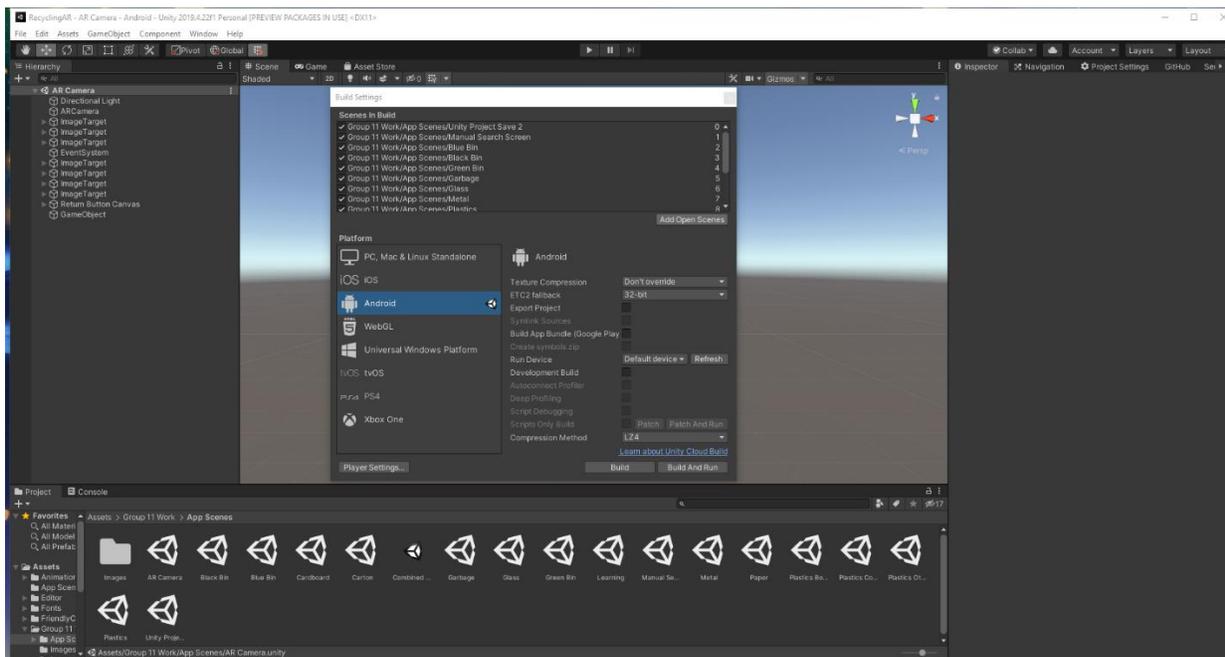


Figure 20: Build Settings Menu with Android support

Once all the scenes are added, the AR engine used in the app must be enabled. The user will need to go to the “Window” button in the upper left of the screen and open the dropdown menu. From there, the user should click on the selection called Package Manager. (figure 21).

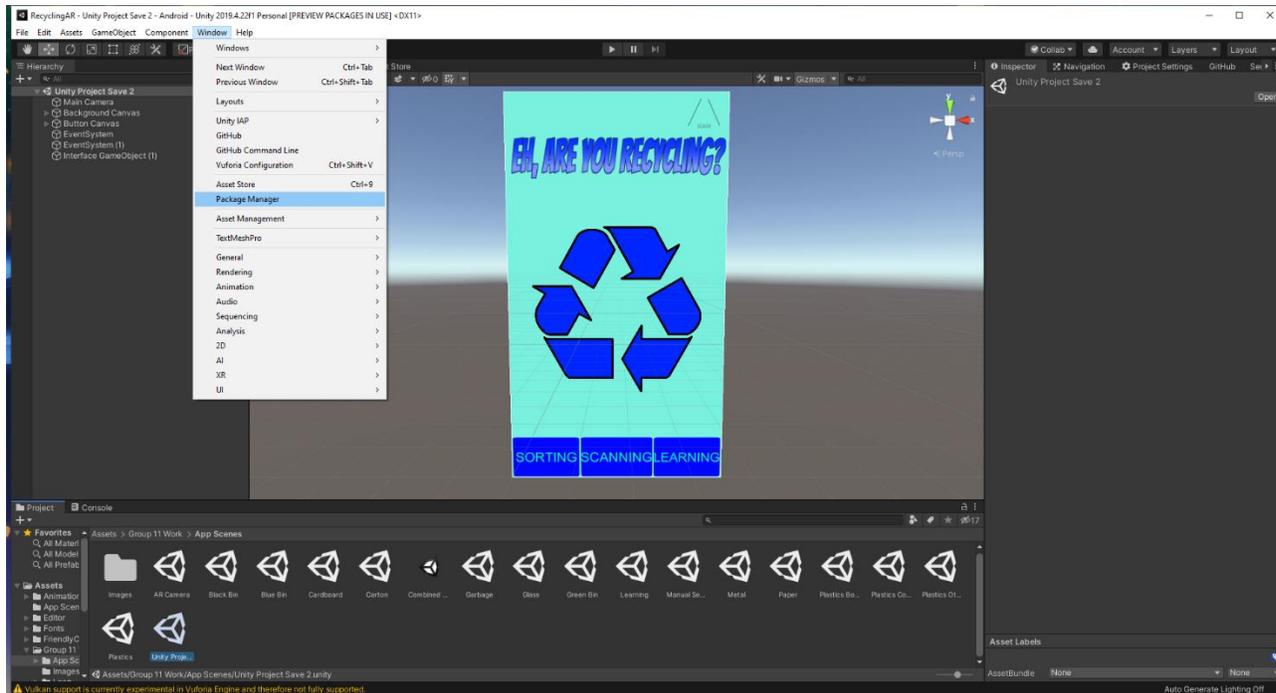


Figure 21: Location of Package Manager in Unity

In the “Package Manager” menu, go to the search bar at the top and type in “Vuforia” and press the search button. This should now show a single option in the menu on the left side called “Vuforia Engine AR”. Press the “Install” button on the bottom right to install Vuforia, the AR engine used in this project, to Unity (figure 22).

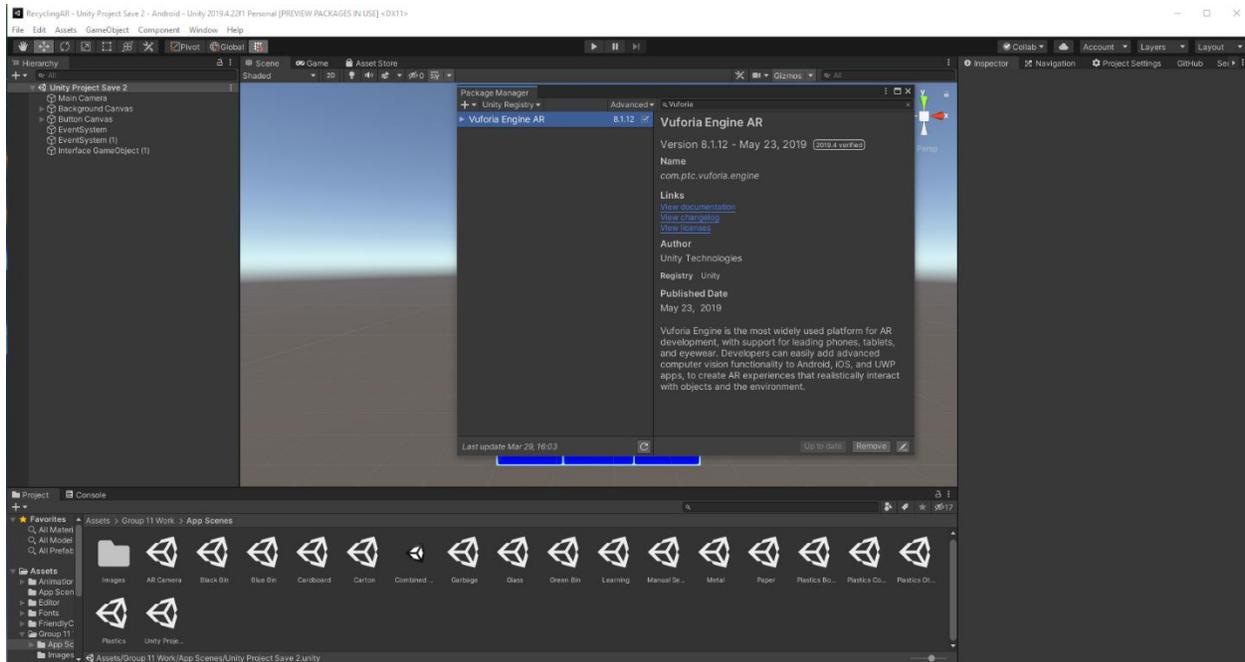


Figure 22: Package Manager menu while searching for the Vuforia Engine

Now the app is almost ready to work. The user now needs to navigate to the scene called “AR Camera” from the Project window. Once there, the user should go to the inspector window to the right of the screen and click on the object called “ARCamera” which will populate another menu on the opposite side of the scene called “Inspector” (figure 23)

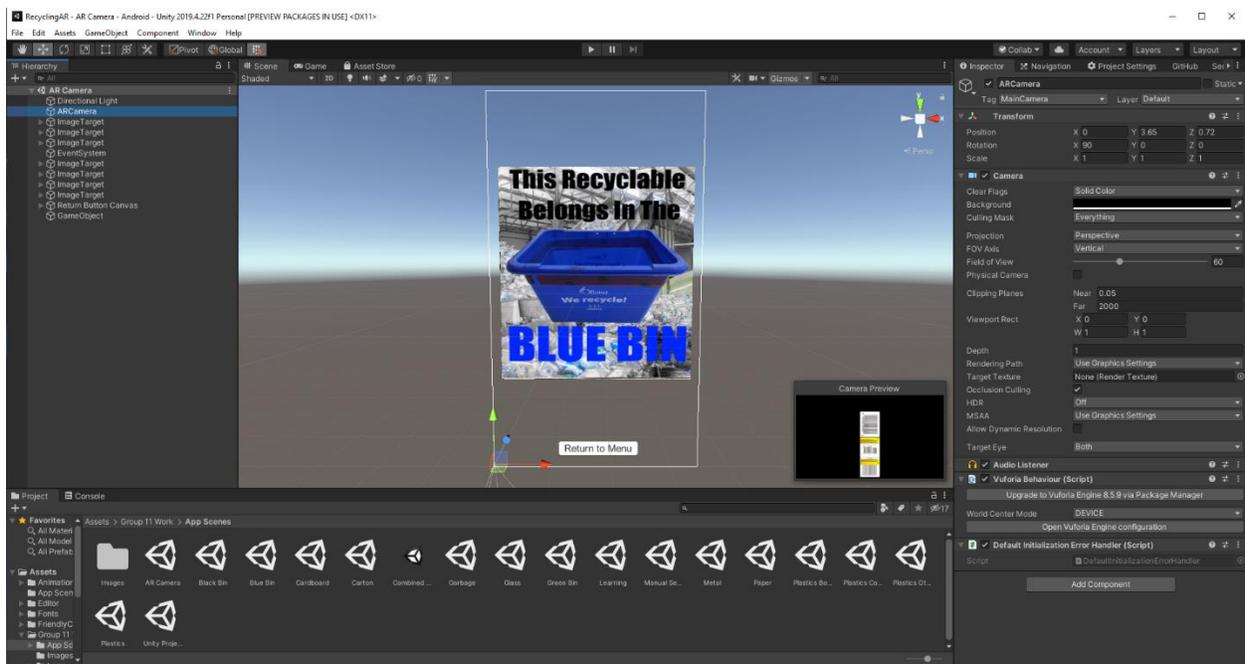


Figure 23: Inspector window for ARCamera

From the inspector menu, the user must locate the button labelled “Open Vuforia Engine configuration” at the bottom of the inspector window. This will open another menu overtop of the inspector window. In that window, there will be large blank box that the user must copy and paste the following code into the box (figure 24):

```
AT83afX/////AAABmU2HG48UUUfmjII8miP1P21sl6coVtgSGc0tLolod3eKJkyiRi24rPyNi8jX
70nxx+YwnkavsZ0QsRldxziNpCHwtTCyN8OPwVW/uN6pWhDcwb2BNVPQeoFOucWb7xsq
YJxv7yQDoB/zhU8QDoWkonQNNExd1Qq0uNgThfo+hY/QpF05aWNV16Shz7pPKzZD3TDZ
wXrDJ9gE0PJj2bWKKPS0uiJrdTvXNAffKeqo1MowNzzCwBSem/Vn9eUGig1ZnfDJkeqUyex
Ub0k+qDHCStcMEyBK+MiLxzGrCFix0egvqJAGgZRMQNDUaekU+HqgzSY7F/KgJCjxs/KL
uGO5vEc1TpvizBBShgPPIJYeMeHK
```

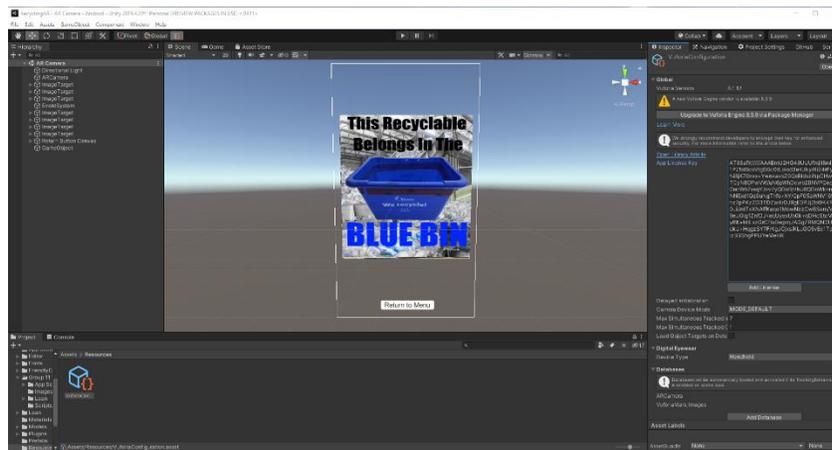


Figure 24: Vuforia configuration menu with inputted code

The app will now function on Unity. By pressing the play button on the center top of the screen, the app can be sampled within Unity.

To build the app onto the user’s phone, the user will need to plug their phone into their computer and allow their phone to download apps not from the app store. From Unity, the user can press the file button on the top right corner of the page and click the button “Build and Run”. This will cause Unity to build the game and install it onto the user’s phone before running the app on the phone (figure 25).

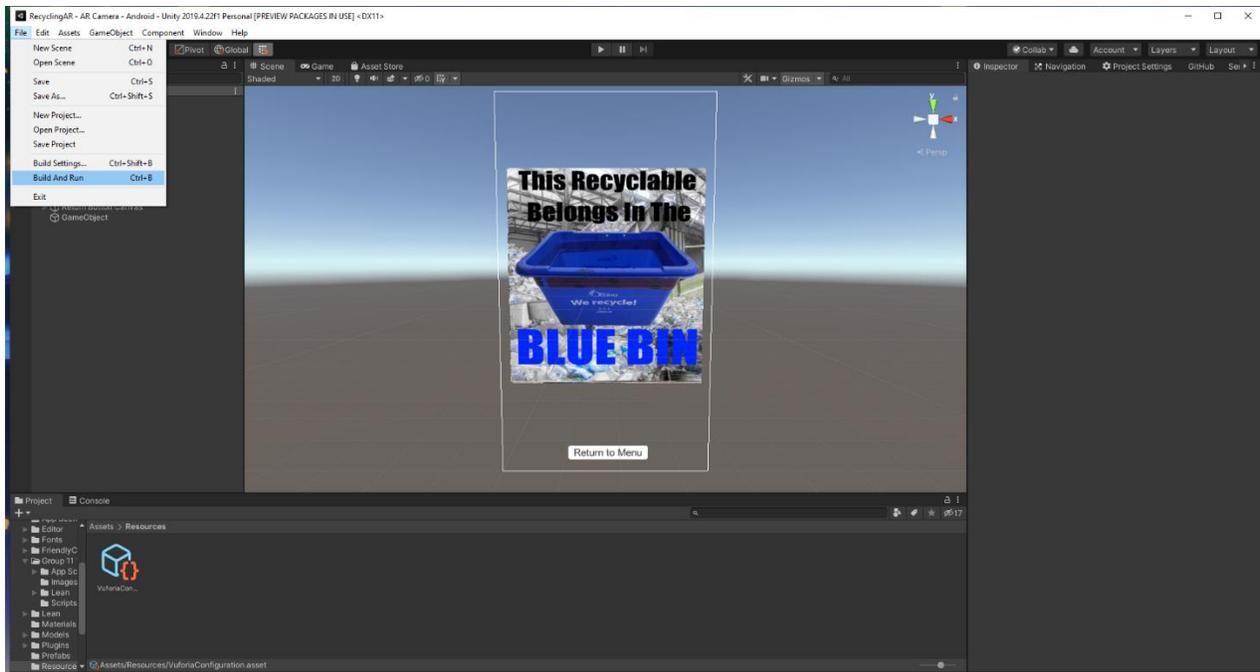


Figure 25: Unity with file open to locate the Build and Run button

3.1.3. User Access Considerations

Since the app has not been published to the app store, before the app can be installed onto the user's phone, the phone must be turned to developer's mode. This section will go through a step-by-step procedure on how to turn an android phone into developer's mode and install the app onto the phone.

To turn developer's mode on in android phones, the user must go to their phone's setting's page and scroll down to the bottom of the settings page where the user will find a section labeled "About phone" (figure 26).

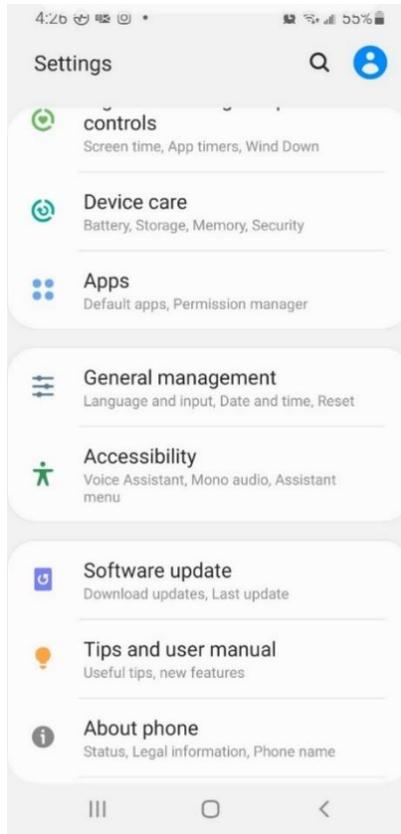


Figure 26: Bottom of Setting menu on Android phone

Once the user is on the “About phone” page, they will be able to press the button called “Software information” (figure 27).

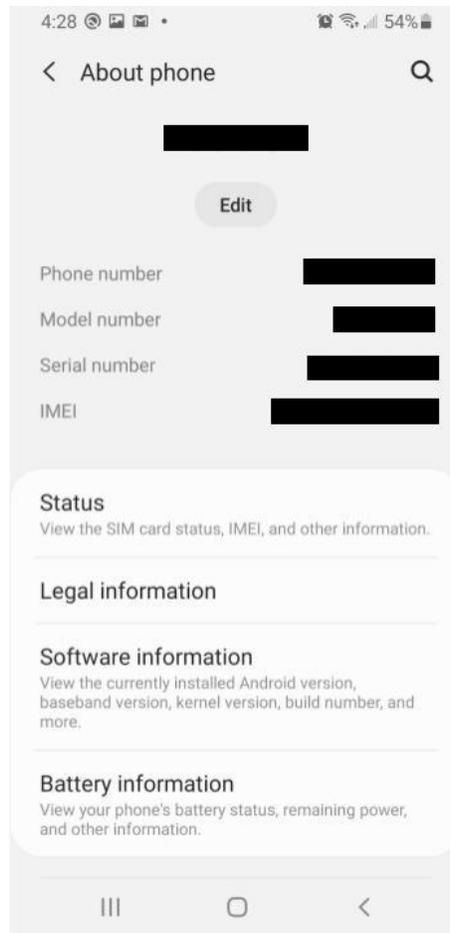


Figure 27: About phone menu

After opening the “Software information” page the user will need to click on the item called “Build number” 9 times. This will activate developer’s mode on the phone and allow the user to download apps not from the app store which is necessary to add “Eh, Are you Recycling” onto the user’s phone in its current state.

3.1.4. Accessing the System

Once the app has been downloaded onto their phone, the user will simply be able to click on the app icon to launch it. Once launched, the user will be brought to the app’s main screen (Figure 1) from which they can reach all the available features.

3.1.5. System Organization & Navigation

From the main screen, three buttons are displayed at the bottom of the page: “SORTING”, “SCANNING”, and “LEARNING”. These buttons allow access to each of the different scenes designed with Unity.

Clicking the “SORTING” button will bring the user to the manual sorting page (Figure 35). This page displays buttons for the user to press which will bring them to different category specific screens, such as the “Metals” page (Figure 36). To return to the main screen, the user can easily tap the “Return to Menu” button at the bottom of the screen.

From here, the user can click the “SCANNING” button to go to the AR scanning page and the “LEARNING” button to go to the education page. Returning to the main screen for both pages works the same way as for the sorting page.

The education page contains several links to trusted external websites providing information and news on recycling. To visit the sites, only a tap on the “LEARN” button beside the website info is required.

3.1.6. Exiting the System

To leave the app and return to the phone’s home page, the user simply does the same for all other apps on their phone. This can be done by clicking their phone’s home button or swiping up on their screen, depending on its type and generation.

3.2. Website

The website has a very straight forward set-up for use.

3.2.1. Set-Up and User Access

All that is required is an electronic device that can connect to the internet, such as a phone, computer, tablet, etc. and a browser. This website can be found on any search engine.

To get full access to all the features on the website, the user will have to create a profile to become a member. This will allow the user to post and interact on the discussion forum and allow notifications from the website. To create a profile, select the ‘Log In’ button in the top right side of the header and sign up using a Facebook or Goggle account or sign up with an email.

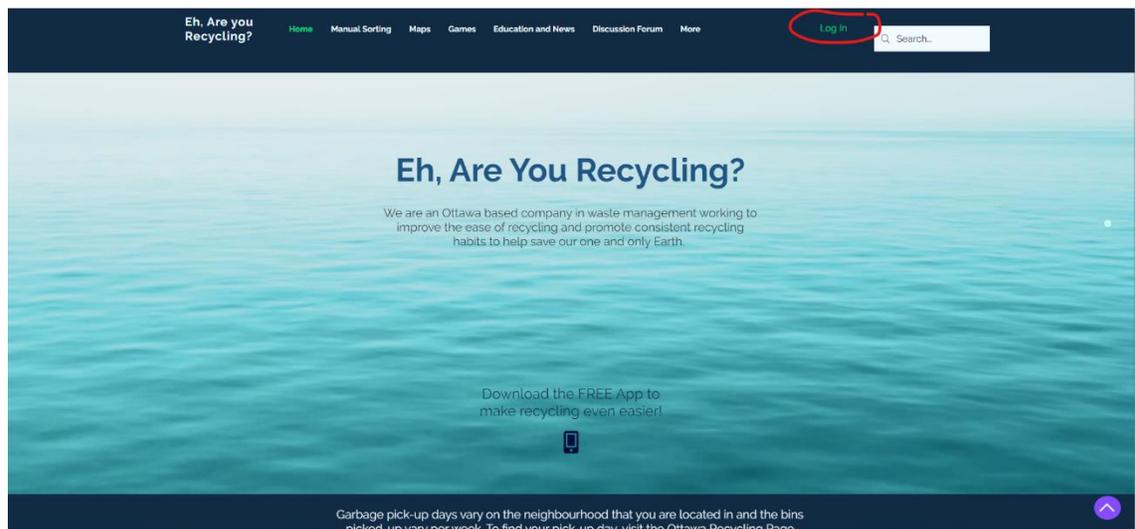


Figure 28: Log In Button on Home Page

3.2.2. System Organization & Navigation

The website is very easy to navigate because there is a navigation bar (Figure 29) at the top of all the pages. This allows the user to switch between pages easily and all available pages can be seen constantly.



Figure 29: Navigation Bar on Website

The website has also been organized to make it more user friendly with animated buttons that change colour (figure 30) to show they can be clicked to lead to external websites and instructions for certain features to help users understand the functionality of the website (figure 31). The search bar function also allows users to find key words throughout the website to make the website even more navigable (figure 32). The website can be accessed at any time by anyone.

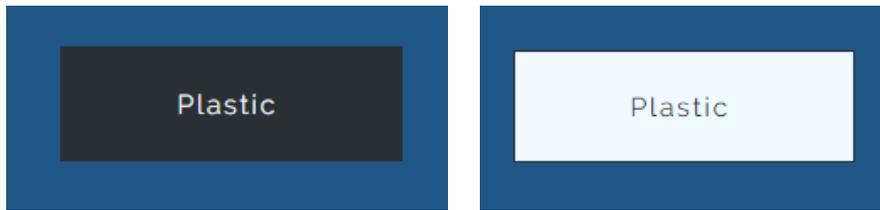


Figure 30: Button Animation Colour Change

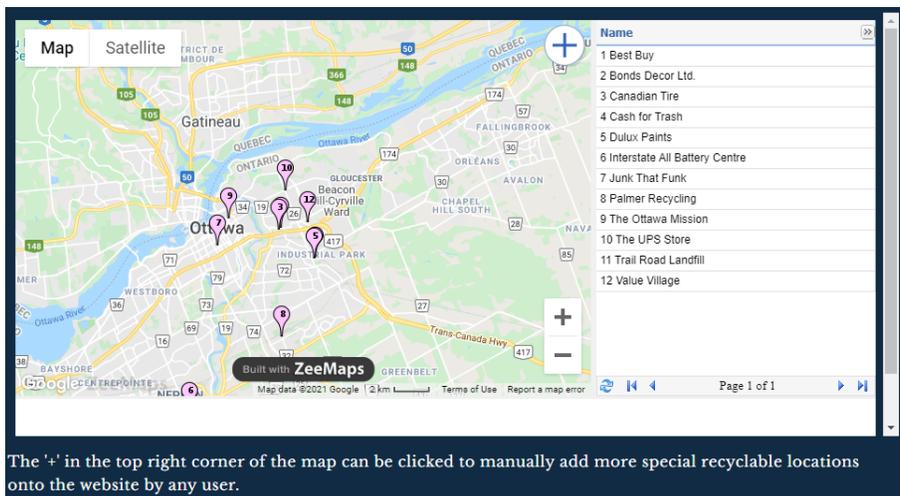


Figure 31: Example of Instructions on Website

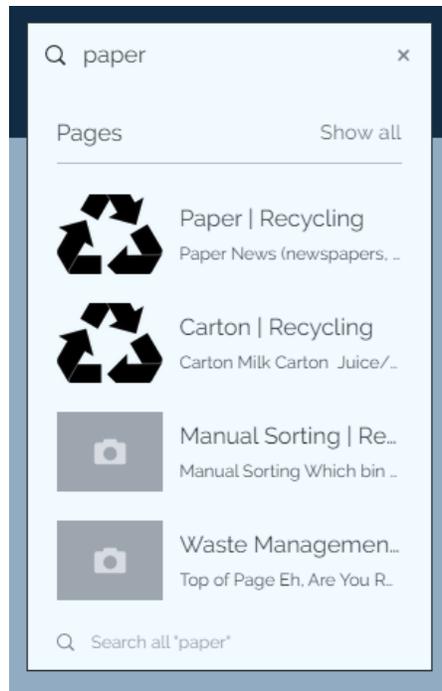


Figure 32: Search Bar with Dropdown Results

4. Using the System

This section describes how to use the product.

4.1. App

This section describes how to use each feature of the app.

4.1.1. UI

The home screen contains the product title, a recycling symbol, and three buttons to help traverse the app (Figure 1). These buttons switch to dedicated screens when interacted with by the user's finger. They also provide small animations when pressed, such as resizing the button, changing its colour, and producing clicking sounds to demonstrate confirmation of the user's actions. The "SORTING" button leads to the manual sorting page, the "SCANNING" button leads to the AR camera, and the "LEARNING" button leads to the Education page.

Throughout the development of this app, the user interface was changed periodically to better capture focus and present its designs and features. Originally, the main screen was formatted to fit a computer screen, had a social button rather than learning, and contained clashing colours between the app name and the background.



Figure 33: Original Home Screen

Additionally, the manual sorting page was a blank canvas with only a smaller version of the search bar.

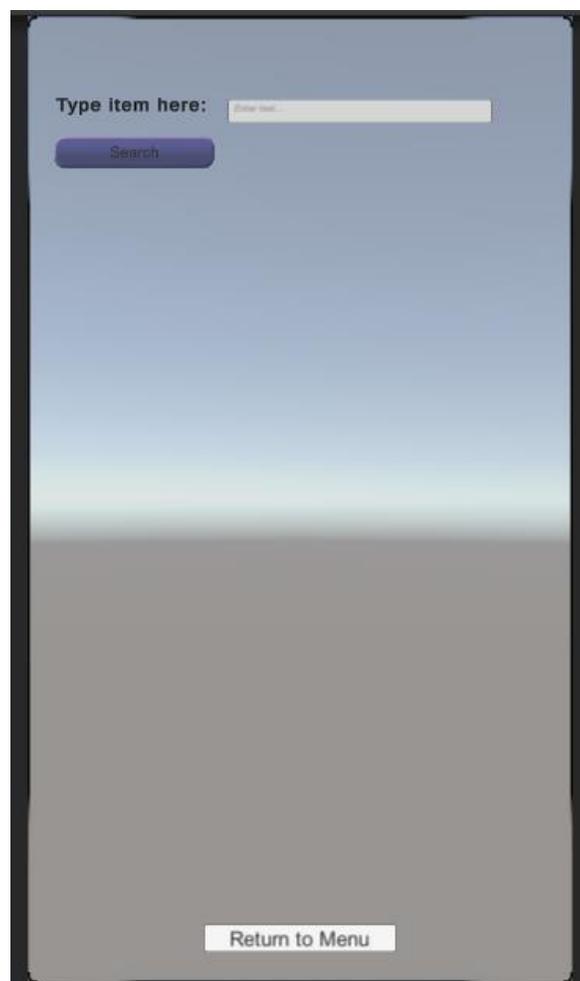


Figure 34: Original Search Screen

Improvements were made to each screen to provide more images, buttons, and functions.

4.1.2. Sorting

The Sorting button leads to the Manual sorting page (figure 35) that hosts 2 main methods of sorting items. The method that commands the most presence is the category options while the other is the search bar at the top of the page. To return to the main menu, a button at the bottom of the page called “Return to menu” can be pressed.



Figure 35: Manual sorting page

The user can click on any of the buttons for 6 categories and the manual sorting page will change to a category specific page (figure 36). The category specific page will have a list of items along with an image of an example of said object along with an arrow that points to the right-hand side of the page where the location of the item is located. Located at the bottom of the page is a button called “Return” which will return the user to the manual sorting page.

Metal



Figure 36. Specific item category sorting page

The “Plastics” button operates slightly different as it brings the user to another menu of options to further specify the type of plastics looking to be recycled (figure 37). This menu has a “Return” button that will bring the user back to the sorting page. However, the plastics “Return” buttons will bring the user back to the plastics page instead of the sorting page.

Plastics

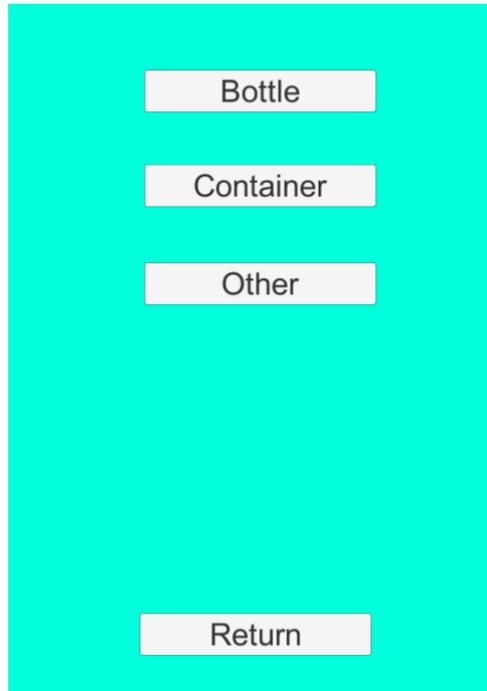


Figure 37: Plastics sub-section search page

The other main feature of the sorting page is the search function at the top of the screen. By tapping the “Enter text...” box at the top right, the user can search a specific item by typing the name of the object into the box. When the box is tapped on the phone, a keypad will open for the user to begin typing their object. Below the box, a list of items from the app’s database will begin to appear after typing in two letters (figure 38). If the user mistypes a word, the items in the box below will disappear to inform the user they are not on track.

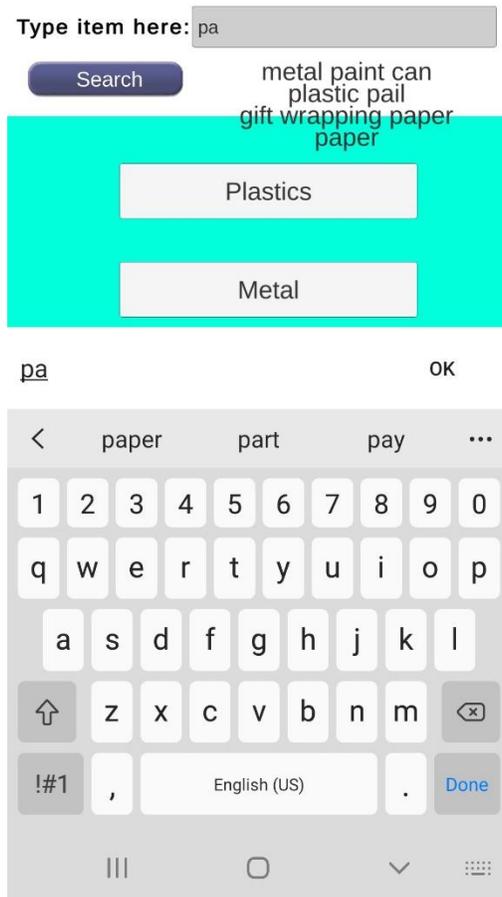


Figure 38: Example of item search in search bar

Once an item has been typed into the text box, the user can press the “Search” button which will cause the program to search the database for the item (figure 39).

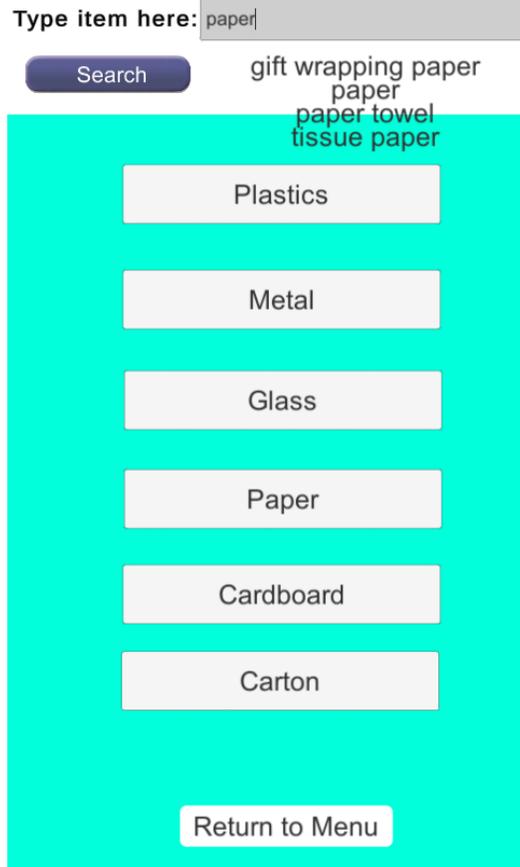


Figure 39: Item typed into search bar and ready to be searched

Pressing the “Search” button will result in two different possibilities. If the item is in the database, a new scene will load which tells the user the bin that the search item belongs in (figure 40). This scene will have the location printed in words along with a graphical representation of the Ottawa bin. At the bottom of the screen, a “Return” button will bring the user back to the manual sorting page.



Figure 40: Item location scene for searched item in database

Alternatively, if the item is not in the app's database, it will print out an error message in the text box that listed items (figure 41). The error message will inform the user that the item they are looking for is not registered and advise them on how to better use the search bar in the future. By returning to the search bar, the user can retry, and the error message will be replaced once more by the item suggestions.

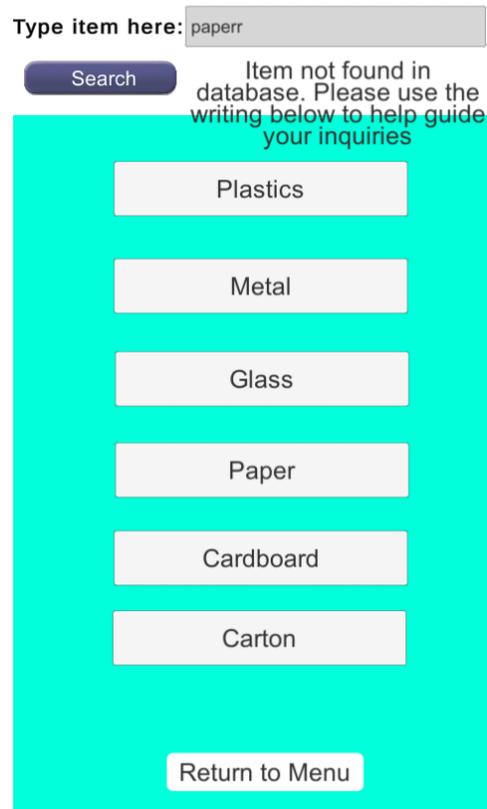


Figure 41. Error message when an incorrect item is searched

4.1.3. Scanning

To access the scanning function of the app, click on the scanner button located in the middle at the bottom of the home page.

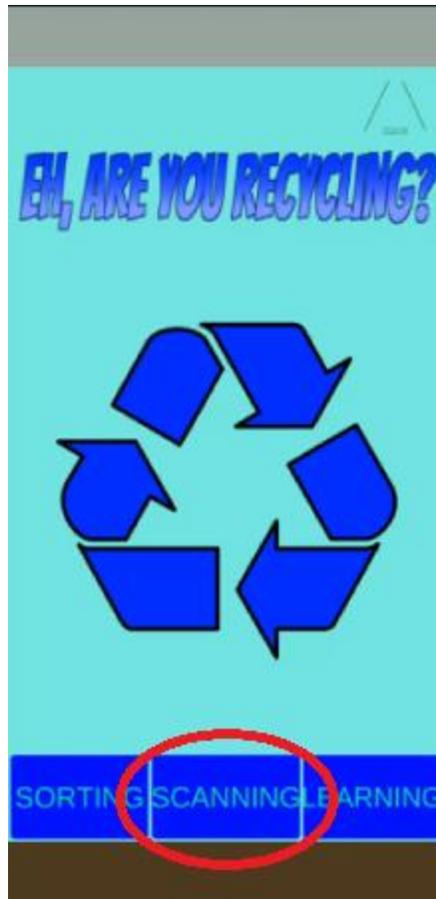


Figure 42: Accessing the Scanning Function

Place bar code in a position where it is clearly visible to the AR camera and the scanner will automatically display a notice corresponding to the correct bin to deposit the item. For optimal conditions ensure that the item is well lit.



Figure 43: Utilizing the Scanning Function

To return to the home screen click the button at the bottom of the screen labeled “Return to Menu”.



Figure 44: Exiting the Scanning Function

4.1.4. Learning



Figure 45: Learning Page

The purpose of the app is also to be educational and teach the user how to recycle and learn other stuff about recycling. When the user goes to the learning tab, they get directed to the screen above which contains a few links to educational sources about recycling. Users can access the links by using the “Learn” button. The links labelled with a title and described by some text.

4.2. Website

This section describes each page of the website.

4.2.1. Manual Sorting

The Manual Sorting feature of the website has a step-by-step breakdown to find the correct bin to place a desired item in. The first page has six buttons for different general categories; plastic, metal, glass, paper, cardboard and carton and it has general information about recycling symbols that are commonly found on recyclables (figure 46). Each button on the main manual sorting page leads to another page for further breakdown and items with their corresponding bins (figure 47).

Manual Sorting

Which bin does your item go in?

Follow the step-by step buttons to find the right bin and do your part

Plastic
Metal
Glass

Paper
Cardboard
Carton









Ottawa accepts all numbers of plastic except for hard #6 plastics. This is the category of expanded polystyrene (ie. styrofoam containers, packaging, coffee cup lids, meat trays, foam clam shells)

Figure 46: Manual Sorting Page

Metal

Aluminum Foil (clean or food soiled)	→	Black Bin
Can	→	Black Bin
Chip Bags	→	Garbage Bin
Jar Lids	→	Blue Bin
Empty Aerosol Cans	→	Blue Bin
Aluminum Containers	→	Blue Bin
Scrap Metal	→	Special Location Return
Metal Clothes Hangers	→	Garbage Bin

Figure 47: Metal Sorting Page Breakdown

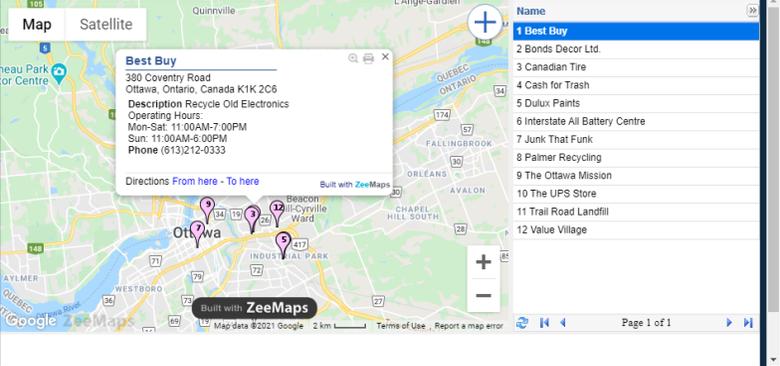
4.2.2. Maps

There are 2 map functions on the website

4.2.2.1. Special Recyclables Map

Special Recycling Locations

Some items cannot be disposed in your home, such as batteries, paint, old electronics and more. This is a list of some locations around Ottawa where these items can be disposed of.



Name
1 Best Buy
2 Bonds Decor Ltd.
3 Canadian Tire
4 Cash for Trash
5 Dulux Paints
6 Interstate All Battery Centre
7 Junk That Funk
8 Palmer Recycling
9 The Ottawa Mission
10 The UPS Store
11 Trail Road Landfill
12 Value Village

The '+' in the top right corner of the map can be clicked to manually add more special recyclable locations onto the website by any user.

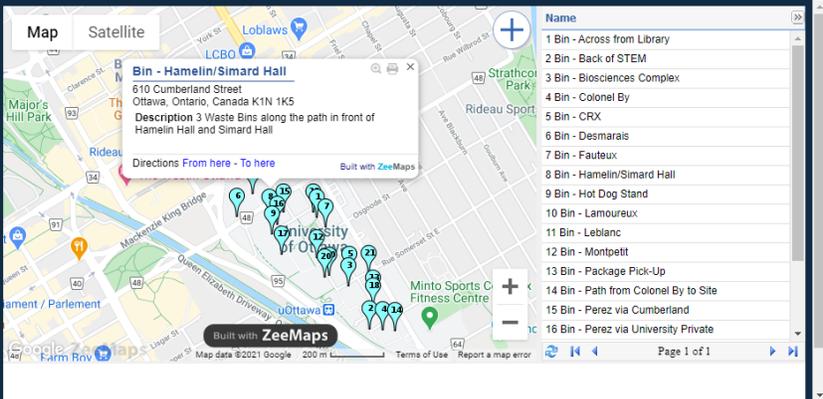
Figure 48: Special Recycling Locations Map

The Special Recycling Locations Map (figure 48) has 12 locations where special recyclables can be brought for proper disposal. Each location is listed on the right side of the map and when selected, it will bring up the address, what items can be recycled there, operating hours and the contact phone number. Users can also add new locations to this map by clicking the '+' in the upper right corner of the map, and then inputting an address and the details.

4.2.2.2. Waste Bins Map

Waste Bins on Campus

Find the waste bin nearest you and ensure to throw away your item properly!



Name
1 Bin - Across from Library
2 Bin - Back of STEM
3 Bin - Biosciences Complex
4 Bin - Colonel By
5 Bin - CRX
6 Bin - Desmarais
7 Bin - Fauteux
8 Bin - Hamelin/Simard Hall
9 Bin - Hot Dog Stand
10 Bin - Lamoureux
11 Bin - Leblanc
12 Bin - Montpetit
13 Bin - Package Pick-Up
14 Bin - Path from Colonel By to Site
15 Bin - Perez via Cumberland
16 Bin - Perez via University Private

The '+' in the top right corner of the map can be clicked to manually add more waste bin locations onto the website by any user.

Figure 49: Waste Bins on Campus Map

The Waste Bins on Campus Map (figure 49) functions very similarly to the Special Recyclables Map and it shows the locations of all the waste bins on the University of Ottawa's campus. The list of locations is on the right-hand side and when locked on, it shows a description of where the bin can be found and specifies if there are more than one bin in that location. The feature for users to add locations is also featured in this map and can be placed anywhere in Ottawa, so as to not restrict this map solely to campus.

4.2.3. Games

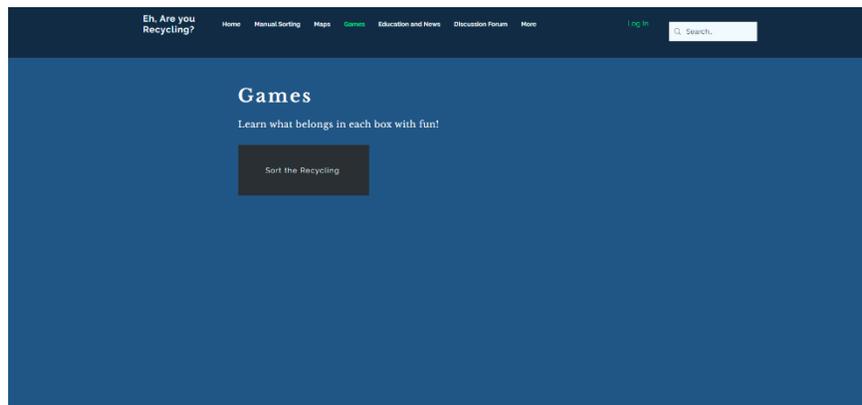


Figure 50: Games Page

The Game Page only has one button at the moment that leads you to a personalized game for the website. It is a Sort the Recycling game (figure 51) and by clicking on the button it will take you to a new tab where the users can play the game.

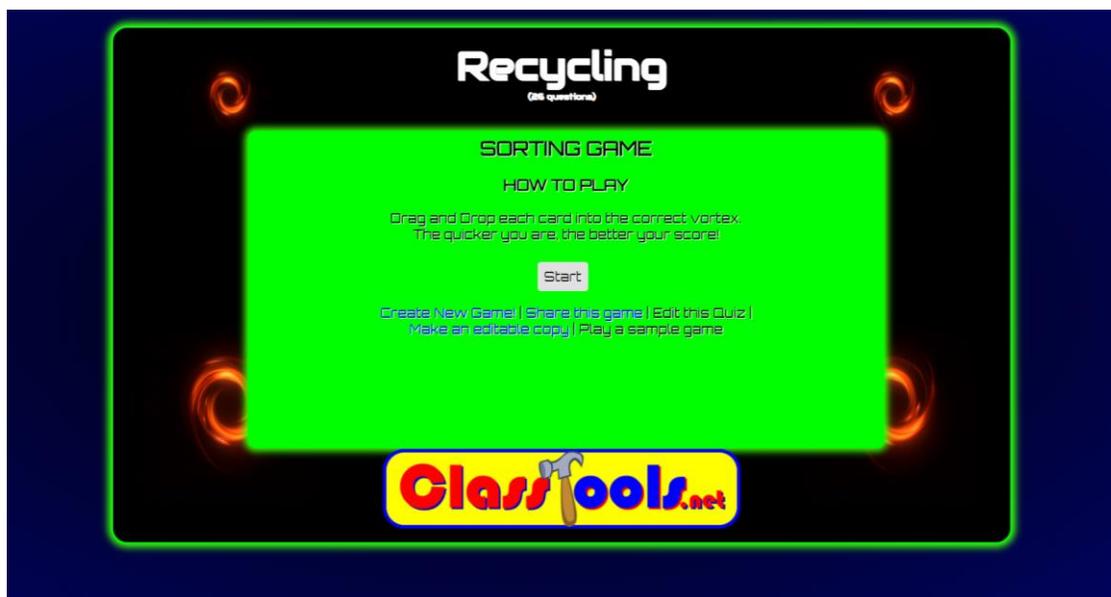


Figure 51: Sort the Recycling Game

4.2.4. Education and News

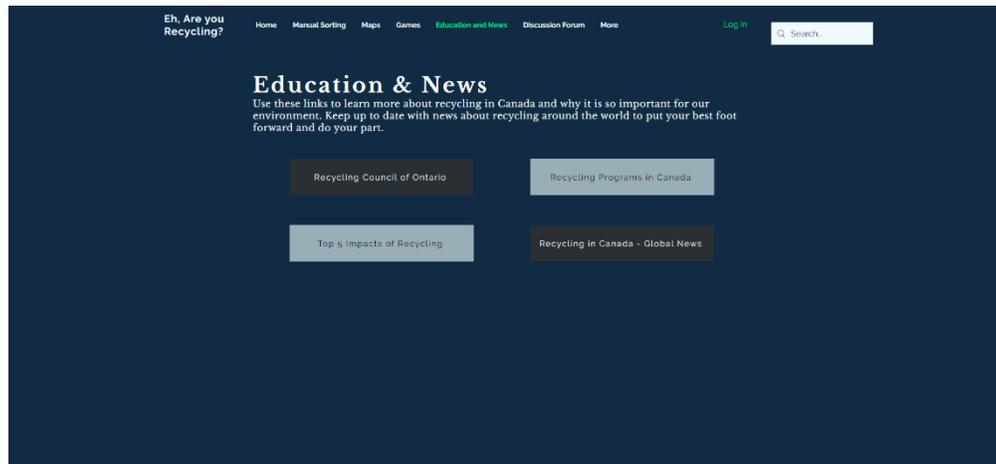


Figure 52: Education and News Page

The Education and News page currently has 4 buttons with links to various external websites. By clicking on any of the buttons, a new tab will be created for the resource. Each link is different with recycling information or it will bring you to a recycling focused news page.

4.2.5. Discussion Forum

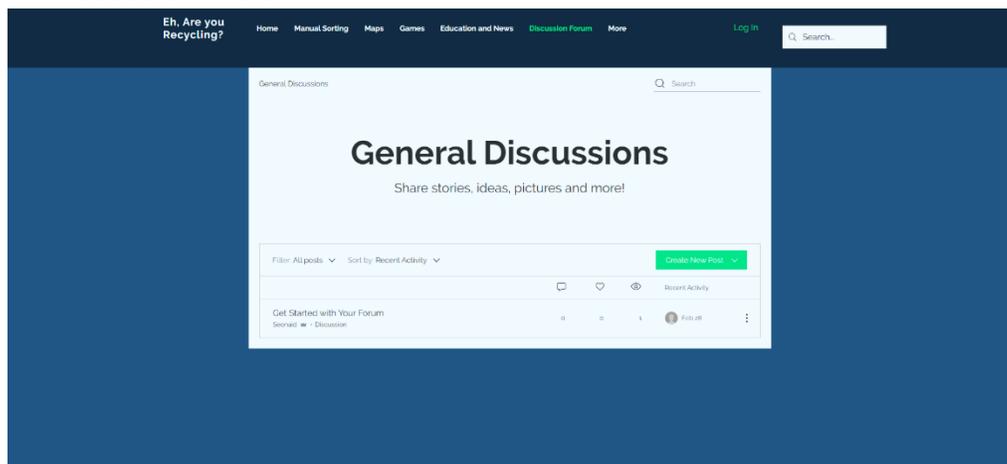


Figure 53: Discussion Forum

The discussion forum allows members to post their own questions or thought, and like and comment on others posts. Once the traffic of the website increases and more members join, the discussion forum will fill up with posts.

4.2.6. Members

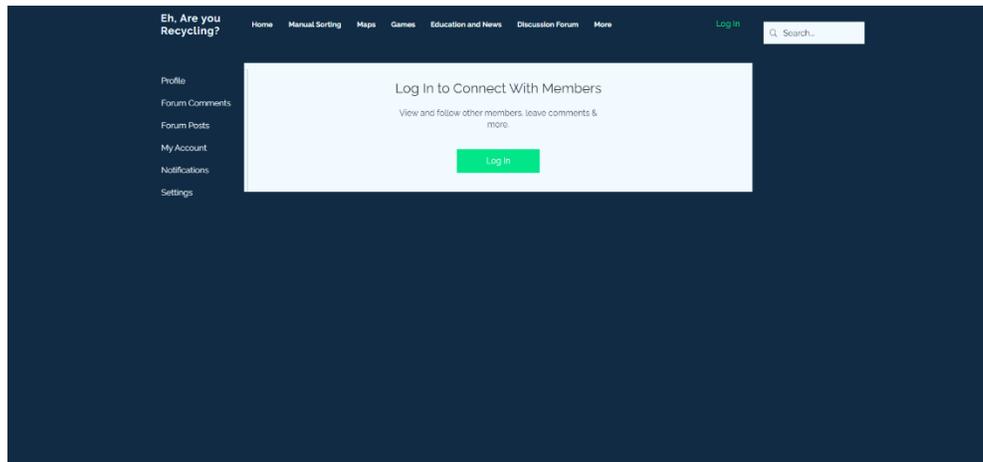


Figure 54: Members Page When Not Logged In

The Members Page is very straightforward with a navigation bar to the left. The initial page when no account is logged in cannot access the features of the page (figure 54). When an account has logged in the main page has the profile of the members of the community and members can then select to follow or view the profile of others (figure 55). Members are also allowed to view their personal profiles, forum comments and posts, and their account settings where they can change their profile and notification settings.

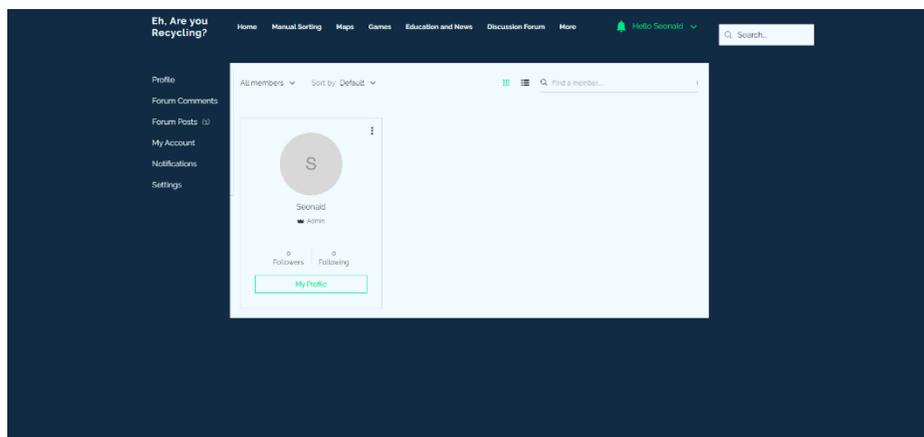


Figure 55: Members Page When Logged In

5. Troubleshooting & Support

5.1. Error Messages or Behaviors

Due the app being constructed in Unity and the website being constructed in Wix, many of the error messages and behaviors are accounted for by the development environment. In the app, the button functionality along with many other features were taken care of by my Unity and the creator did not need to worry about errors. The main places where errors could occur was the manual search engine which modified the existing Unity functionality with C# script.

The greatest error with the search bar is that the user could input any type of text: numbers, letters, emojis. Even with letters which the search bar wants, there is no guarantee that the letters will spell out a word that exists within the app's database. Luckily, the input field converts the input typed into the search bar into a string and does not allow emojis to be typed into the input field. Even if nothing is typed into the input field, the input field still provides the declared string with an empty string.

To account for user error, the return code is initially an error message. If the item is found in the database, the error message is overwritten before the string is printed to the output text box. The given string from the input field is first turned to all lowercase letters to avoid any cap lock issues or capitalization errors. Similarly, the input field automatically trims the string to remove any spaces before or after the text.

The string is then used to search a list of keys for a database of Recycleables. If a match is found, the Recycleable is copied to a new Recycleable item. If an item is not found, the Recycleable item is null which could cause errors when the program attempts to load the scene attached to the Recycleable item.

To ensure that no errors occur, the code that loads the new scene does not run if the Recycleable item is null. This allows the error message to be printed out if the item is not in the database and for the item's scene to be loaded if it is.

Furthermore, this method works even when numbers are typed into the search bar, as anything can be converted to a string, but only the strings that are spelt the exact same as the strings in the database will be converted to a Recycleable. Otherwise, the string will return a null Recycleable and the error message will be printed out (figure 56).

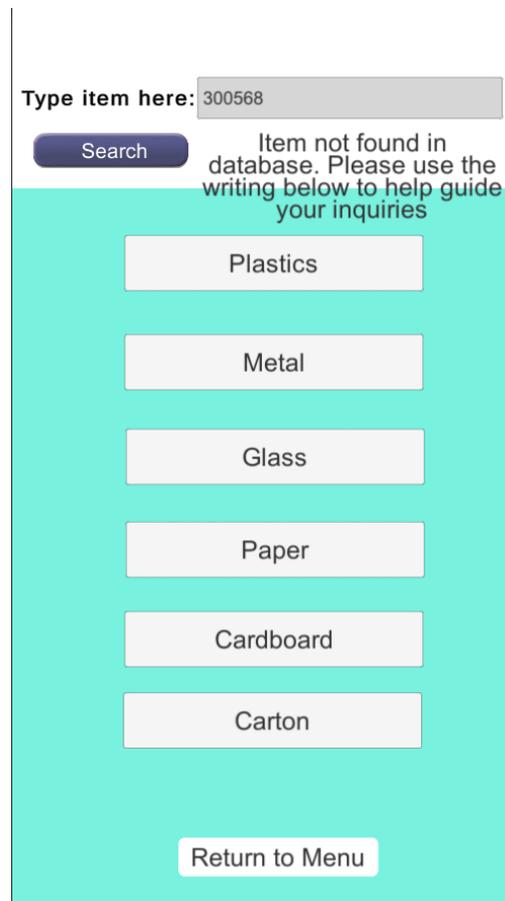


Figure 56: Error message when an incorrect item is typed into the search bar

5.2. Maintenance

As with many successful apps and websites, regular system checks, and updates are required to keep it running at top performance. Maintenance updates to fix bugs and implement patches should be done on a weekly basis with larger updates occurring every 2-4 weeks. These larger updates would include functional changes such as adding a point system or adding more items to the database to expand the number of recyclables that the app can recognize.

5.3. Support

There is an email on the website and app that can be contacted for any support or questions regarding the product. The email is chareyourecyclingteam@gmail.com and it will be checked regularly for any inquiries.

6. Product Documentation

6.1. App

This is the product documentation of the app system of the prototype

6.1.1. Bill of Materials of App

Table 3: Bill of Materials of App

Item Name	Purpose	Price
Friendly & Clean UI	Make user interface look clean and friendly	\$9
Versatile Game Sound Effects	Sound effects for recycling games	Free
Free Sound Effects Pack	General sound effects for the app	Free
Lean GUI	Make user interface look clean and friendly	Free
GitHub for Unity	Assist in coding in Unity	Free
	Total Final Price:	\$9.00

6.1.2. Equipment List

Table 4: Equipment List of App

Equipment	Use
Unity	Software used to create the app
Vuforia	AR software

6.1.3. Instructions

This section will go over how the app and website was produced in a step-by-step manner for every section.

6.1.3.1. UI

The following sections will explain in detail how to create and develop the user interface used in this app.

6.1.3.1.1. Background

The first step to perform is to make the scene show in 2D. This can be done in the scene window, at the top left, by clicking 2D which is to the right of the “Shaded” option. Since the app is developed to fit a mobile device, the build settings should be changed to support the iOS or Android platform. To do this select “File” -> “Build Settings” -> select “iOS” or “Android” under the “Platform” box -> Click “Switch Platform” (figure 57).

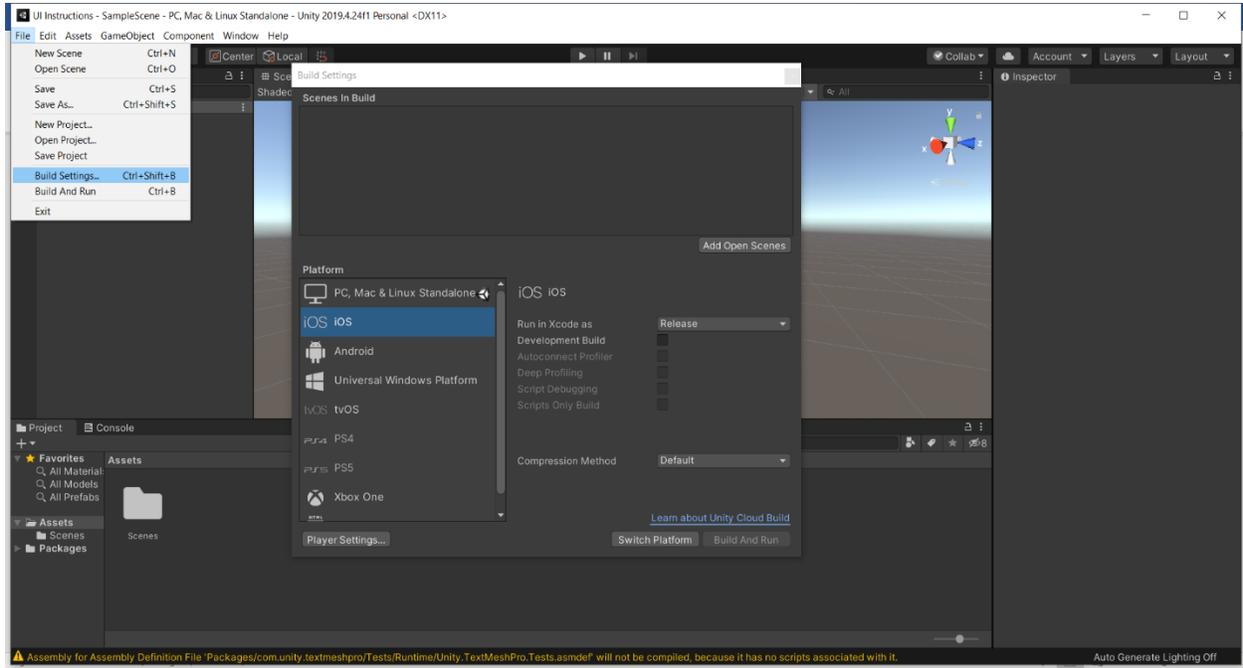


Figure 57. Switching Platforms

Next, the scene window can be viewed in the proper aspect ratio to fit a phone screen. To do this, simply click the “Game” tab -> select “Free Aspect” -> then choose “16:9 Portrait (9:16)” (figure 58).

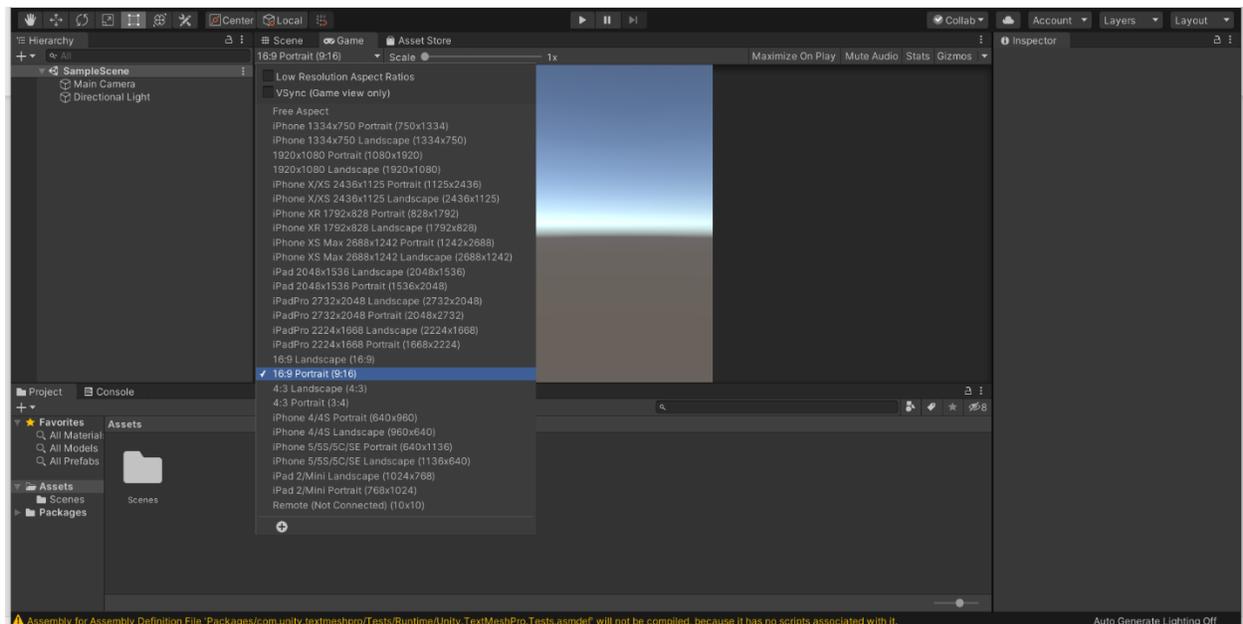


Figure 58. Changing Aspect Ratio

Now to create the UI. The background is very easy to implement as its only components are a canvas and a blank image. To add these, simply click “GameObject” -> “UI” -> “Canvas” (figure 59). Also rename it to “Background Canvas” to avoid confusion with other canvases.

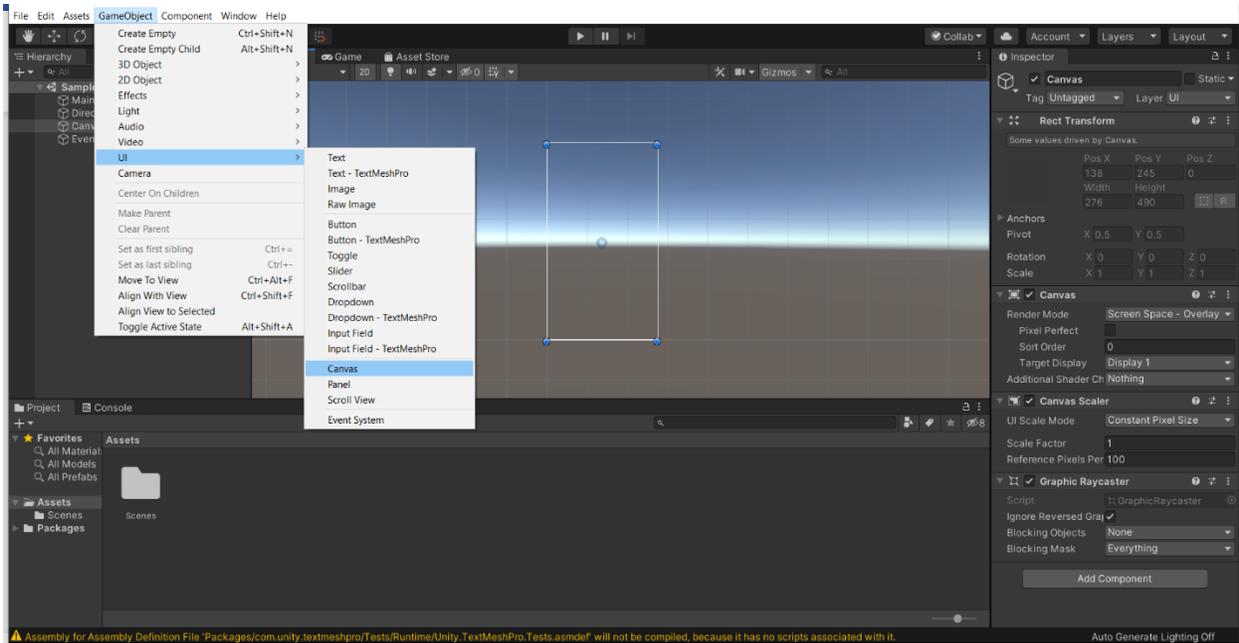


Figure 59. Inserting a Canvas

With the background canvas now created, a blank image can now be inserted to provide a colourful background. Simply select “GameObject” again -> “UI” -> “Image” (figure 60). From here, the image can be scaled using the scaling tools located at the top left of Hierarchy window.

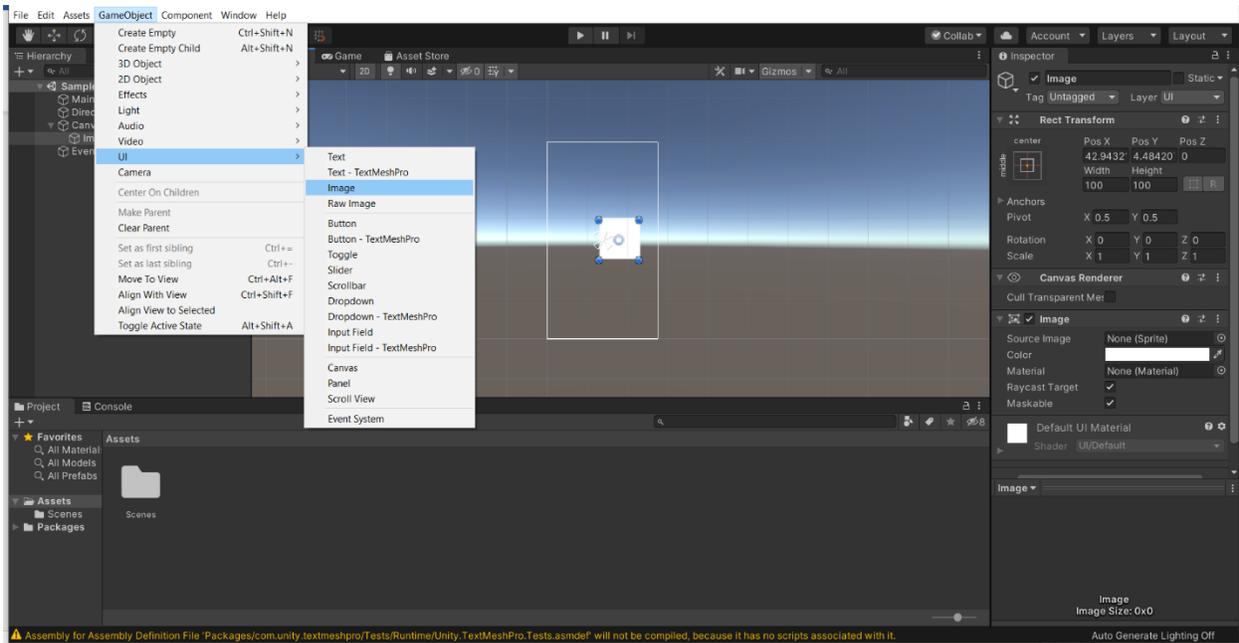


Figure 60. Inserting a Blank Image

The final touch to the background is to choose a colour. This is done by selecting the image beside the “Colour” option under the Inspector window. From there, any desired colour can be

chosen from the colour wheel and intensity spectrum or by typing in a hexadecimal code (figure 61).

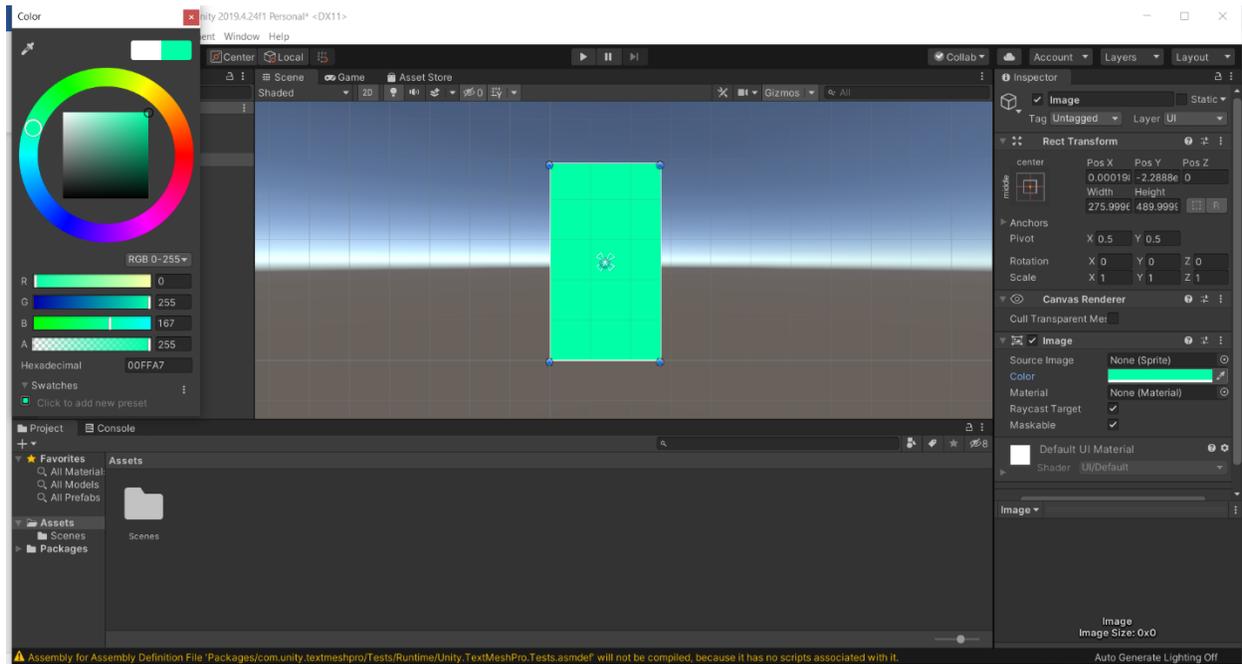


Figure 61. Changing the Colour

With that, the background canvas is complete, unless further detail is desired.

6.1.3.1.2. Button Canvas

The next step in developing the UI is inserting the buttons that allow navigation throughout the app. The chosen buttons were downloaded from an available, free GUI package on the Unity Asset Store. To access this, open the “Asset Store” tab in the scene window or type in “Unity Asset Store” on your browser (figure 62).

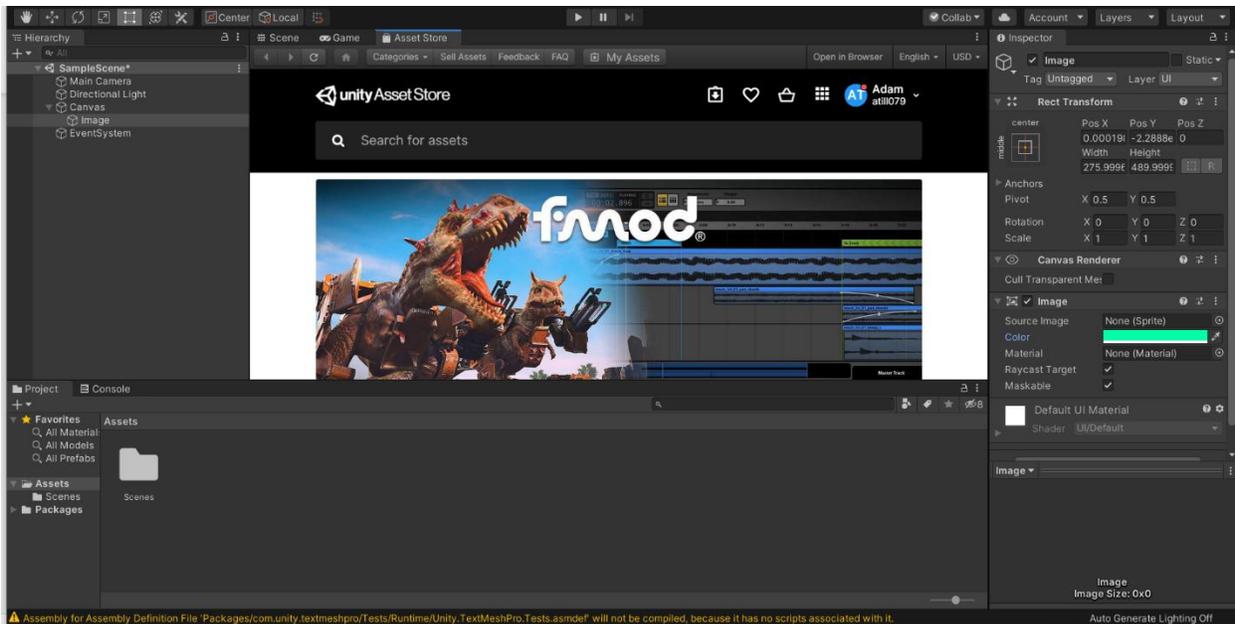


Figure 62. Opening the Asset Store

From here, any desired asset is searchable. Since the Lean GUI was used, type that into the search bar. Two versions are available, one for free and the other for \$5. Select the free option for a cost-effective option. Click the download button to receive this package then “Import” to move it to the Unity program (figure 63).

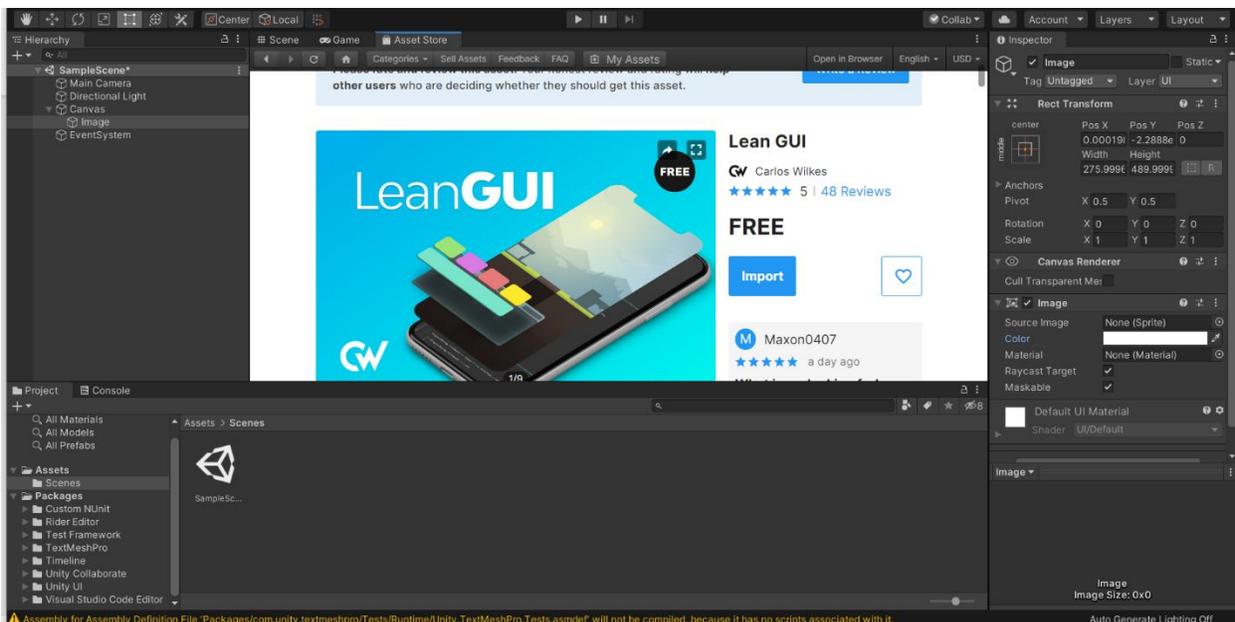


Figure 63. Downloading and Importing Lean GUI

A box will appear in your Unity screen showing all the assets imported. Click “Import” at the bottom right of the box to finish the process. The Lean assets should now be available under “Assets” in the project window (figure 64).



Figure 64. Lean GUI Assets Location

Now create another canvas for the buttons and rename it “Button Canvas”. Lean provides an example scene of pre-made buttons with integrated animations and sounds in their own scripts. To access this, select “Lean” under “Assets” -> “GUI” -> “Examples” -> drag “01 Button” onto the “Button Canvas”. This will add the pre-made scene to the recently created scene on the button canvas (figure 65).

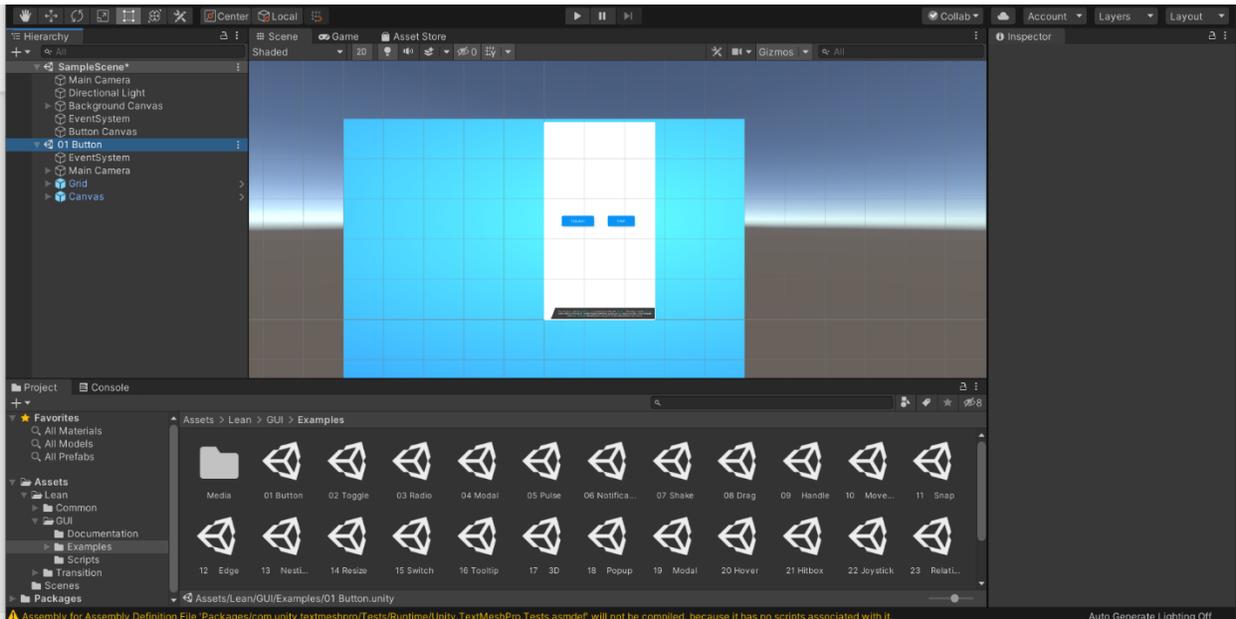


Figure 65. Adding 01 Button to Button Canvas

This pre-made scene contains some useless functions and additions now that it is attached to the “Button Canvas”. It would be worthwhile to remove the blue highlighted “Grid” and the second “Main Camera” entirely from the scene. To do this, simply right-click on the undesired asset and select “Delete” (figure 66).

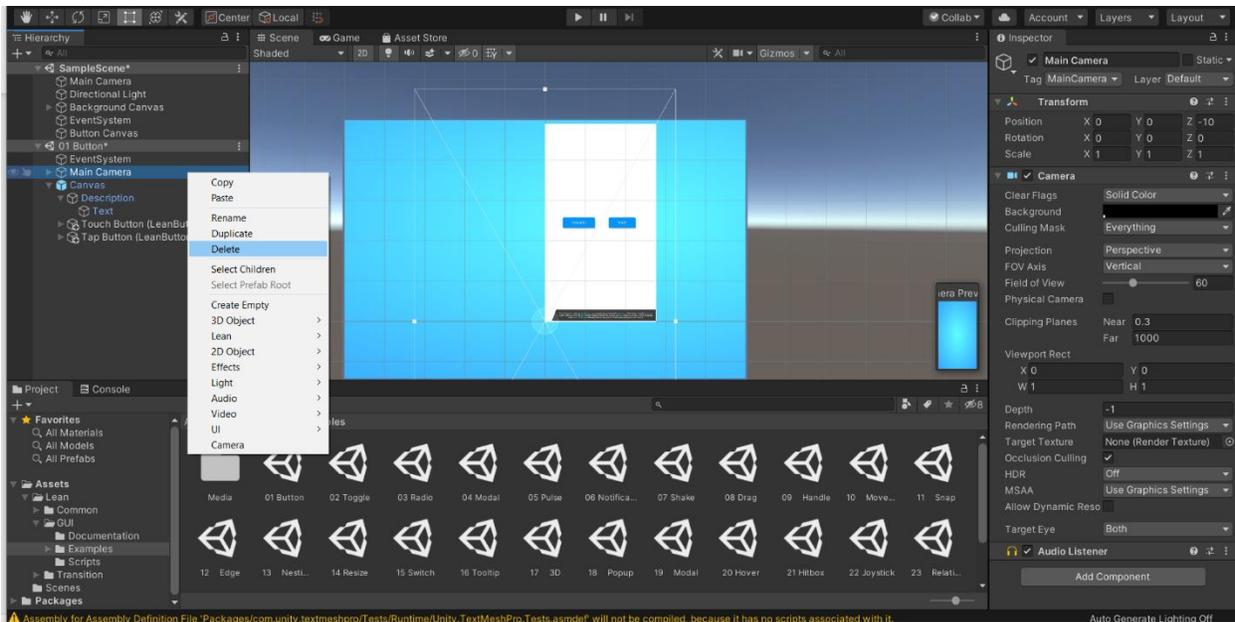


Figure 66. Removing Grid and Main Camera

One other undesirable asset is the “Description” at the bottom of the canvas. Unfortunately, when trying to delete this, a prompt appears, informing the user that it is a child of a Prefab instance and that it cannot be deleted without further steps (figure 67).

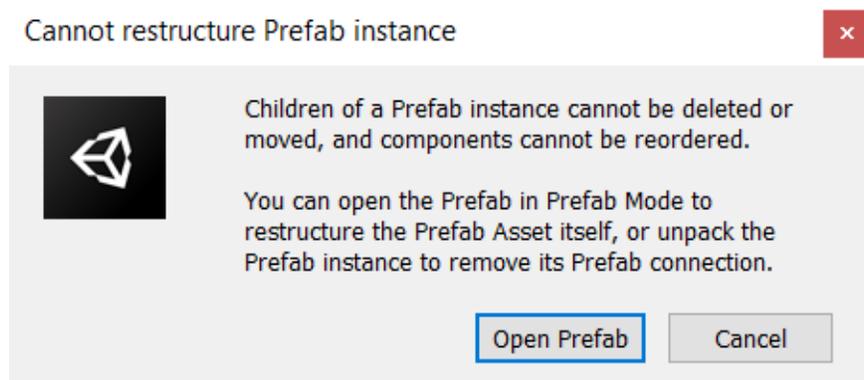


Figure 67. Description Error Prompt

To work around this, the “Description” can simply be moved outside of the canvas so that it won’t be visible on the game screen. Select the “Move Tool” at the top left of the screen and shift the description asset outside of the canvas, either up, down, or to the side (figure 68).

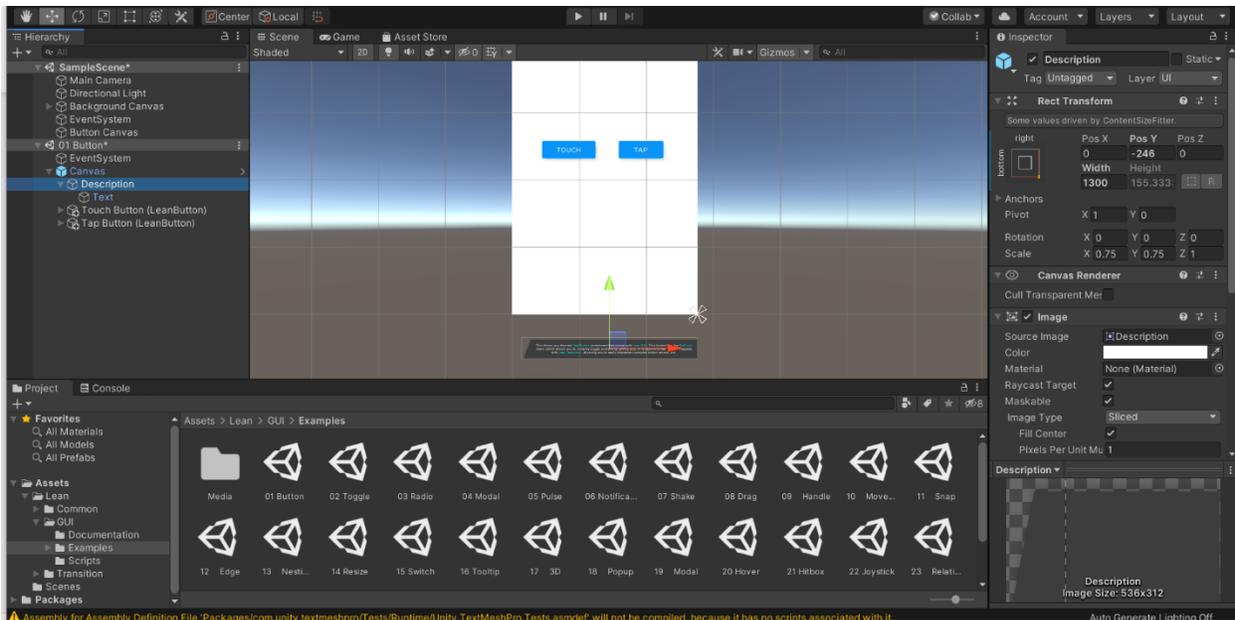


Figure 68. Moving Description out of View

Now only two buttons are left, “Touch” and “Tap”. Unfortunately, the background canvas is no longer available to view as it is blocked by the blue highlighted “Canvas”. To fix this, shift the “Touch Button” and “Tap Button” to the “Button Canvas”. From there, the problem with deleting the description and blue “Canvas” is no longer an issue. Select the blue highlighted “Canvas”, right-click, and select delete. The white background should be gone now and the “Background Canvas” should be visible (figure 69).

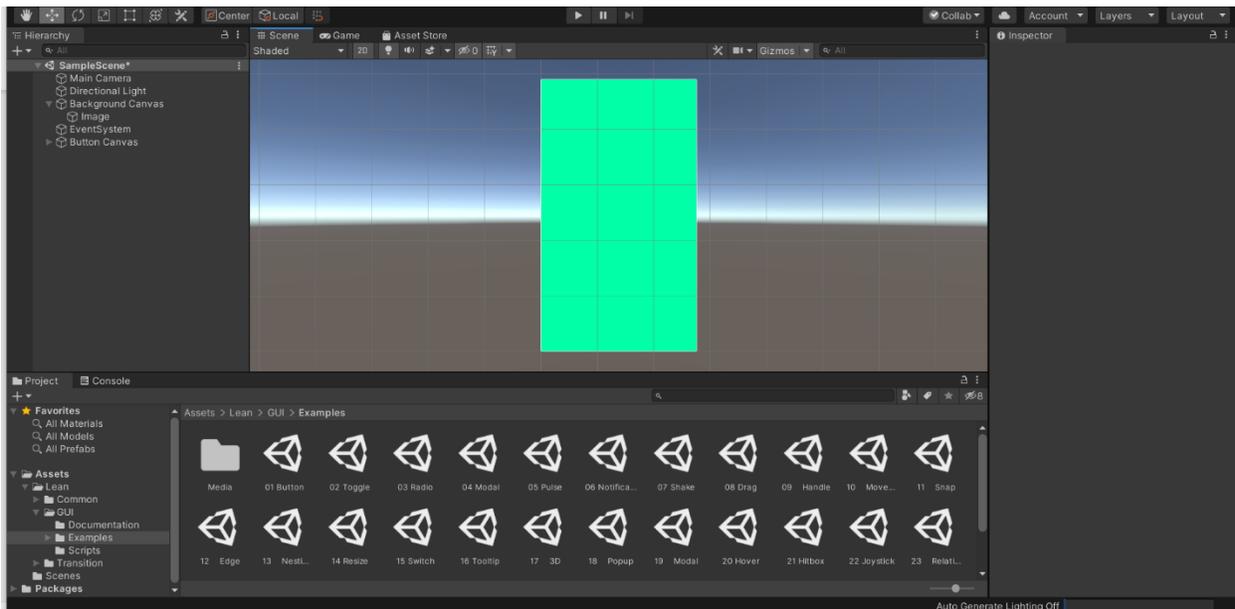


Figure 69. Deleting Blue Canvas and Description

To simplify the hierarchy window, removing the scene “01 Button” may prove to be beneficial. Unfortunately, now the buttons are no longer visible, but they do exist in the scene. To view and

interact with them, move the “Button Canvas” in the hierarchy window onto the “Background Canvas” such that it is now a part of it. Ensure that “Image” is above the buttons in hierarchy (figure 70).

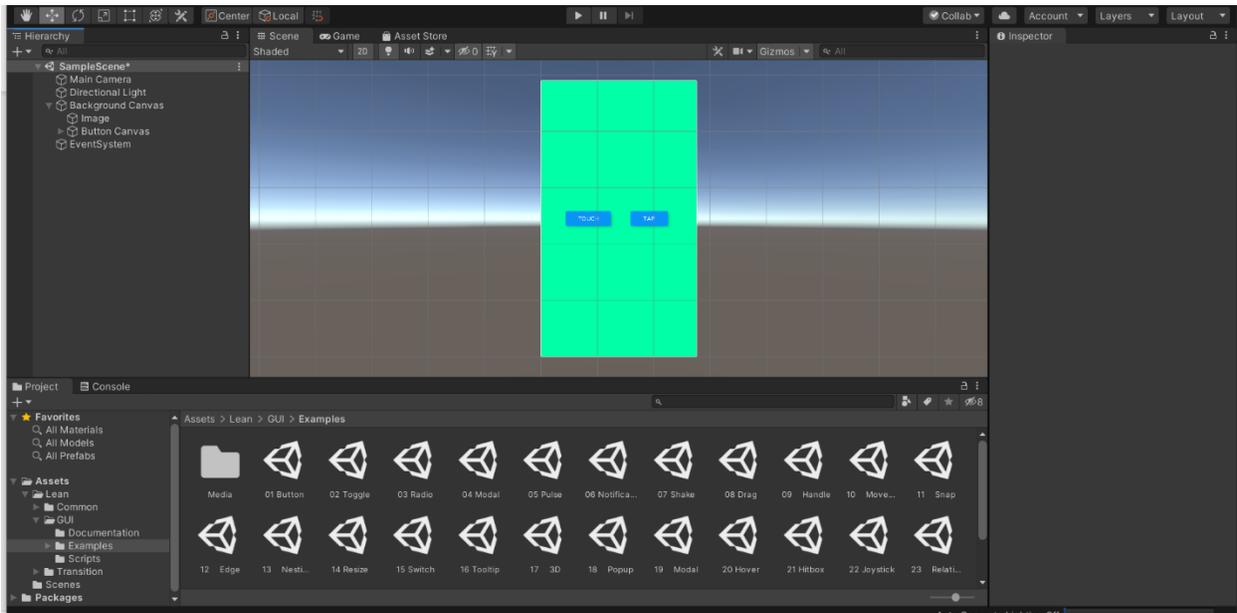


Figure 70. Moving Buttons to Foreground

The only problem now is that the buttons are too small to read or interact with. To fix this, select the “Touch Button” -> choose the “Rect Tool” at the top left -> Scale the button area to a desirable size (figure 71).

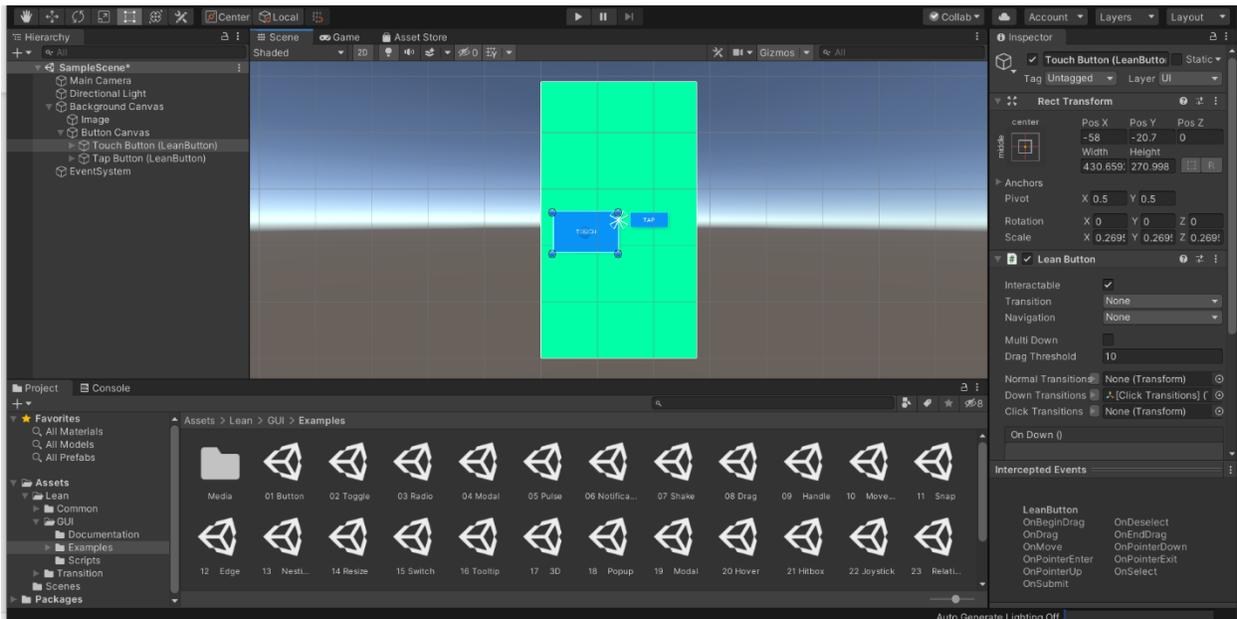


Figure 71. Rescaling the Touch Button

The next step is to resize the text to make it legible from afar. First, expand the “Touch Button” in the Hierarchy window, then expand the “Cap” asset. From there, select the “Text” asset then change the “Font Size” in the Inspector window to a reasonable size for reading (figure 72).

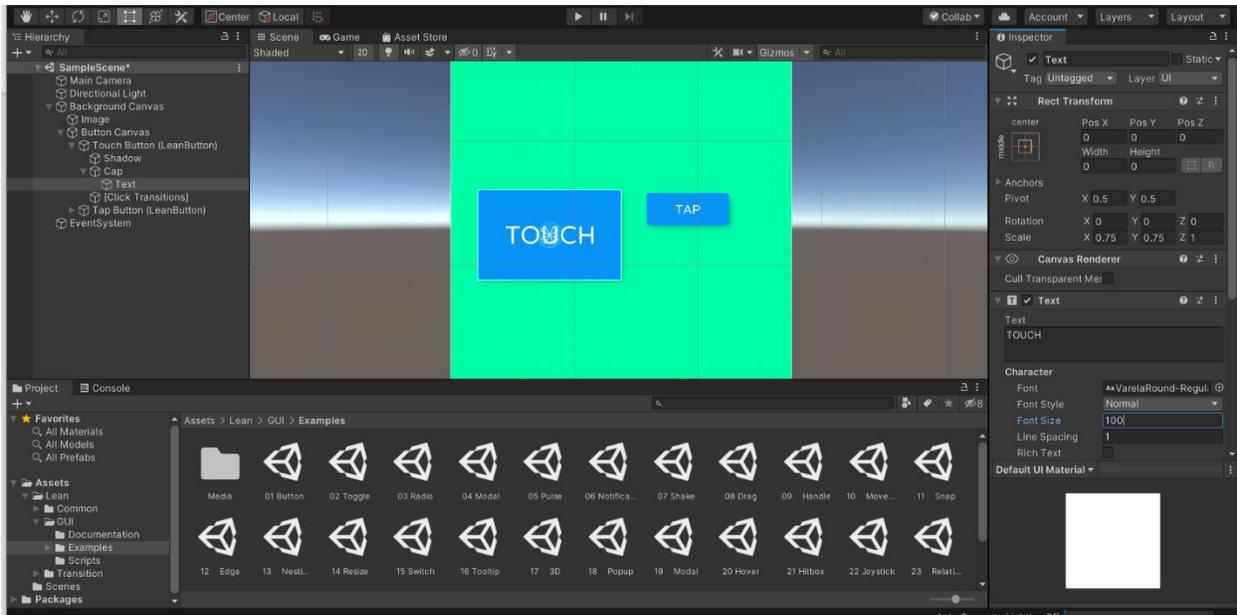


Figure 72. Changing Font Size

Repeat this process for the “Tap Button” and move the buttons to the bottom of the game screen. The app requires a third button to help with navigation so space should be left on the bottom-right side of the screen (figure 73).

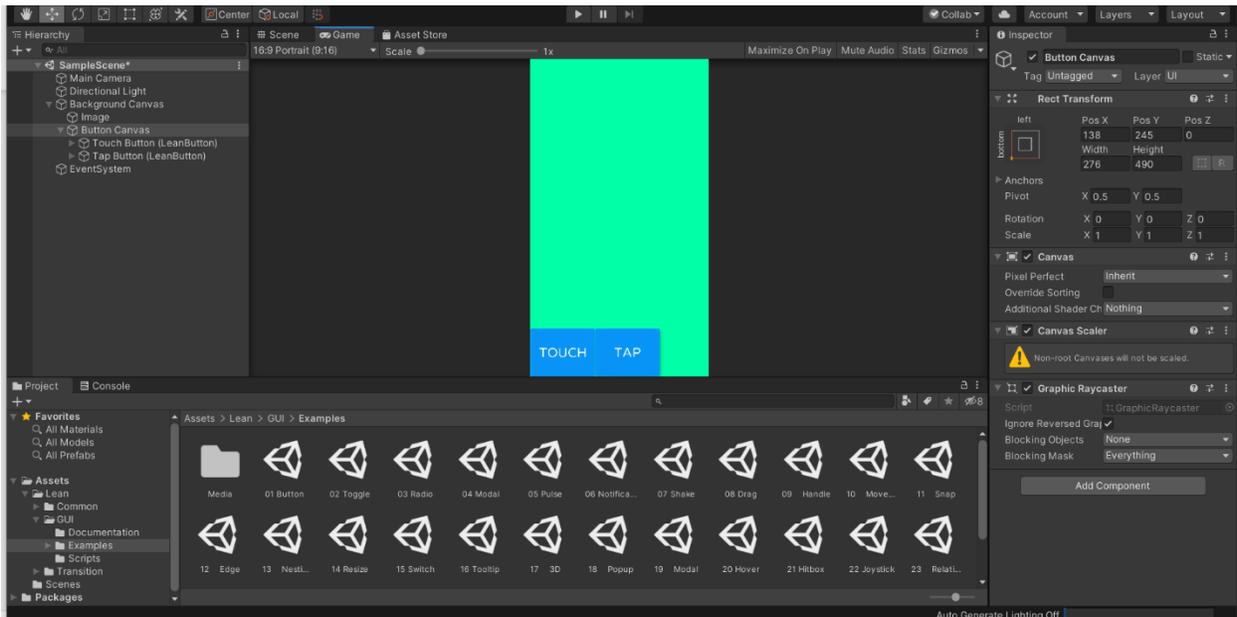


Figure 73. Formatting the Buttons

Now, to create the third and final button, select “GameObject” -> “UI” -> “Button”. This will add a new button onto the canvas in its default state (figure 74).

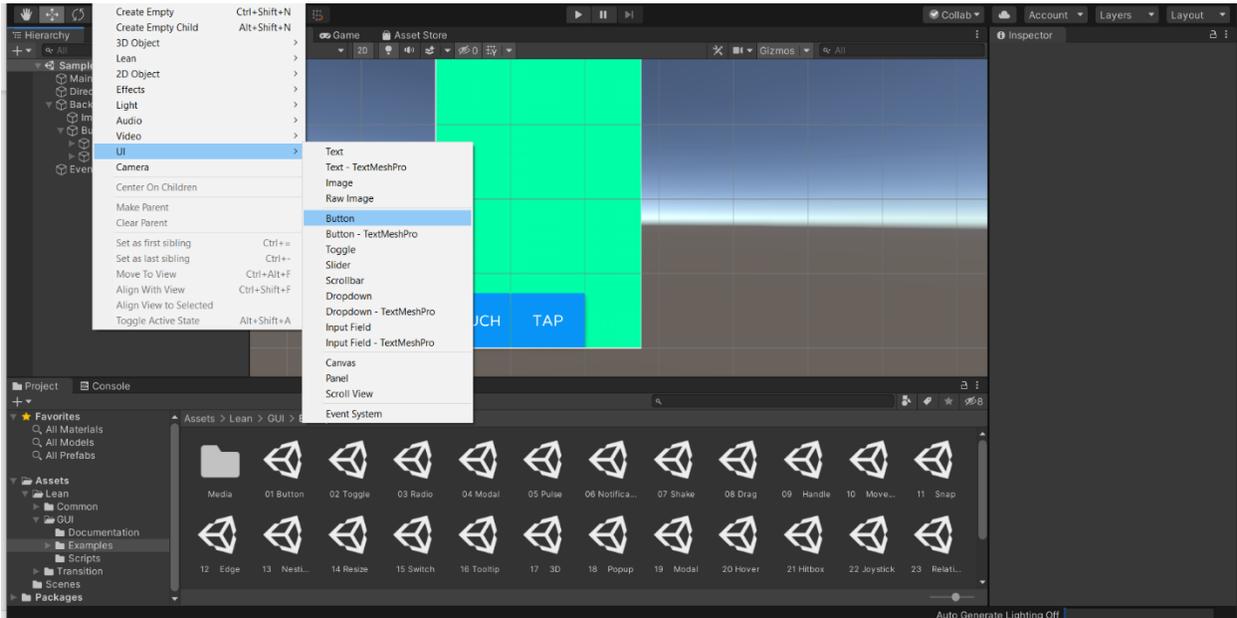


Figure 74. Inserting a New Button

The goal is to make this new button almost indistinguishable from the other two. First of all, resize the button to fit the open space with the “Rect Tool”. Then, record the colour of the other two buttons by clicking “Cap” on one of them, then clicking the blue box beside “Colour”. In this case, the hexadecimal code is “0894F7”. Save that code and insert it into the colour option on the new button. Finally, rename the button to “New” (figure 75).

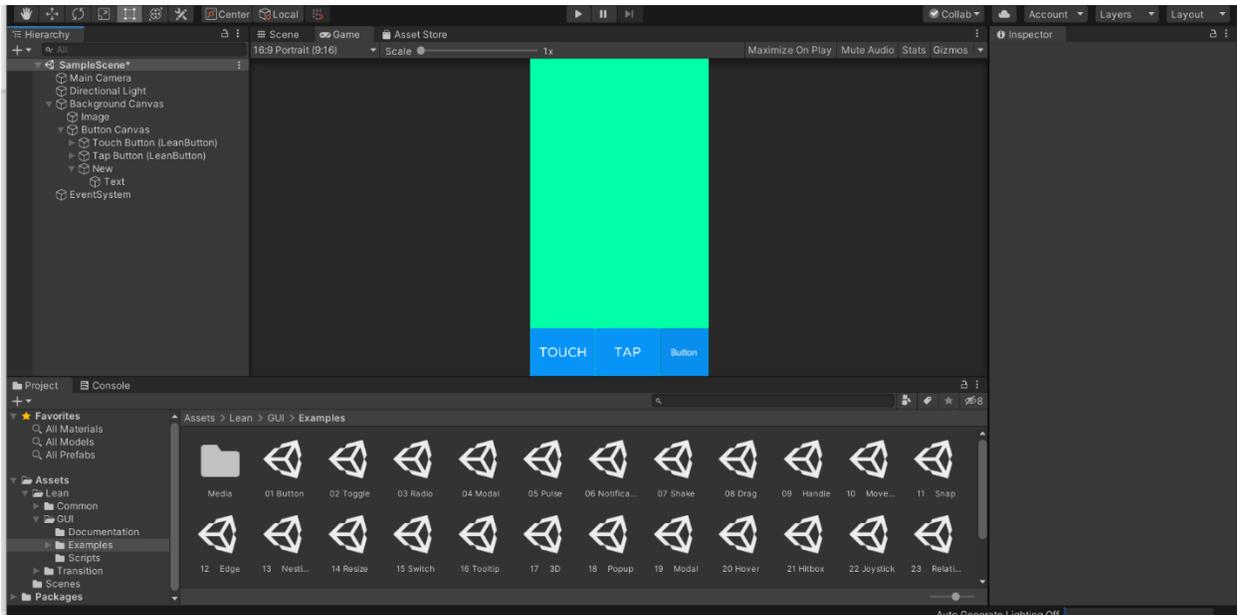


Figure 75. Modifications to New Button

Unfortunately, the text is noticeably different than in the other two. To fix that, change the font to “VarelaRound-Regular Font” by clicking the small, circular icon to the right of the corresponding box. Also, play around with the font size and other options to receive a desired style (figure 76).

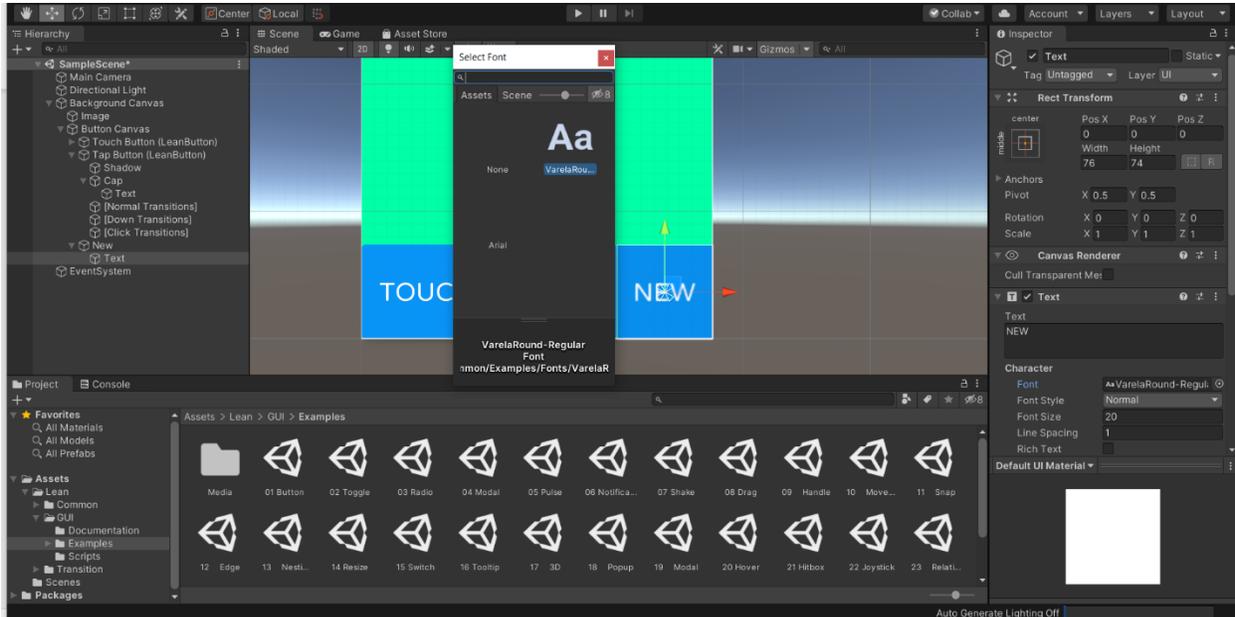


Figure 76. Changing Fonts

These are all the steps to follow to make the button UI.

6.1.3.1.3. Title

The final UI feature to insert to the title screen is the title itself. This can either be done by adding text onto a title canvas then typing the title name, or by importing a pre-made image of the title then formatting from there. Since the former is much easier, the focus will be on the latter.

First of all, make the title in an external program, either photoshop or a similar program where more detail can be easily added. From there, download the image onto the computer (figure 77). Note, since the title was made available through discord, click “Open original” to properly save the image to the computer.



Figure 77. Downloading the Title

In Unity, create a new canvas, then add a raw image by clicking “GameObject” -> “UI” -> “Raw Image” (figure 78).

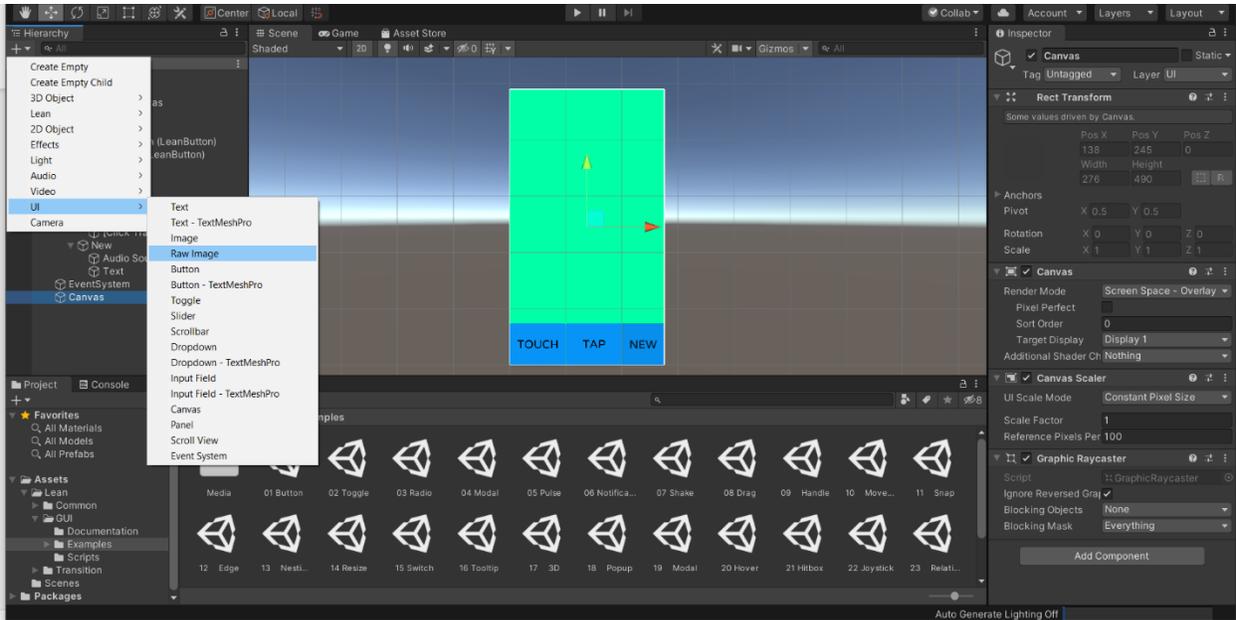


Figure 78. Adding Raw Image

Next, select “Assets” at the top left of the screen then click “Import New Asset”. This will open up the file browser to search for the desired asset. Select the recently saved title then click “Import’ (figure 79).

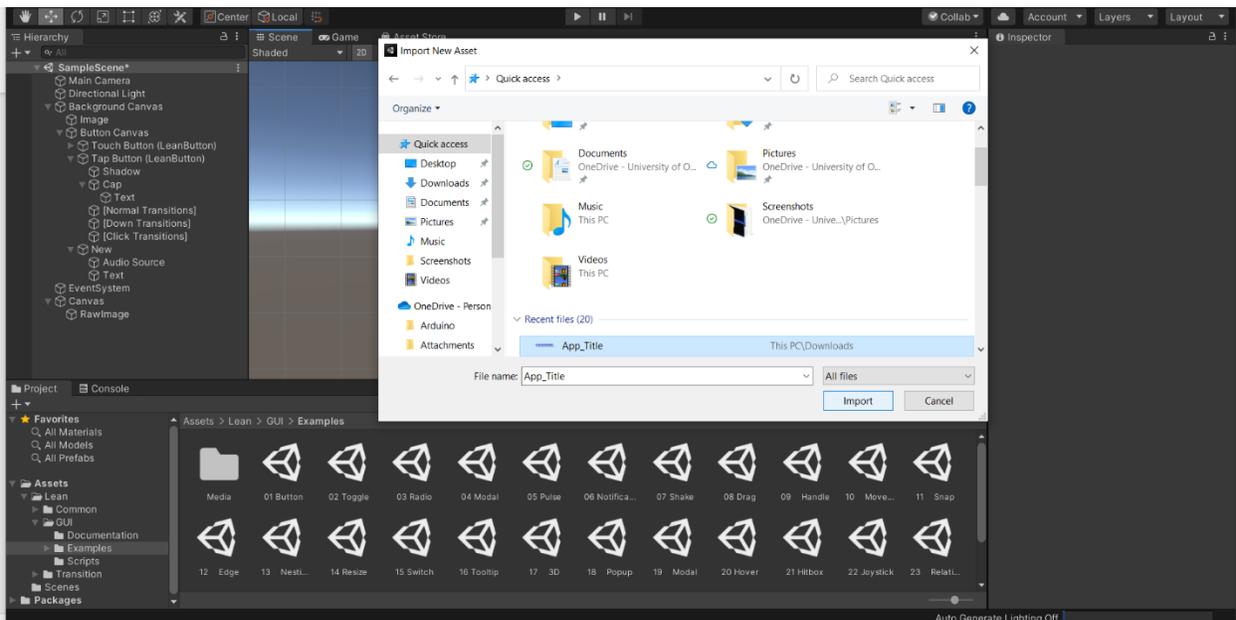


Figure 79. Importing Title Asset

The asset should appear under “Assets” in the Project window. Now click the “Raw Image”, click the small, circular icon beside “Texture” in the Inspector window, then choose the desired asset (figure 80).

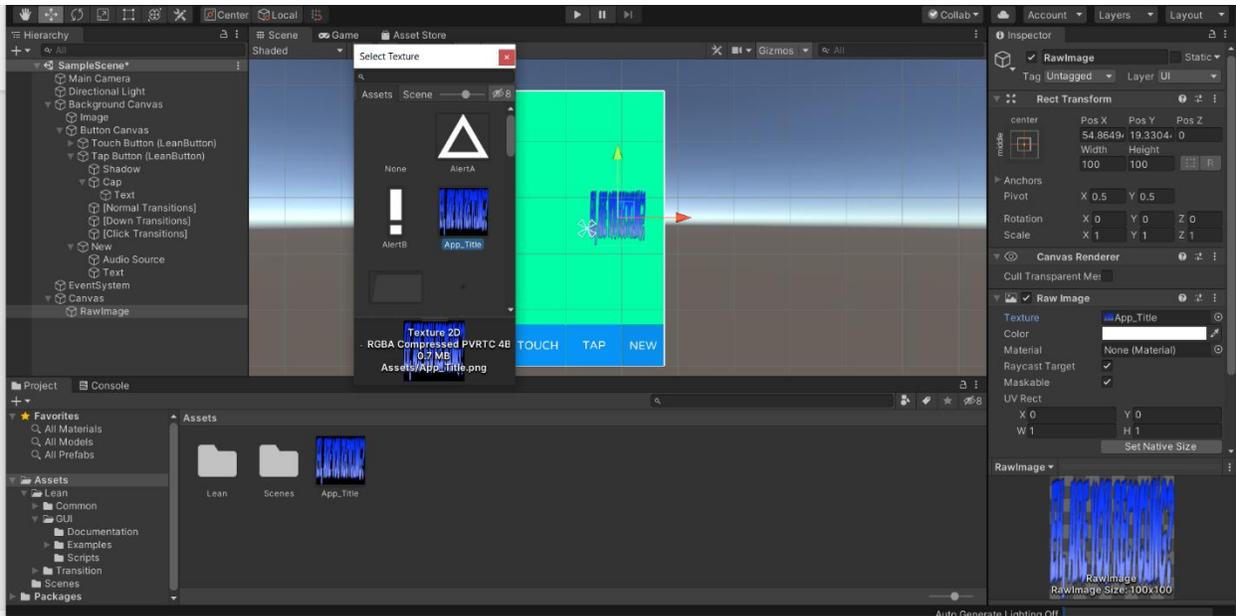


Figure 80. Attaching the Asset to the Raw Image

The final step is to resize the image to a desirable size and relocate it to a desirable position (figure 81).

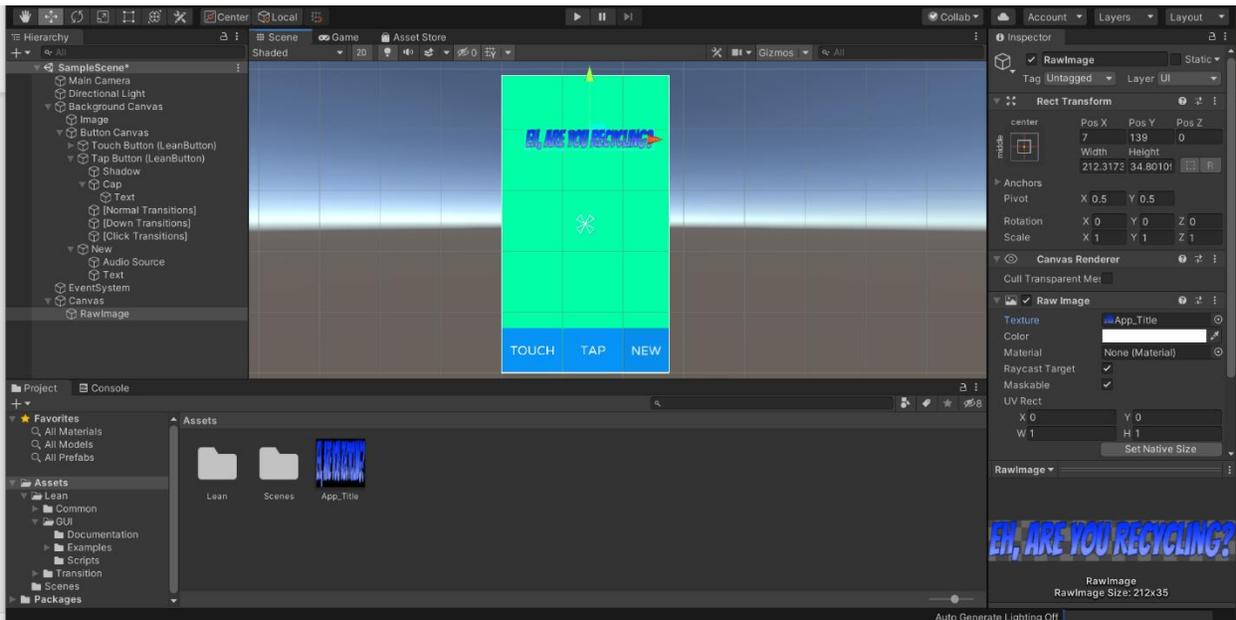


Figure 81. Final Product of Title Screen UI

This concludes the UI instructions for the title screen.

6.1.3.2. Search Bar and Manual Sort

When creating the app, make sure that all canvases are in Scale with Screen Size. Otherwise, the page will not format correctly. And ensure that in the Game mode to the top left of the Unity, the aspect ratio is on 9:16.

6.1.3.2.1. Search Bar

To construct the search bar, first open a new scene in Unity by either right clicking the project window to open a menu, scrolling the cursor to create, and then clicking scene. (figure 82) Name the scene “Manual Search Screen”.

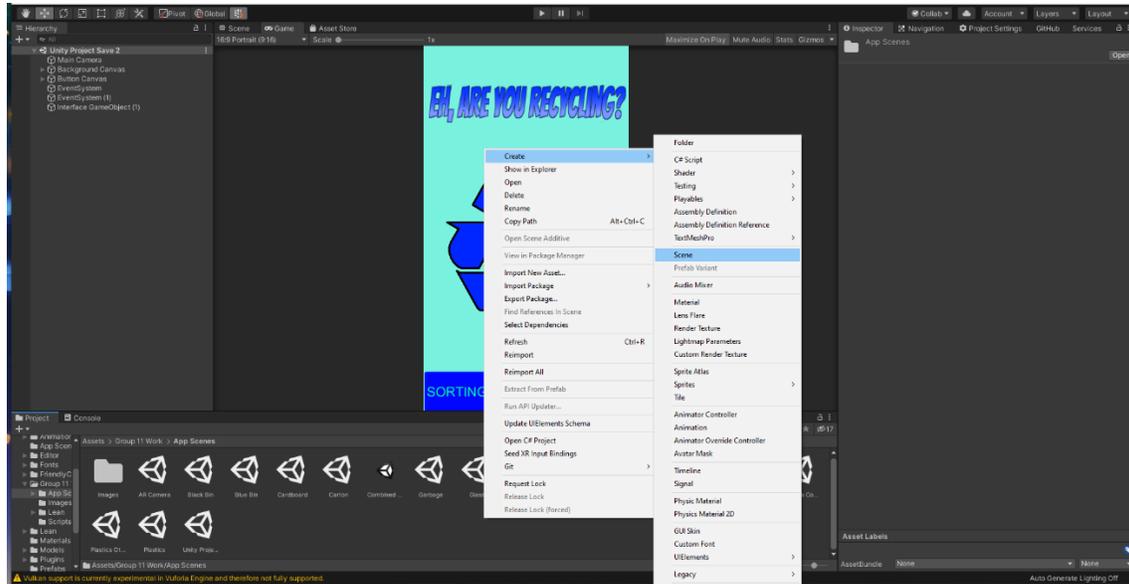


Figure 82. Creating a new Scene

Once the new scene is created, right click in the Hierarchy menu, navigate to UI and pick canvas. (figure 83)

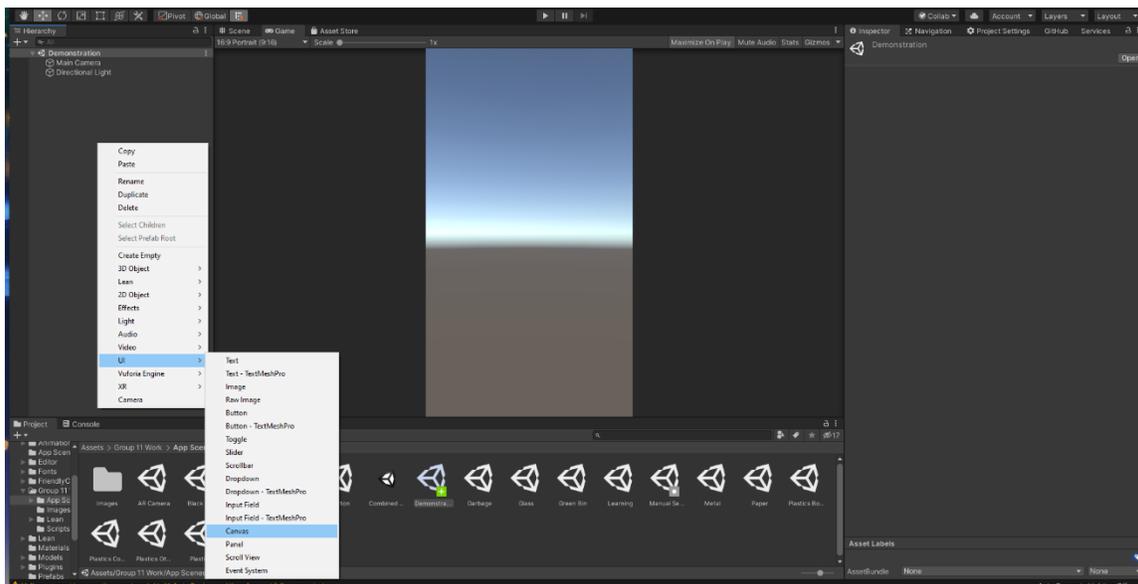


Figure 83. Adding a canvas to the scene

Once the canvas is created, right click on the canvas, click on the canvas and check the Inspector menu on the right of the screen. Look under Canvas and make sure Render Mode is Screen

Space – Overlay. Then, check Canvas Scaler and ensure that the UI Scale Mode is in Scale with Screen Size. If the settings are not correct, click on the box to open a menu window and choose the correct setting (figure 84). The name of the Canvas can also be renamed by changing the word “Canvas” at the top of the Inspector window.

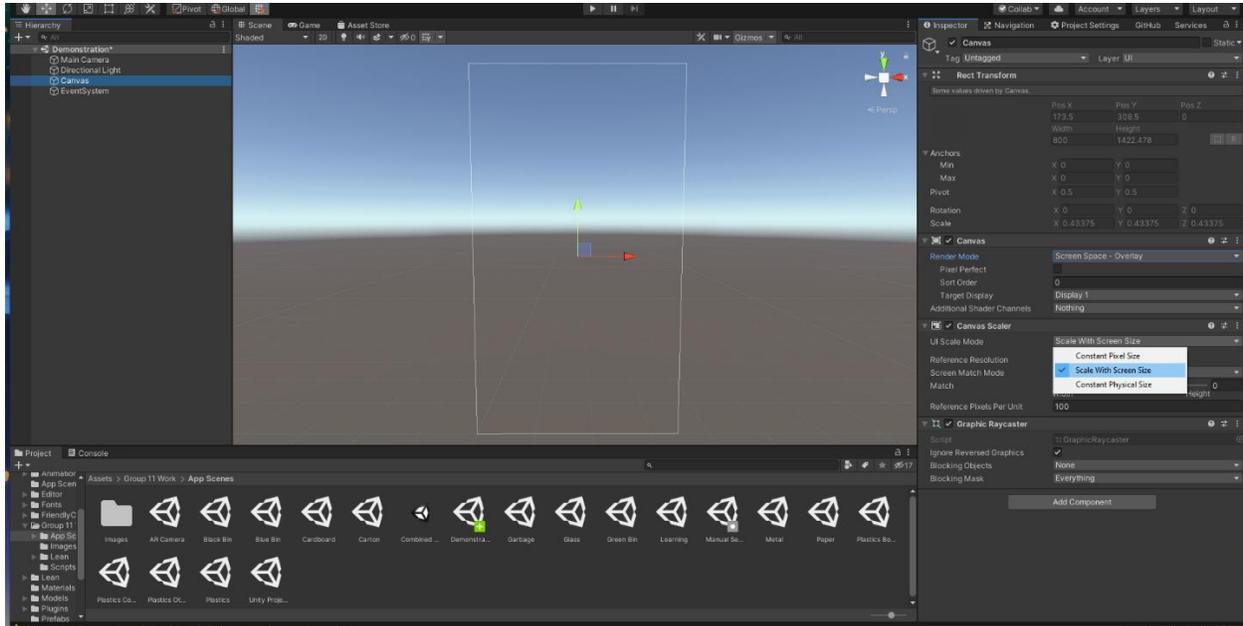


Figure 84. Change the Canvas to Scale with Screen Size

Once the Canvas has been scaled properly, right click on the Canvas name in the Hierarchy menu scroll down to UI and insert a raw image (figure 85).

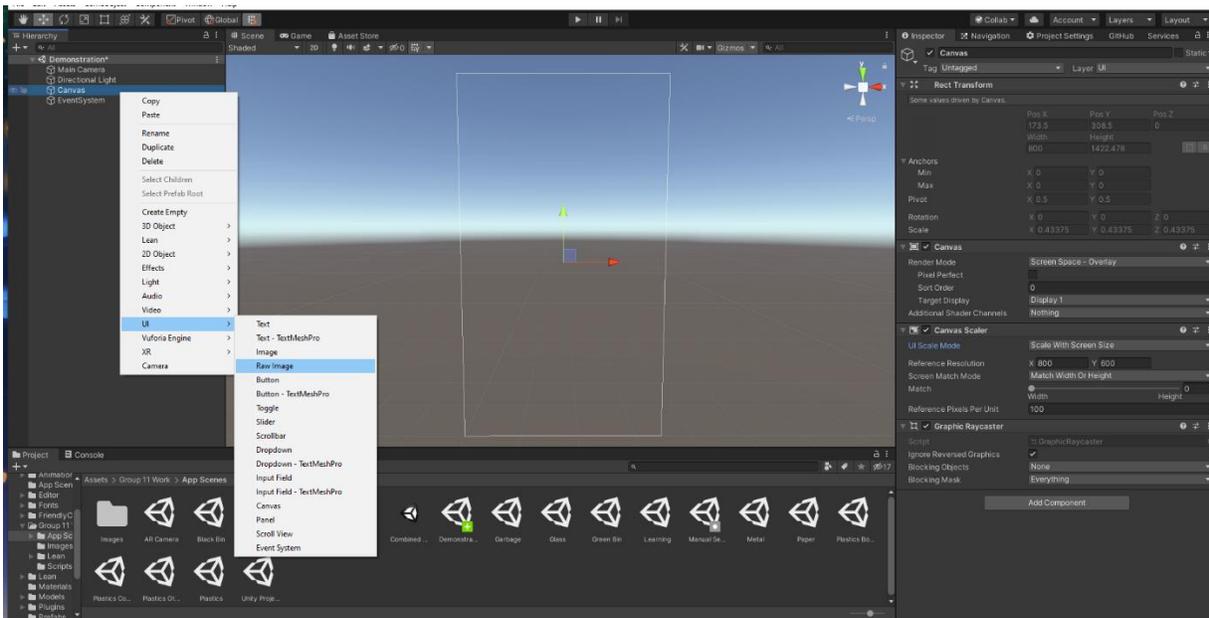


Figure 85. Adding a raw image to the canvas

Reshape the raw image so that it fully covers the canvas shape. To do this, click on the button on the top left of the Unity window in the shape of a square with a dot at each end (figure 86). Four blue dots should appear around the white image. Resize the image to fit the Canvas (the white outline) by dragging those blue dots.

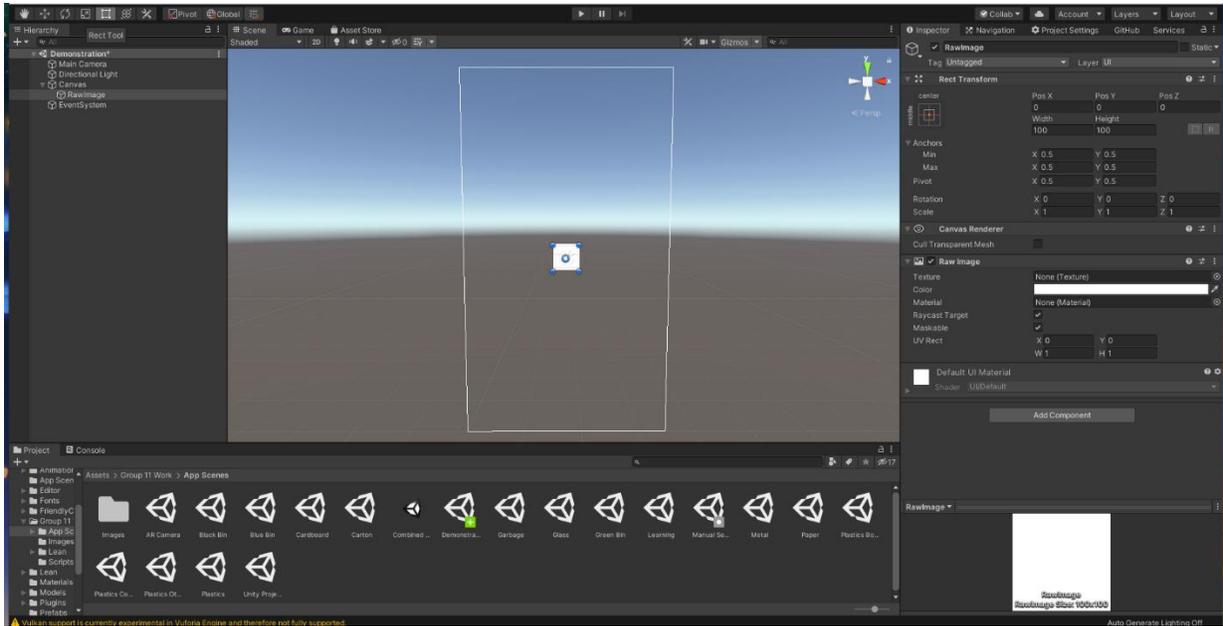


Figure 86. Using the Rect tool to resize the raw image

Now return to the Canvas in the Hierarchy, right click it and add a button from the UI menu (figure 87).

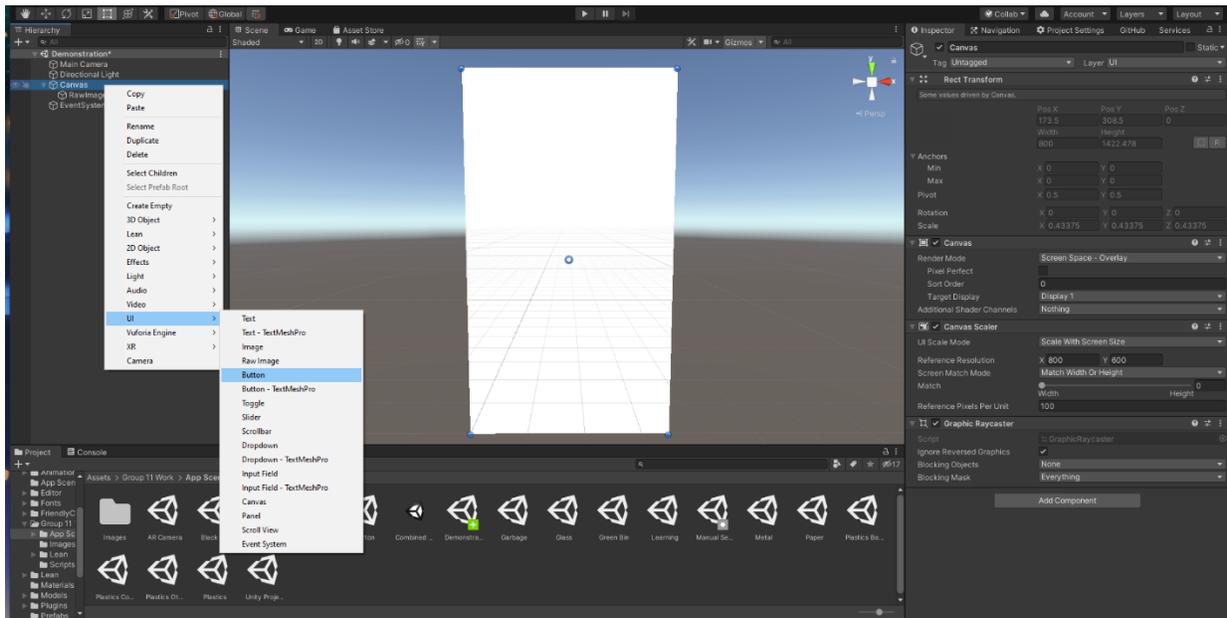


Figure 87. Adding a button to the canvas

Use the rectangle tool to resize the object and the tool with the multi-facing arrows to move the button to wherever in the canvas (figure 88). Then click on the arrow beside the Button in the Hierarchy and click on the “Text” prompt that opens below it.



Figure 88. Using the move tool to change the location of the button

In the inspector menu for the text, scroll down to the heading called “Text” and type “Search” into the box where the word “Button” is. Then scroll down to the Font Size and increase it to about 40 – or when the text becomes visible in the Scene. (figure 89)

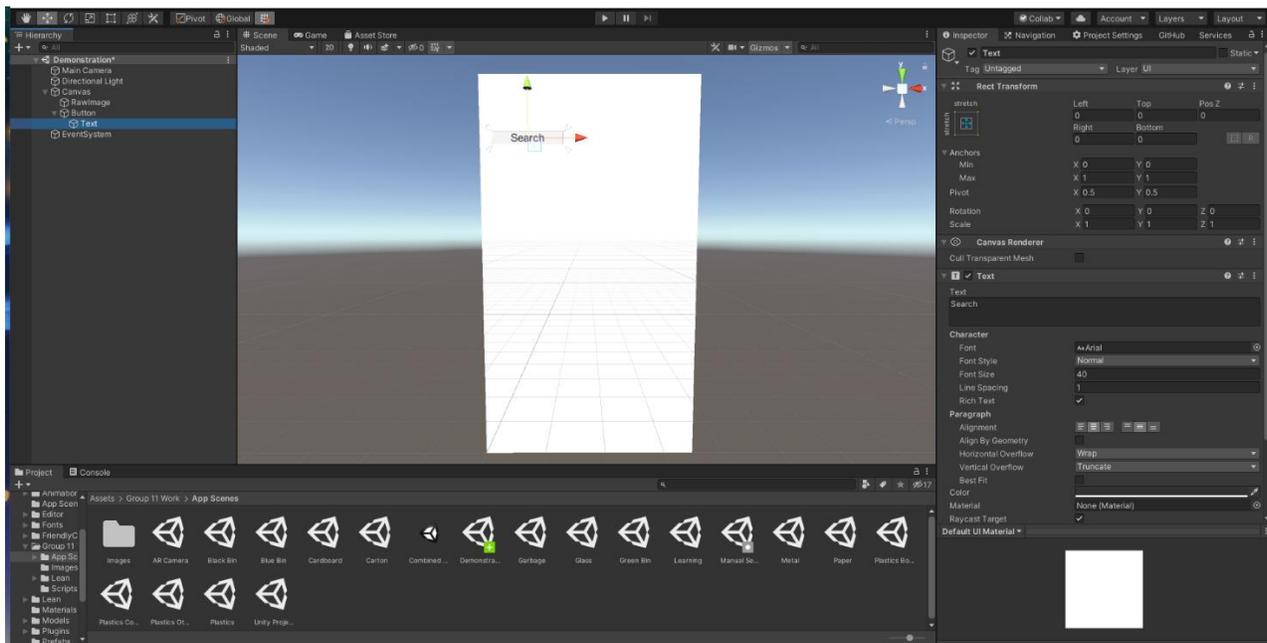


Figure 89. Changing the text and size of the text in the button

Now that the Search button has been created, right click on the Canvas name in the Hierarchy menu, scroll down to UI and insert an Input Field (figure 90).

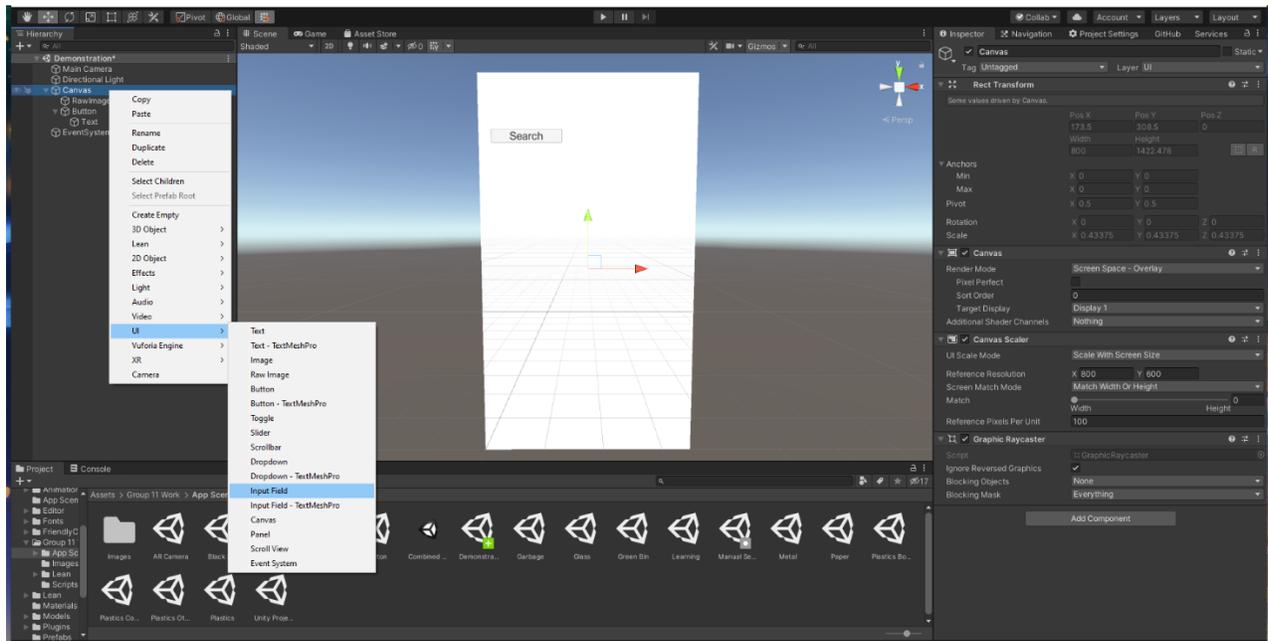


Figure 90. Adding an Input Field to the canvas

Similarly to the Button, use the rect tool to scale the search bar and the move button to move the input field beside the button. Then click on the arrow beside the inputfield in the hierarchy menu and click on Text. Change the text font size to around 40, but make sure the text fits in the scaled inputfield. The size of text that fit in the text box can be checked by clicking on the placeholder item and changing the font size of that text to the same size. (figure 91).

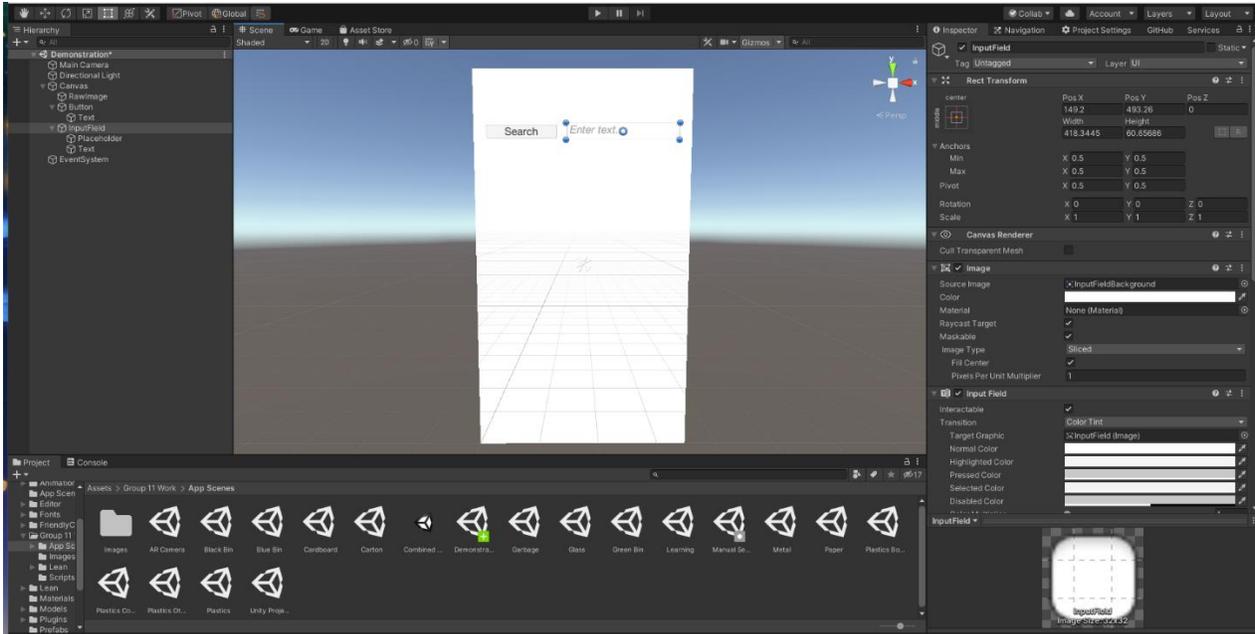


Figure 91. Changing the size of the placeholder text

Return to the hierarchy once more, right click on the canvas, scroll to UI, and input a text box. Move the text box right below the search bar and scale it to a large size. Also, click on the text to open the inspector menu and change the font size to the same used with the inputfield and button (figure 92). Ensure that the default text in the box is removed to leave a blank text box.

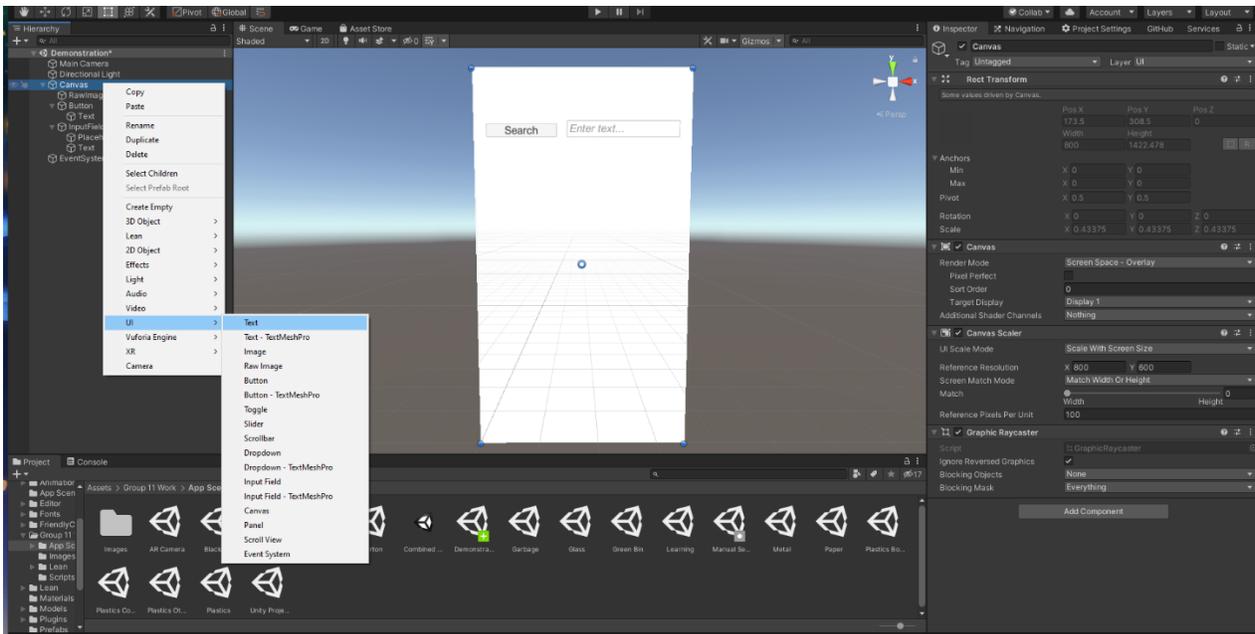


Figure 92. Adding text to the canvas

Now, return to the project menu at the bottom. Right click, move to create and click on C# script (figure 93). The script will be asked to be named. Name the script “InputToText” and then double click on the script to open it.

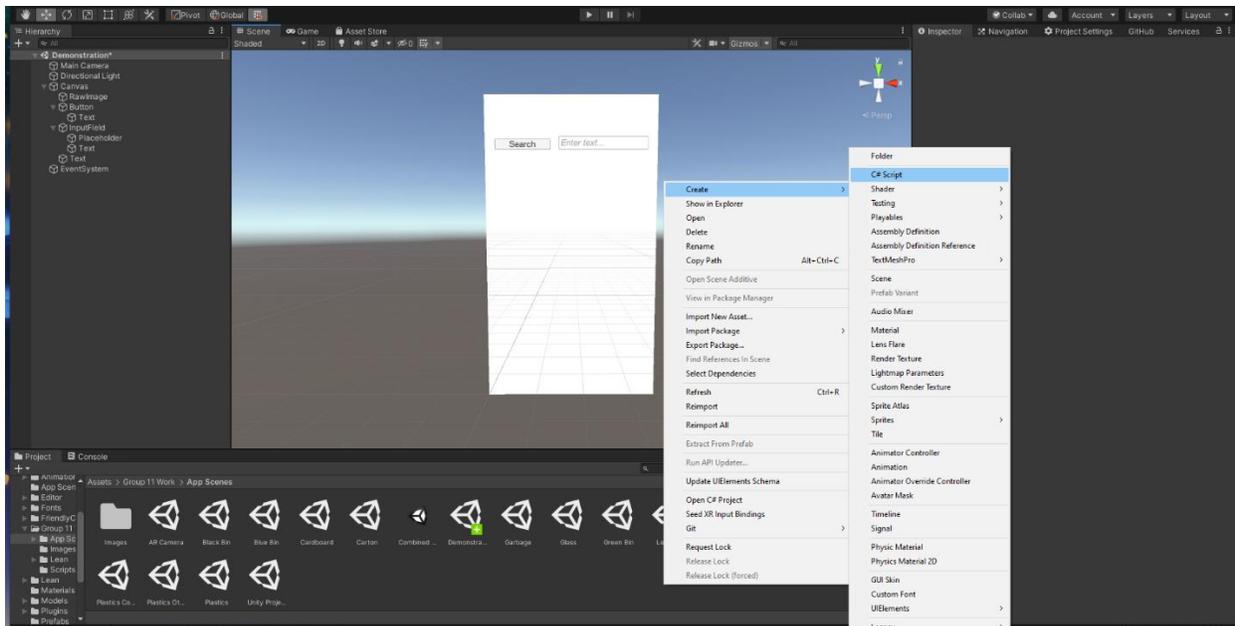


Figure 93. Adding a C# script to Unity

Now delete the automatically generated code and replace it with the following code:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.SceneManagement;

public class InputToText : MonoBehaviour
{
    public InputField SearchItem;
    public Text output;
    private List<Dropdown.OptionData> listoptions;
    private List<string> keys;

    /*
    Laura Chin
    void -> void
    creates the database when the project first starts
    */
    private void Start() {
        RecycleableDatabase.create();
        keys = RecycleableDatabase.getKeys(); //Creates a list of all keys in dat
abase
    }

    /*
```

```

Laura Chin
void -> void
retrieves the item in the search bar and prints out the output
*/
public void getText()
{
    string item = SearchItem.text;
    output.text = IsRecycleable(item);
}

/*
Laura Chin
String -> String
Takes in the name and prints out a if the item is not recycleable
If the item is recycleable, switches scenes to the item's recycling scene
*/
public static string IsRecycleable(string name)
{
    string isRecycleable = "Item not found in database. Please use the writing below to help guide your inquiries";
    string thing = name.ToLower();
    string destination;
    Recycleable item = RecycleableDatabase.Search(thing);

    if(item != null)
    {
        destination = item.GetScene();
        isRecycleable = "";
        try
        {
            SceneManager.LoadScene(destination);
        }
        catch (System.Exception)
        {
            isRecycleable = "error";
        }
    }

    return isRecycleable;
}
/*Try to make something that will list out possible choices from the database

```

```

Only begins after 2 letters are typed into the button
*/
public void guessItem()
{
    int length;
    string outPutText = "";
    string Item = SearchItem.text;
    Item.ToLower().Trim();
    length = Item.Length;
    List<string> things = new List<string>();

    if (length>1)
    {
        foreach (string i in keys)
        {
            if (i.Contains(Item))
            {
                things.Add(i);
            }
        }
    }

    if(things.Count > 0)
    {
        foreach(string i in things)
        {
            outPutText = outPutText + i + System.Environment.NewLine;
        }
    }

    output.text = outPutText;
}
}
}

```

Figure 94. InputToText code

Save the code before closing the code menu. Click on the script once more in Unity to ensure that the code has been updated. The code will probably note errors. To fix this, return to Unity and create 3 other scripts named: “Recycleable”, “RecycleableDatabase” and “RecycleableItems”. In “Recycleable” delete the generated code and paste:

```

using System.Collections;
using System.Collections.Generic;

```

```

using UnityEngine;

public class Recycleable
{
    protected string name;
    protected bool recycle;
    protected string scene;

    public Recycleable(string name, bool recycle, string scene){
        this.name = name;
        this.recycle = recycle;
        this.scene = scene;
    }

    public bool CanRecycle(){
        return recycle;
    }
    public string GetScene(){
        return scene;
    }
}

```

Figure 95. Recycleable code

In “RecycleableDatabase” delete the generated code and paste:

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public static class RecycleableDatabase
{
    public static Hashtable Database = new Hashtable(); //creates the database

    public static void create() {
        Database = RecycleableItems.build();
    }

    /*
    Laura Chin
    string -> Recycleable
    Returns the item associated with the inputed name or returns null if the
    item is not in the database
    */
    public static Recycleable Search(string key)

```

```

    {
        //Database = RecycleableItems.build();
        try
        {
            Recycleable item = (Recycleable) Database[key];
            return item;
        }
        catch
        {
            return null;
        }
    }

    public static List<string> getKeys()
    {
        ICollection stepOne = Database.Keys;
        List<string> keys = new List<string>();
        foreach(var i in stepOne)
        {
            keys.Add(i.ToString());
        }
        return keys;
    }
}

```

Figure 96. RecycleableDatabase code

Finally, for “RecycleableItems” replace the generated code with:

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public static class RecycleableItems
{
    public static Hashtable build()
    {
        Hashtable Database = new Hashtable();

        Recycleable AerosolAirFreshener = new Recycleable("aerosol air freshener"
, true, "Bllue Bin");
        Database.Add("aerosol air freshener", AerosolAirFreshener);
    }
}

```

```
    Recycleable AerosolCan = new Recycleable("aerosol can", true, "Blue Bin");
;
    Database.Add("aerosol can", AerosolCan);
    Recycleable AluminumCan = new Recycleable ("aluminum can", true, "Blue Bin");
;
    Database.Add("aluminum can", AluminumCan);
    Recycleable AluminumContainer = new Recycleable("aluminum container", true, "Blue Bin");
;
    Database.Add("aluminum container", AluminumContainer);
    Recycleable aluminumfoil = new Recycleable("aluminum foil", true, "Blue Bin");
;
    Database.Add("aluminum foil", aluminumfoil);
    Recycleable bakeryPlastics = new Recycleable("bakery plastics", true, "Blue Bin");
;
    Database.Add("bakery plastics", bakeryPlastics);
    Recycleable BeerbottleCaps = new Recycleable("beer bottle cap", true, "Blue Bin");
;
    Database.Add("beer bottle cap", BeerbottleCaps);
    Recycleable clearplasticcontainers = new Recycleable("clear plastic container", true, "Blue Bin");
;
    Database.Add("clear plastic container", clearplasticcontainers);
    Recycleable cookietin = new Recycleable("cookie tin", true, "Blue Bin");
;
    Database.Add("cookie tin", cookietin);
    Recycleable deodorantspraybottles = new Recycleable("deodorant spray bottle", true, "Blue Bin");
;
    Database.Add("deodorant spray bottle", deodorantspraybottles);
    Recycleable drinkbox = new Recycleable("drink box", true, "Blue Bin");
;
    Database.Add("drink box", drinkbox);
    Recycleable frozenjuicecan = new Recycleable("frozen juice can", true, "Blue Bin");
;
    Database.Add("frozen juice can", frozenjuicecan);
    Recycleable gabletopcontainer = new Recycleable("gable top container", true, "Blue Bin");
;
    Database.Add("gable top container", gabletopcontainer);
    Recycleable glassbottle = new Recycleable("glass bottle", true, "Blue Bin");
;
    Database.Add("glass bottle", glassbottle);
    Recycleable hairspray = new Recycleable("hairspray can", true, "Blue Bin");
;
    Database.Add("hairspray container", hairspray);
    Recycleable icecreamcontainer = new Recycleable("ice cream container", true, "Blue Bin");
;
    Database.Add("ice cream container", icecreamcontainer);
    Recycleable juicebox = new Recycleable("juice box", true, "Blue Bin");
;
    Database.Add("juice box", juicebox);
```

```
        Recycleable LCBObag = new Recycleable("lcbo bag (paper)", true, "Blue Bin");
    Database.Add("lcbo bag (plastic)", LCBObag);
    Recycleable margarinetubs = new Recycleable("margarine tube", true, "Blue Bin");
    Database.Add("margarine tub", margarinetubs);
    Recycleable metalbottlecap = new Recycleable("metal bottle cap", true, "Blue Bin");
    Database.Add("metal bottle cap", metalbottlecap);
    Recycleable metalcanlid = new Recycleable("metal can lid", true, "Blue Bin");
    Database.Add("metal can lid", metalcanlid);
    Recycleable metalcan = new Recycleable("metal can", true, "Blue Bin");
    Database.Add("metal can", metalcan);
    Recycleable metalpaintcan = new Recycleable("metal paint can", true, "Blue Bin");
    Database.Add("metal paint can", metalpaintcan);
    Recycleable milk = new Recycleable("milk carton", true, "Blue Bin");
    Database.Add("milk carton", milk);
    Recycleable milkbox = new Recycleable("milk box", true, "Blue Bin");
    Database.Add("milk box", milkbox);
    Recycleable plasticjar = new Recycleable("plastic jar", true, "Blue Bin");
    Database.Add("plastic jar", plasticjar);
    Recycleable plasticjug = new Recycleable("plastic jug", true, "Blue Bin");
    Database.Add("plastic jug", plasticjug);
    Recycleable plasticcap = new Recycleable("plastic cap", true, "Blue Bin");
    Database.Add("plastic cap", plasticcap);
    Recycleable plasticbottlecap = new Recycleable("plastic bottle cap", true, "Blue Bin");
    Database.Add("plastic bottle cap", plasticbottlecap);
    Recycleable plasticbottle = new Recycleable("plastic bottle", true, "Blue Bin");
    Database.Add("plastic bottle", plasticbottle);
    Recycleable plasticbucket = new Recycleable("plastic bucket", true, "Blue Bin");
    Database.Add("plastic bucket", plasticbucket);
    Recycleable plasticfoodtrays = new Recycleable("plastic food tray", true, "Blue Bin");
    Database.Add("plastic food tray", plasticfoodtrays);
    Recycleable plasticpail = new Recycleable("plastic pail", true, "Blue Bin");
    Database.Add("plastic pail", plasticpail);
```

```
        Recycleable plasticplantingtray = new Recycleable ("plastic planting tray", true, "Blue Bin");
        Database.Add("plastic planting tray", plasticplantingtray);
        Recycleable plastictuplid = new Recycleable("plastic tub lid", true, "Blue Bin");
        Database.Add("plastic chip tube", plastictuplid);
        Recycleable potatochiptub = new Recycleable("potato chip tub", true, "Blue Bin");
        Database.Add("potato chip tube", potatochiptub);
        Recycleable yogurt = new Recycleable("yogurt container", true, "Blue Bin");
        Database.Add("yogurt container", yogurt);
        Recycleable can = new Recycleable("pop can", true, "Blue Bin");
        Database.Add("pop can", can);
        Recycleable sourcreamtub = new Recycleable("sour cream tub", true, "Blue Bin");
        Database.Add("sour cream tub", sourcreamtub);
        Recycleable soupbox = new Recycleable("soup box", true, "Blue Bin");
        Database.Add("soup box", soupbox);
        Recycleable steelcan = new Recycleable("steel can", true, "Blue Bin");
        Database.Add("steel can", steelcan);
        Recycleable takeoutclamshell = new Recycleable("take out clam shell", true, "Blue Bin");
        Database.Add("take out clam shell", takeoutclamshell);
        Recycleable takeoutcontainer = new Recycleable("take out container", true, "Blue Bin");
        Database.Add("take out container", takeoutcontainer);
        Recycleable takeoutclearplasticcup = new Recycleable("take out clear plastic cup", true, "Blue Bin");
        Database.Add("take out clear plastic cup", takeoutclearplasticcup);
        Recycleable tetrapaks = new Recycleable("tetra pak", true, "Blue Bin");
        Database.Add("tetra pak", tetrapaks);
        Recycleable tincan = new Recycleable("tin can", true, "Blue Bin");
        Database.Add("tin can", tincan);
        Recycleable whippingcream = new Recycleable("whipping cream bottle", true, "Blue Bin");
        Database.Add("whipping cream bottle", whippingcream);
        Recycleable winebottlecap = new Recycleable("wine bottle cap", true, "Blue Bin");
        Database.Add("wine bottle cap", winebottlecap);
        Recycleable JarLid = new Recycleable("jar lid", true, "Blue Bin");
        Database.Add("jar lid", JarLid);
        Recycleable paintcan = new Recycleable("paint can", true, "Blue Bin");
        Database.Add("paint can", paintcan);
```

```
Recycleable boxboard = new Recycleable("boxboard", true, "Black Bin");
Database.Add("boxboard", boxboard);
Recycleable cardboard = new Recycleable("cardboard", true, "Black Bin");
Database.Add("cardboard", cardboard);
Recycleable catalogues = new Recycleable("catalogue", true, "Black Bin");
Database.Add("catalogue", catalogues);
Recycleable flyer = new Recycleable("flyer", true, "Black Bin");
Database.Add("flyer", flyer);
Recycleable folder = new Recycleable("folder", true, "Black Bin");
Database.Add("folder", folder);
Recycleable tissuepaper = new Recycleable ("tissue paper", true, "Black B
in");
Database.Add("tissue paper", tissuepaper);
Recycleable giftwrappingpaper = new Recycleable("gift wrapping paper", tr
ue, "Black Bin");
Database.Add("gift wrapping paper", giftwrappingpaper);
Recycleable magazine = new Recycleable("magazine", true, "Black Bin");
Database.Add("magazine", magazine);
Recycleable newspaper = new Recycleable("newspaper", true, "Black Bin");
Database.Add("newspaper", newspaper);
Recycleable pizzaboxclean = new Recycleable("pizza box (clean)", true, "B
lack Bin");
Database.Add("pizza box (clean)", pizzaboxclean);
Recycleable shreddedpaper = new Recycleable("shredded paper", true, "Blac
k Bin");
Database.Add("shredded paper", shreddedpaper);
Recycleable telephonebook = new Recycleable("telephone book", true, "Blac
k Bin");
Database.Add("telephone book", telephonebook);
Recycleable tissuebox = new Recycleable("tissue box", true, "Black Bin");
Database.Add("tissue box", tissuebox);
Recycleable book = new Recycleable("book", true, "Black Bin");
Database.Add("book", book);
Recycleable paper = new Recycleable("paper", true, "Black Bin");
Database.Add("paper", paper);
Recycleable cerealbox = new Recycleable("cereal box", false, "Garbage");
Database.Add("cereal box", cerealbox);

Recycleable coffeecup = new Recycleable("coffee cup", true, "Green Bin");
Database.Add("coffee cup", coffeecup);
```

```

        Recycleable papertowel = new Recycleable("paper towel", true, "Green Bin"
);
        Database.Add("paper towel", papertowel);

        Recycleable cerealboxliner = new Recycleable("cereal box liners", false,
"Garbage");
        Database.Add("cereal box liners", cerealboxliner);
        Recycleable candywrappings = new Recycleable("candy wrapping", false, "Ga
rbage");
        Database.Add("candy wrapper", candywrappings);
        Recycleable ceramicplate = new Recycleable("ceramic plate", false, "Garba
ge");
        Database.Add("ceramic plate", ceramicplate);
        Recycleable windowpane = new Recycleable("window pane", false, "Garbage")
;
        Database.Add("window pane", windowpane);
        Recycleable styrofoam = new Recycleable("styrofoam", false, "Garbage");
        Database.Add("styrofoam", styrofoam);
        Recycleable plasticcups = new Recycleable("plastic cup", false, "Garbage"
);
        Database.Add("plastic cup", plasticcups);
        Recycleable plastictoy = new Recycleable("plastic toy", false, "Garbage")
;
        Database.Add("plastic toy", plastictoy);

        return Database;
    }
}

```

Figure 97. RecycleableItem code

Return to Unity after saving each of the scripts. Click on the Hierarchy, off the Canvas, and click, "Create Empty (figure 98)"

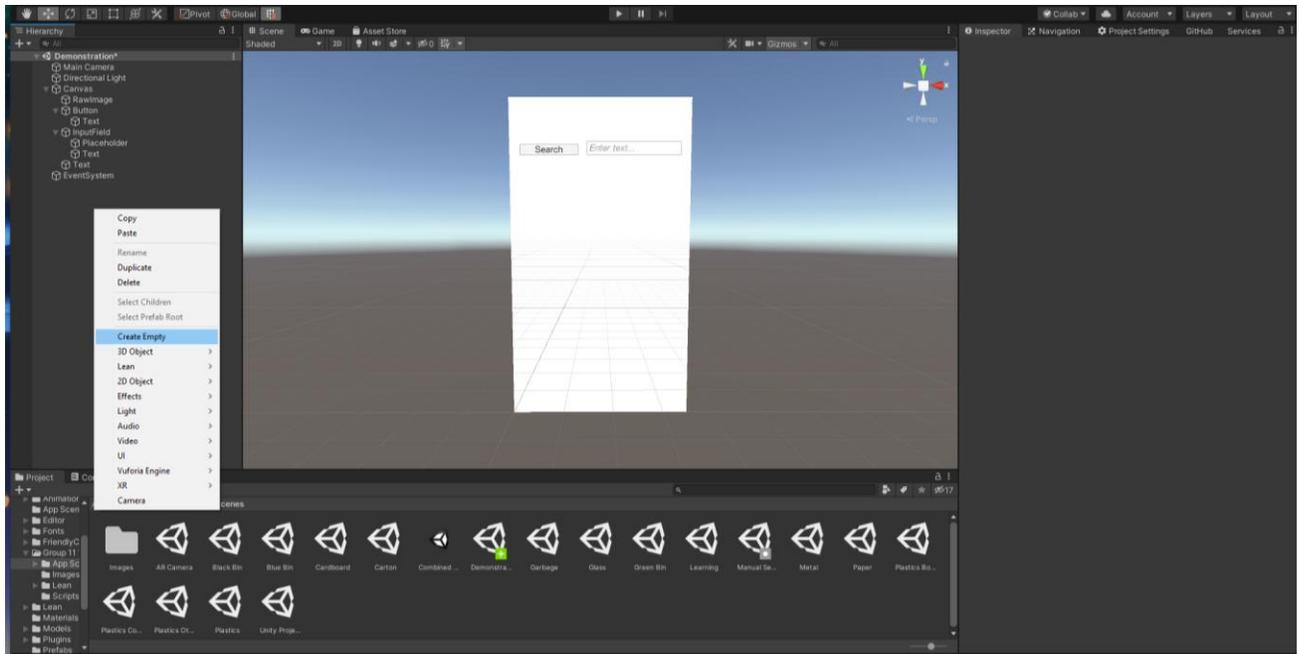


Figure 98. Create and Empty GameObject

Now drag and drop the “InputToText” script onto the “GameObject” that the “Create Empty” generated. The game object should now show in the inspector window the script attached to it along with two boxes below the name of the script: “Search Item” and “Output”. Drag the InputField to the Search Item empty box and drag the text to the Output option (figure 99).

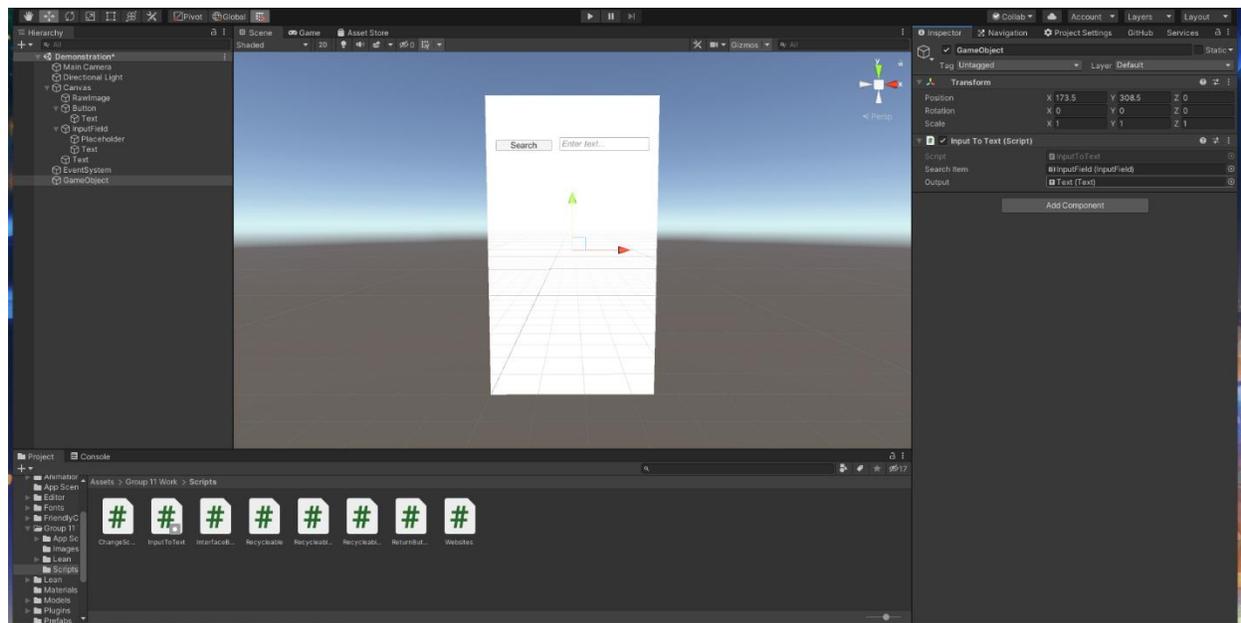


Figure 99. Dragging and dropping the “InputToText” script to the empty GameObject

Now click on the “Button” again and open the Button’s inspector window. Scroll down until a box titled “On Click ()” is found and press the plus arrow (+) at the bottom right of the box (figure 100).

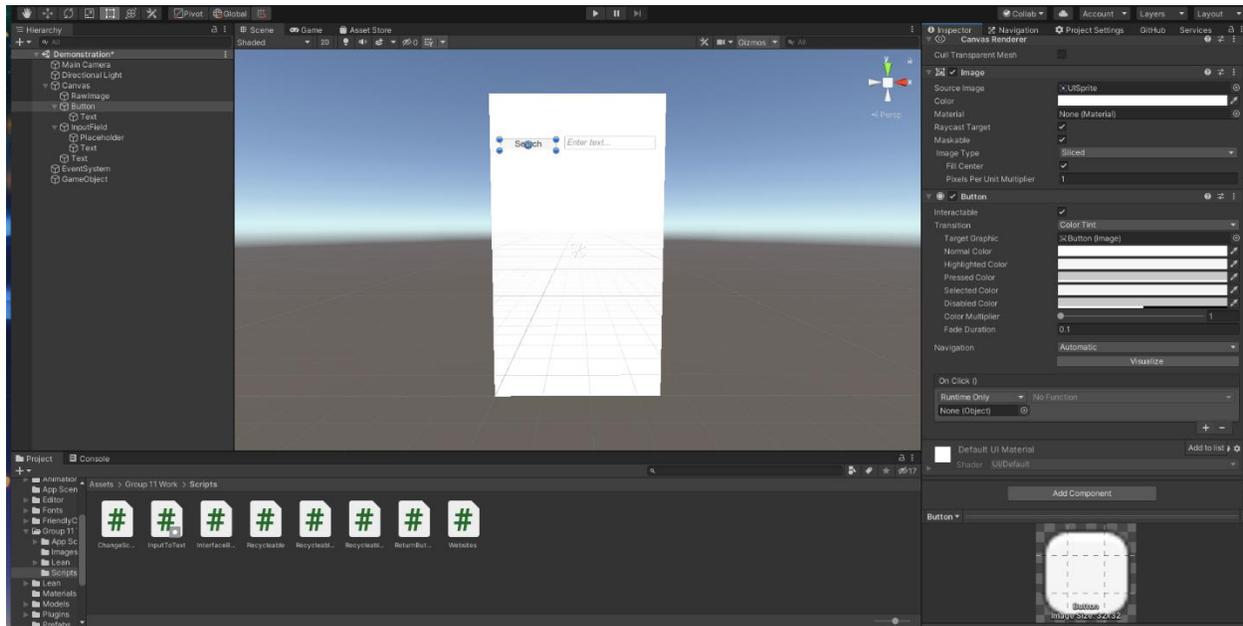


Figure 100. Adding a Runtime Function

When the plus button is clicked, a “Runtime Only” item should appear with a bar below it that says “None (Object)”. Drag the GameObject to the box. This should light up the bar that says, “No Function”. Press on the “No Function” to open a drop-down menu. Hold the cursor over “InputToText” and scroll down to find “getText”. Press on this to add it to the Button (figure 101).

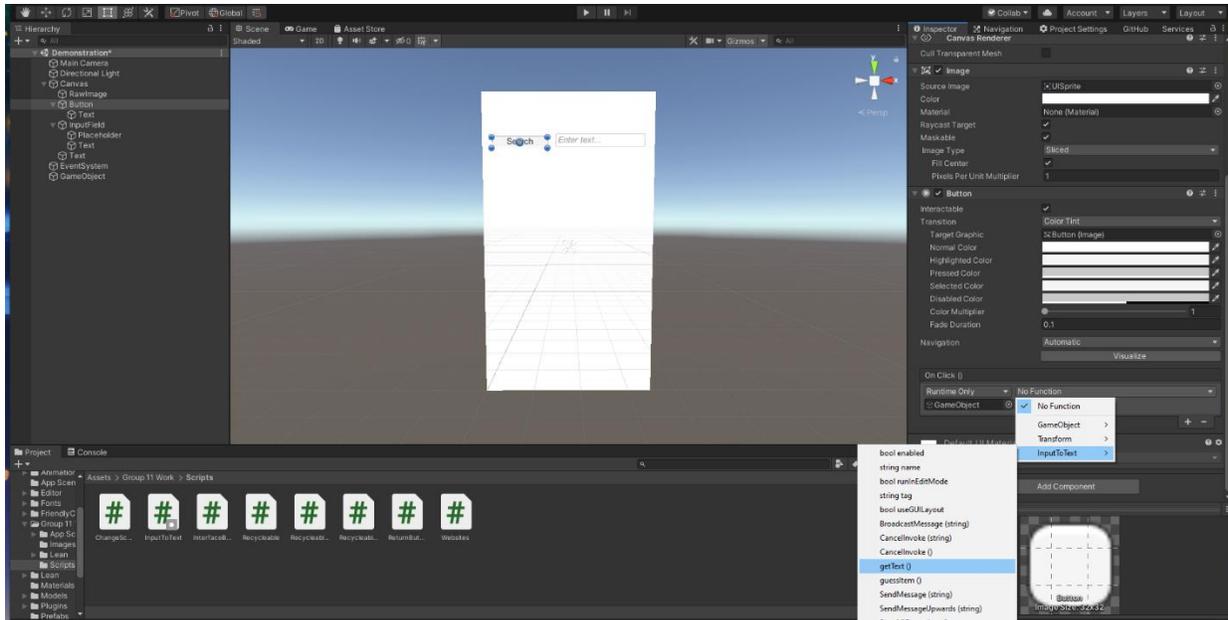


Figure 101. Picking the correct Function for the Runtime Function added to the button

Then click on InputField in the Hierarchy menu and scroll down until the box “On Value Changed (String)” is found. Click on the plus sign (+) and drag the GameObject to the opened box same as with the button. Then, in the “No Function” box, open the dropdown menu, scroll to “InputToText” and press “guessItem” (figure 102).

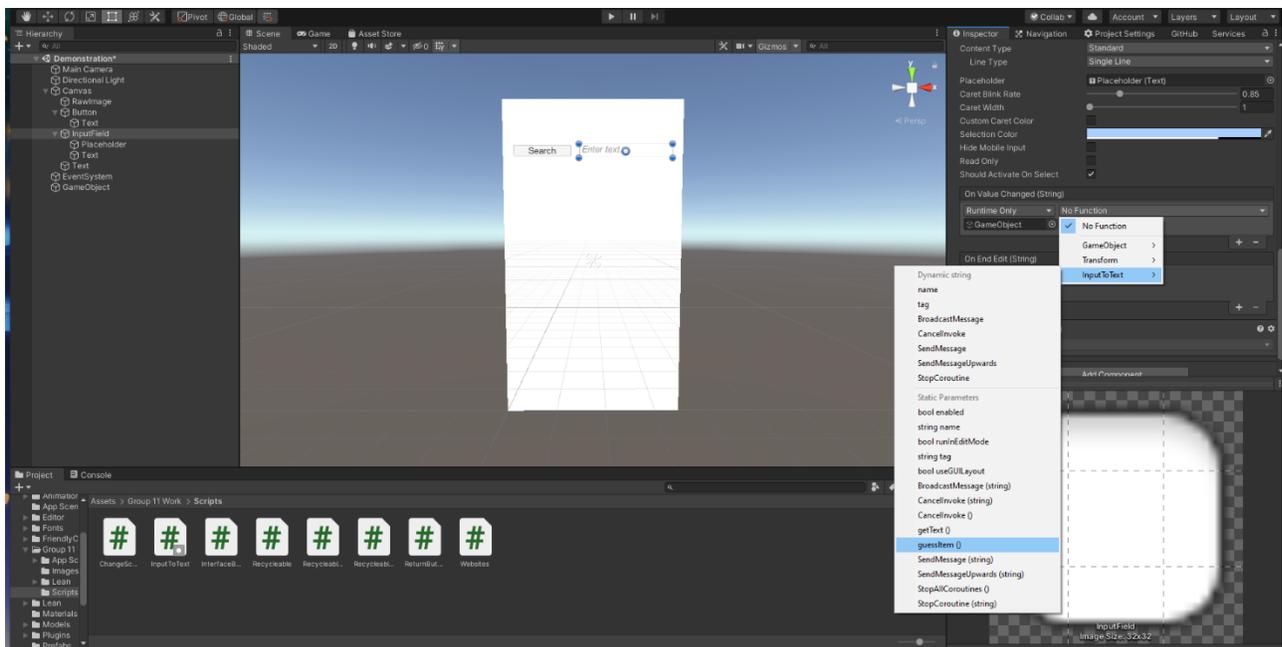


Figure 102. Adding a function to the text box

Now when pressing the play button, when text is typed in the search bar, the output text box should display text. However, the search button will not yet work, as the scenes to where the items should be placed have not yet been create. Create three new scenes in the same way as this

scene was created and name the scenes: “Black Bin”, “Blue Bin”, “Green Bin”, and “Garbage”. To see how to design these scenes, please reference section 6.1.3.2.3. Scenes.

6.1.3.2.2. Manual Search

The manual search is the series of buttons below the search bar on the manual search screen. Using the same method as the in 6.1.3.2.1, right click on the Canvas, go to UI and insert a button. Use the rect tool to resize the buttons and use the move tool to arrange the buttons on the Canvas. To organize the items, double clicking on the name in the Hierarchy will allow the button to be renamed which can also be done at the top of the inspector window for the button. Set 6 buttons in the Canvas and name them: “Plastics”, “Metal”, “Glass”, “Paper”, “Cardboard”, and “Carton” (figure 103).

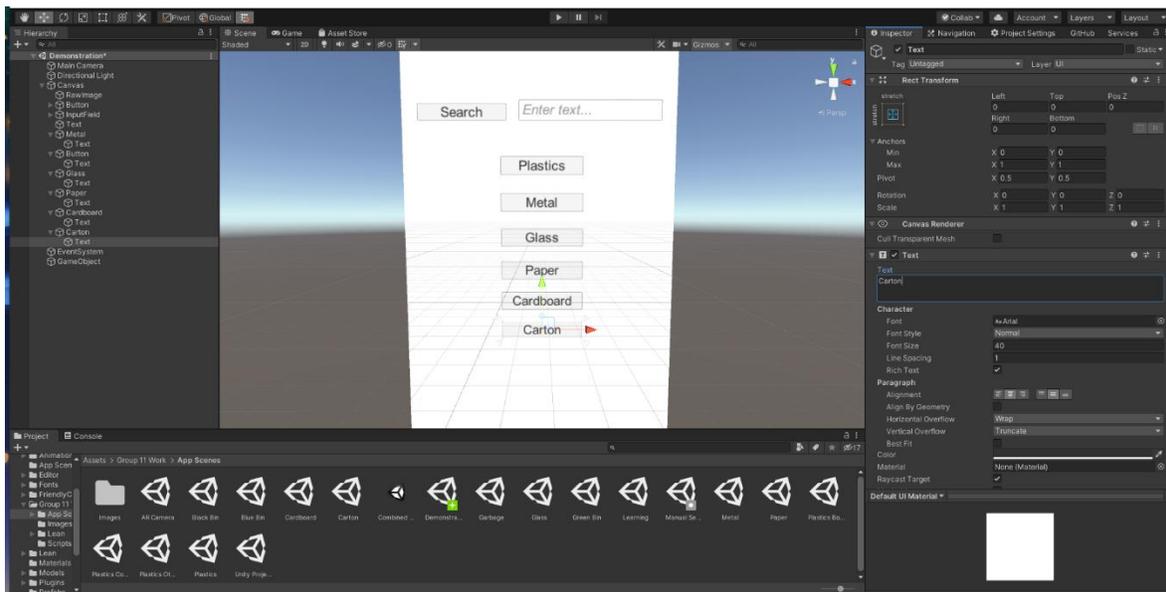


Figure 103. Adding category buttons to the manual sorting page

Now, create a new C# script called “ChangeScene” and copy the following text into the script, overwriting the default script:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class ChangeScene : MonoBehaviour
{
    public void ReturnButton()
    {
        string place = "Manual Search Screen";
        Go(place);
    }
}
```

```
public void Plastics()
{
    string place = "Plastics";
    Go(place);
}

public void Bottle()
{
    string place = "Bottle";
    Go(place);
}

public void Container()
{
    string place = "Container";
    Go(place);
}

public void OtherPlastics()
{
    string place = "OtherPlastics";
    Go(place);
}

public void Metal()
{
    string place = "Metal";
    Go(place);
}

public void Glass()
{
    string place = "Glass";
    Go(place);
}

public void Paper()
{
    string place = "Paper";
    Go(place);
}

public void Cardboard()
{
    string place = "Cardboard";
```

```
        Go(place);
    }

    public void Carton()
    {
        string place = "Carton";
        Go(place);
    }

    public void ToGreen()
    {
        string place = "Green Bin";
        Go(place);
    }

    public void ToBlack()
    {
        string place = "Black Bin";
        Go(place);
    }

    public void ToBlue()
    {
        string place = "Blue Bin";
        Go(place);
    }

    public void ToGarbage()
    {
        string place = "Garbage";
        Go(place);
    }

    public void toPlasticsBottle()
    {
        string place = "Plastics Bottle";
        Go(place);
    }

    public void toPlasticsContainer()
    {
        string place = "Plastics Container";
        Go(place);
    }
}
```

```

public void toPlasticsOther()
{
    string place = "Plastics Other";
    Go(place);
}

public void Go (string place)
{
    SceneManager.LoadScene(place);
}
}

```

Figure 104. ChangeScene code

Create a new GameObject by right clicking on the Hierarchy menu and choosing “Create Empty”. Attach this new script to the GameObject by dragging the script on top of the GameObject in the Hierarchy. For each button, add an On Click function, drag the new GameObject to it, pick the function ChangeScene and choose the function X where the X stands for the name of the button. Example: Metal() for the Metal button (figure 105).

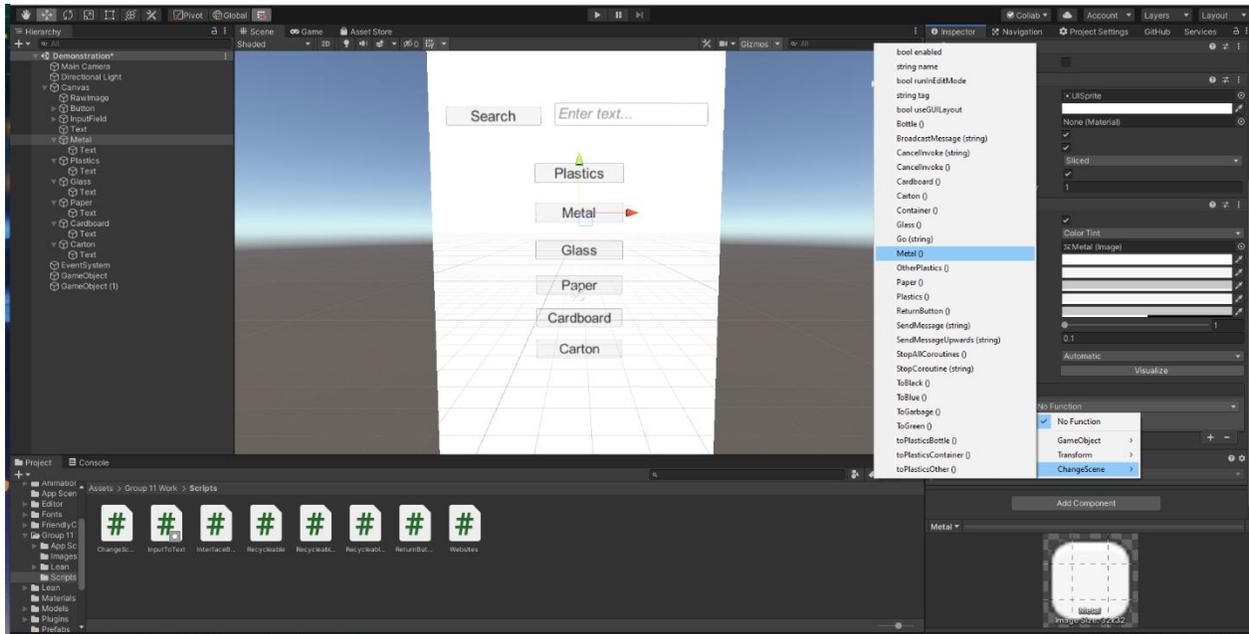


Figure 105. Adding functionality to Category buttons

Now create another nine scenes titled: “Plastics”, “Plastics Bottle”, “Plastics Container”, “Plastics Other”, “Metal”, “Glass”, “Paper”, “Cardboard”, and “Carton”.

In the scene “Plastics”, create a canvas, add a raw image background, and add three buttons with the names “Plastics Bottle”, “Plastics Container” and “Plastics Other”. Place an empty game object and add the script just created to it. From there, place the gameObject to each button’s On

Click() box, pick the function “ChangeScene” and then choose, toX, where X is the name of the button. For example, toPlasticsBottle for the “Plastics Bottle” button.

On each page, insert a canvas with a raw image background, along with a single button that should be titled “Return” as it is the return button (figure 106).

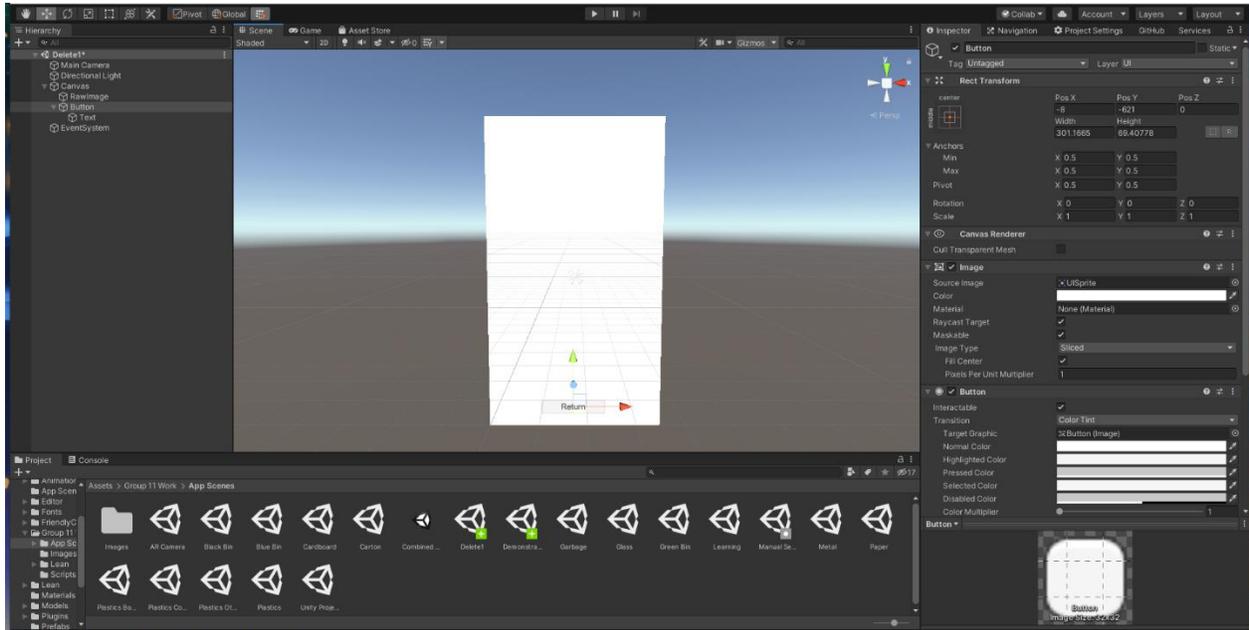


Figure 106. Adding a return button to the type sections

Create a new Script called “ReturnButton” and copy the following code:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class ReturnButton : MonoBehaviour
{
    public void ChangeScene()
    {
        string place = "Manual Search Screen";
        SceneManager.LoadScene(place);
    }
}
```

Figure 107. ReturnButton code

Then create an empty GameObject, put the script into it and attach the script to the Return Button’s On Click() box. Find the function “ReturnButton” and choose ChangeScene().

Finally, add text and raw images to the canvas to populate it with items that fit the page. Download images to unity by dragging them from a folder on the computer and then drag them to the raw image's texture box. Continue to do this until the final page has been created (figure 108).

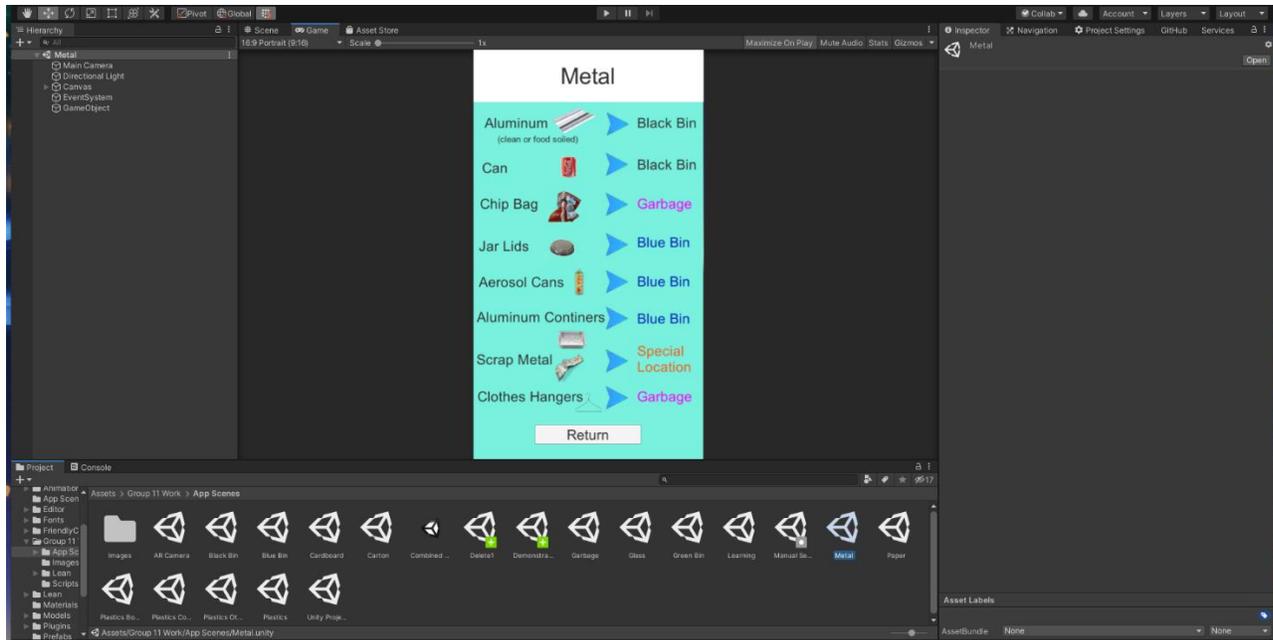


Figure 108. Finished Metal page

Replicate this with all the other scenes to create the manual search section of the page.

6.1.3.2.3. Scenes

The app has 4 scenes which direct the user where to recycle their recyclables. The scenes show up in the AR scanning sorting, the manual sorting, and the search function sorting. The appropriate scenes show up for the appropriate recyclable being scanned or searched. For example, when the AR scanner scans a barcode of a plastic bottle, the scene directing the user to put the recyclable in the blue bin would show up. If the barcode of a paper box was being scanned, then the scene would instead be one that directs the user to put the recyclable in the black bin. The same concept applies if the recyclable was being searched in the search bar or selected in the manual sorting.

There are 4 scenes in total. One for the black bin, one for the blue bin, one for the green bin which mostly applies to organic waste which is non-recyclable and one more for other-non recyclables which instead directs the user to dispose of the item in the garbage bin instead. Each scene displays text that lets the user know where the item belongs. The scene also contains a photo of the container that the item should be disposed in such as a black or blue bin. The background and the colour theme is also relevant to the scene. For example, the background of the black bin scene is a bunch of paper and the background of the garbage bin scene is a bunch of garbage bags. The colours are chosen to match up with the colours of the rest of the scene.



Figure 109: Blue Bin Scene



Figure 110: Black Bin Scene



Figure 111: Green Bin Scene



Figure 112: Garbage Bin Scene

6.1.3.3. AR

To build the AR camera, the Vuforia engine in Unity was implemented. The Vuforia engine allows developers to add image recognition to their AR app. A free development key allows for a 1000 target images to be recognized per month. Vuforia allows developers to store an online database of target images and objects, however the database must be downloaded and imported into the Unity project to be accessed by the app.

6.1.3.3.1. Setting Up Vuforia in Unity

To implement the Vuforia Engine in Unity, the Vuforia plugin must be installed using the package manager. The package manager can be accessed by selecting “package manager” under the windows tab in the tool bar.

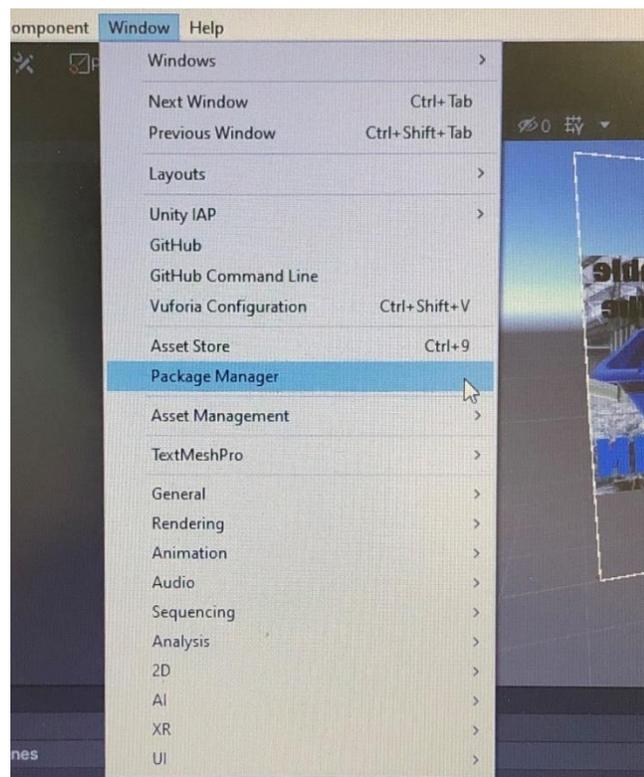


Figure 113: Opening Package Manager

In the package manager find the Vuforia Engine AR package and click install.

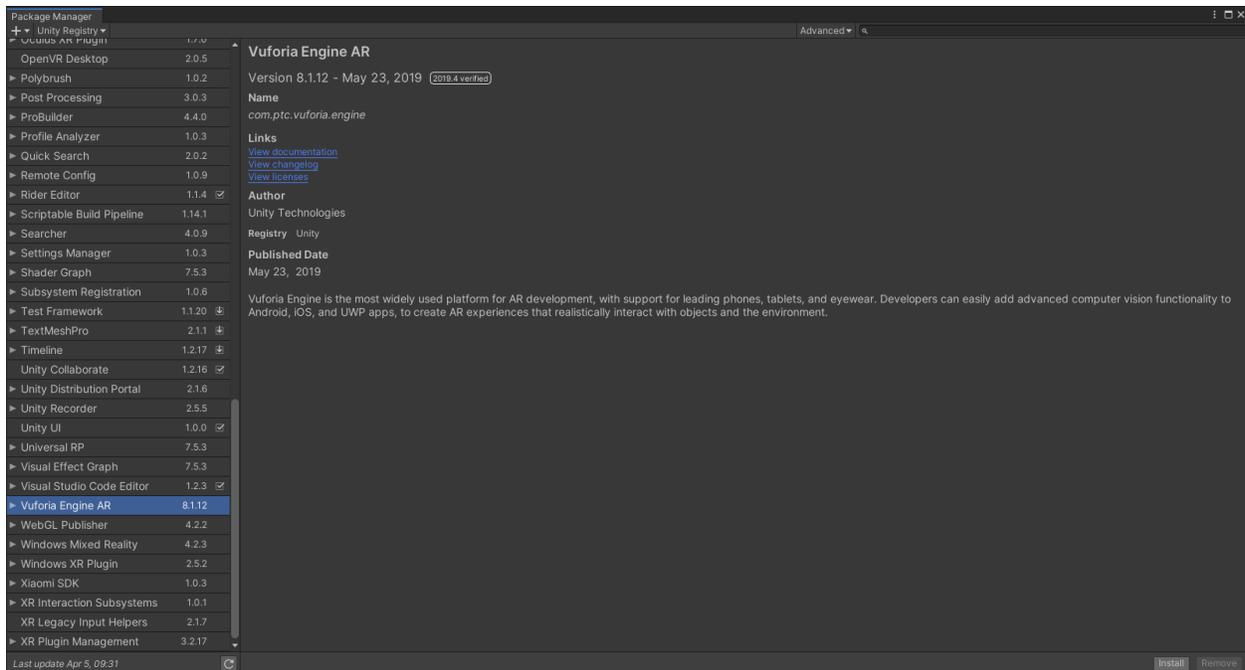


Figure 114: Installing Vuforia Engine AR Package

To utilize the AR camera, an AR camera must be created and configured in the Unity scene. To create the AR camera right click on the Hierarchy tab then select AR camera under Vuforia Engine.

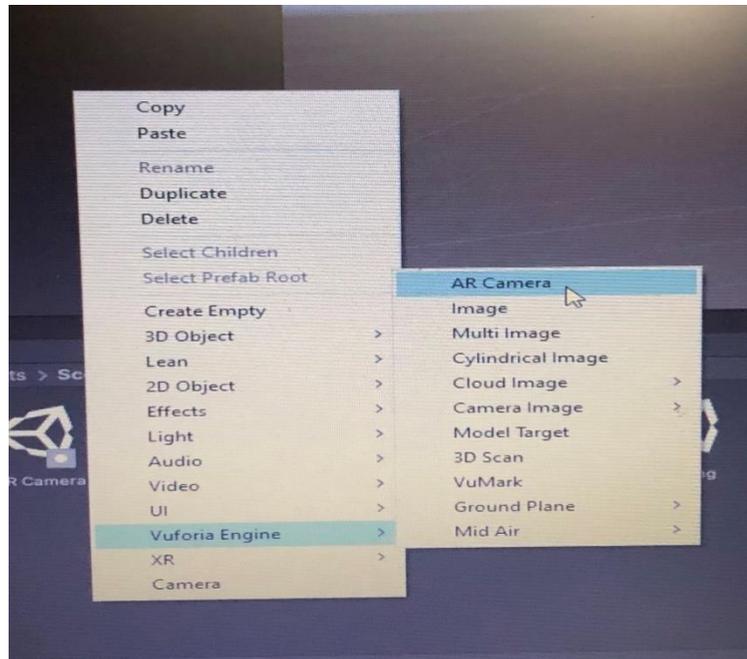


Figure 115: Adding AR Camera

After adding the AR camera make sure that the camera is tagged as the main camera for the scene; this will be done automatically if the AR camera is the only camera in the scene but otherwise will have to be tagged as the main camera in the inspector tab.

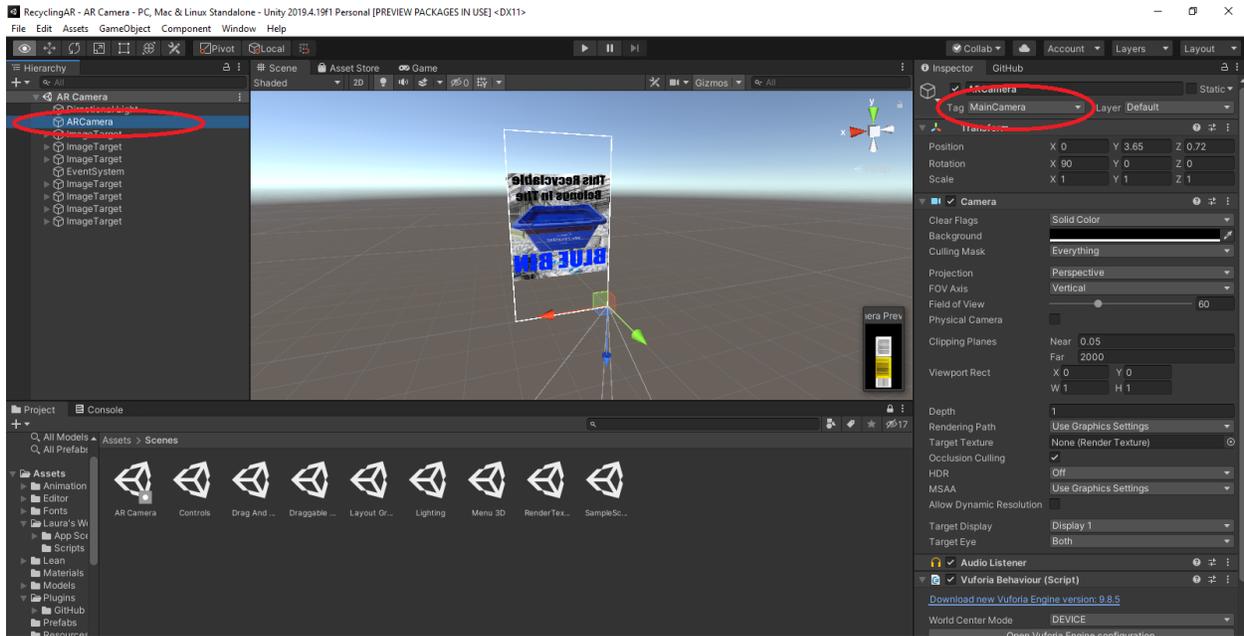


Figure 116: Tagging AR Camera as Main Camera

To configure the AR camera a license key is required. A free license key can be acquired through the Vuforia developer portal online. In the license manager click Get Development Key.

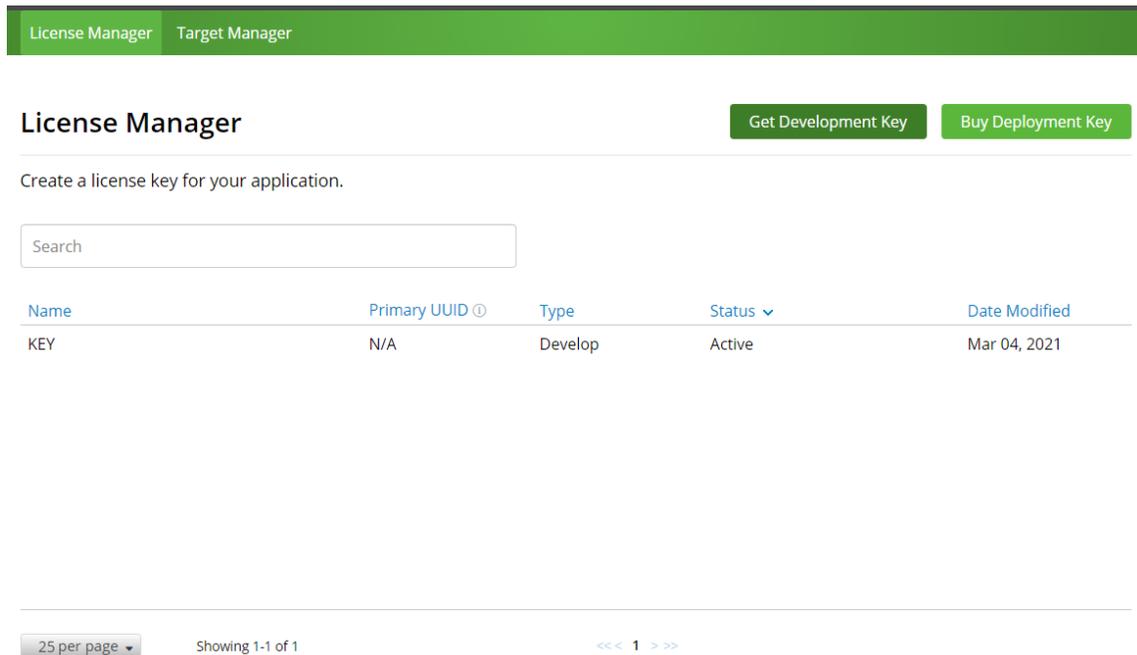


Figure 117: Getting License Key

After clicking on “Get Development Key”, a new page will appear listing the authorizations of the development key, name the development key and accept the license terms and conditions then click confirm.

[Back To License Manager](#)

Add a free Development License Key

License Name *

You can change this later

License Key

Develop

Price: No Charge

Reco Usage: 1,000 per month

Cloud Targets: 1,000

VuMark Templates: 1 Active

VuMarks: 100

By checking this box, I acknowledge that this license key is subject to the terms and conditions of the [Vuforia Developer Agreement](#).

Figure 118: Adding License Key

Once a key has been added to your Vuforia developer account you can now configure your AR camera. The Vuforia Engine configuration can be found in the AR camera inspector tab.

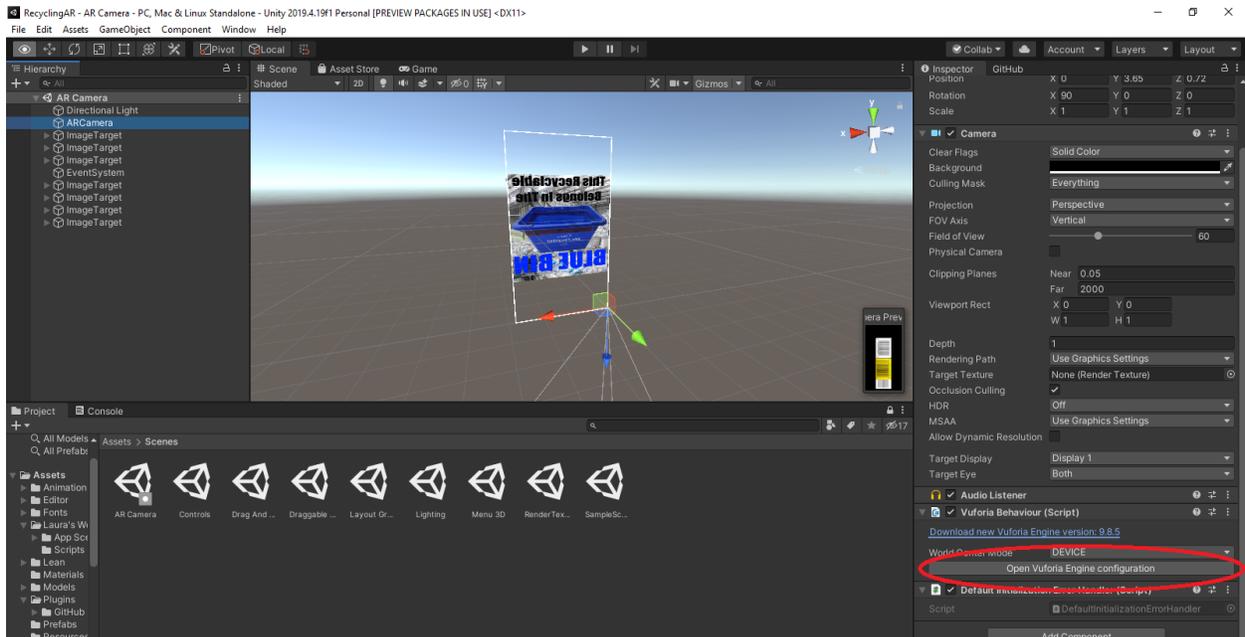


Figure 119: Opening Vuforia Engine Configuration

Copy the license key from the Vuforia developer portal and paste it into the license key box.

[License Manager](#) > KEY

KEY [Edit Name](#) [Delete License Key](#)

[License Key](#) [Usage](#)

Please copy the license key below into your app

```
AT83afX/////AAABmU2HG48UUUfmjII8miP1P21s16ccVtgSGc0tLo1od3eKJky1Ri24rPyNi8jX70nxx+YwnkavsZ0QsR1dxziNpCHwtT
CyN8OPwVW/uN6pWhDcwb2BNVPQeoFOucWb7xsqYJxv7yQDoB/zhU8QDcWkonQNNExd1Qq0uNgThfo+hY/QpF05aWNV16Shz7pPKz2D3TDZ
wXrDJ9gE0PJj2bWKKPS0uiJrdTvXNAffKeqolMowNzzCwBSem/Vn9eUGig1ZnFDJkeqUYexUb0k+qDHcStcMEyBK+MiLxzGrCfix0egvqJ
AGg2RMQNDUaekU+HggzSY7F/KgJCjxs/KLuGO5vEe1TpvizBBShgPPIJYeMeHK
```

Plan Type: Develop

Status: Active

Created: Mar 04, 2021 10:23

License UUID: b74ac5ea2d89475e9f86ac6518763f48

Permissions:

- Advanced Camera
- External Camera
- Model Targets
- Watermark

Figure 120: Vuforia License Key

After pasting the link into the app insure that the maximum tracked images and objects is set respectively to the number of images and objects intended to be tracked by the app.

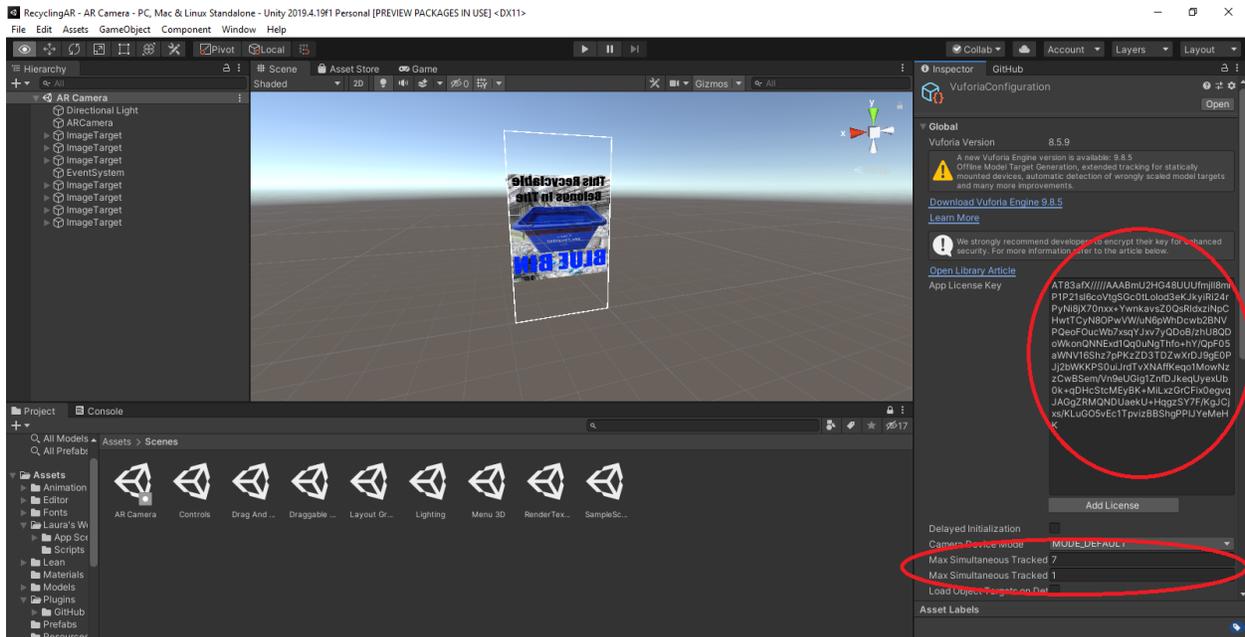


Figure 121: Unity Vuforia Engine Configuration

To build the AR camera, the Vuforia engine in Unity was implemented. The Vuforia engine allows developers to add image recognition to their AR app. A free development key allows for a 1000 target images to be recognized per month. Vuforia allows developers to store an online database of target images and objects, however the database must be downloaded and imported into the Unity project to be accessed by the app.

6.1.3.3.2. Adding Target Image Database to Vuforia

To create a target database click add database in the Target Manager page on the Vuforia Developer website.

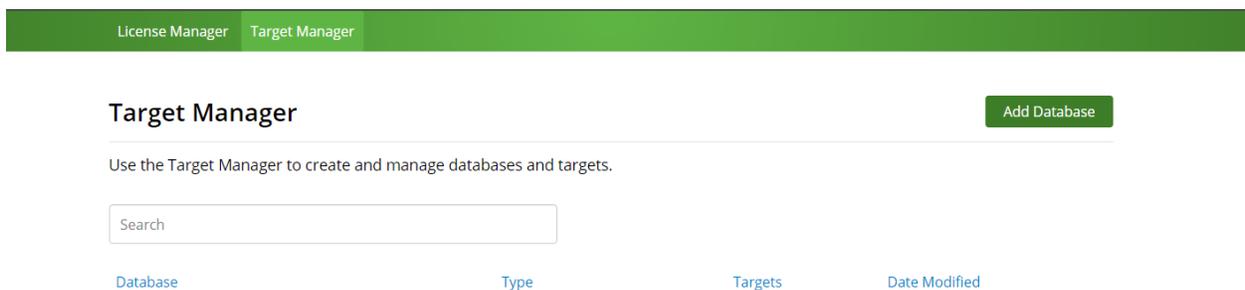


Figure 122: Adding Target Database

Once “Add Database” has been clicked on a pop-up will appear asking for the name of the database and the type of database (On the device, Cloud, or VuMark).

Create Database

Database Name *
DatabaseName

Type:

Device
 Cloud
 VuMark

Cancel Create

Figure 123: Creating Database

6.1.3.3.3. Adding Target Image to Vuforia

To create target images that were easy for the AR camera to recognize, bar codes were scanned and stored as pdf documents. The documents were then converted into jpg files at 300dpi resolution to be compatible with the Vuforia image recognition software, some larger files were converted at a resolution of 200dpi to meet Vuforia's upload storage limit.

To upload an image target to your Vuforia database, in your database menu click on "Add Target".

Target Manager > ARCamera

ARCamera [Edit Name](#)

Type: Device

Targets (7)

Add Target Download Database (All)

<input type="checkbox"/>	Target Name	Type	Rating ^①	Status [▼]	Date Modified
<input type="checkbox"/>	 METAL_CAN_BAR_CODE	Single Image	★★★★☆	Active	Mar 28, 2021 11:09
<input type="checkbox"/>	 BAR_CODE_BAKING_POWDER_C...	Single Image	★★★★☆	Active	Mar 28, 2021 11:08
<input type="checkbox"/>	 TEXTBOOK	Single Image	★★★★★	Active	Mar 28, 2021 11:06
<input type="checkbox"/>	 KLEENEX_BOX	Single Image	★★★★★	Active	Mar 28, 2021 10:56
<input type="checkbox"/>	 GarbageCode1	Single Image	★★★★★	Active	Mar 06, 2021 11:27
<input type="checkbox"/>	 GoldFishBox1	Single Image	★★★★☆	Active	Mar 06, 2021 10:52
<input type="checkbox"/>	 BublyCanCode	Single Image	★★★★★	Active	Mar 04, 2021 15:07

Last updated: Today 08:38 AM [Refresh](#)

Figure 124: Add Image Target

After clicking on “Add Target”, a menu will appear, to add a bar code select single image type and upload the jpg file. For the purposes of this project the width of the image is unimportant, for simplicity a width of 1m was used. Name the image target accordingly as it will be important to distinguish the image targets later.

Add Target

Type:

Single Image Cuboid Cylinder 3D Object

File:

.jpg or .png (max file 2mb)

Width:

Enter the width of your target in scene units. The size of the target should be on the same scale as your augmented virtual content. Vuforia uses meters as the default unit scale. The target's height will be calculated when you upload your image.

Name:

Name must be unique to a database. When a target is detected in your application, this will be reported in the API.

Figure 125: Creating Image Target

Once uploaded to the Vuforia database, defining characteristics on the target images were recognized by the Vuforia software and saved to the target image. Always check the image after it has been identified by Vuforia to ensure that the reference points chosen by Vuforia are appropriate. To do so, select the image target in the database then select show features.

BublyCanCode

Edit Name Remove



Type: Single Image
Status: Active
Target ID: 1cc1002935eb46a69823c697063b6842
Augmentable: ★★★★★
Added: Mar 4, 2021 11:20
Modified: Mar 4, 2021 15:07

Figure 126: Showing Distinguishing Features

Ensure that the image target features relate only to the bar code, not any temporary markings such as glare or dents in the material. A good example of an image target is shown below.

BublyCanCode

Edit Name Remove

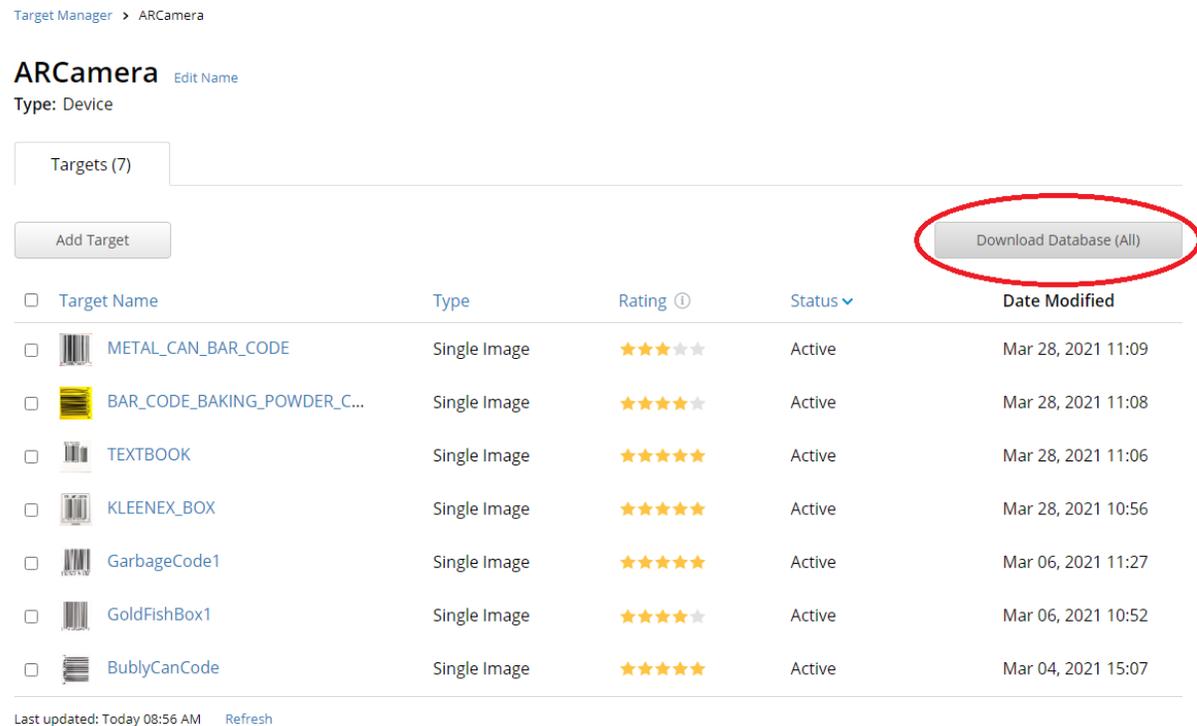


Type: Single Image
Status: Active
Target ID: 1cc1002935eb46a69823c697063b6842
Augmentable: ★★★★★
Added: Mar 4, 2021 11:20
Modified: Mar 4, 2021 15:07

Figure 127: Appropriate Image Target Features

If the image target contains insufficient identifying features or has undesirable features it may be necessary to take another picture or reupload the target image.

After the database contains the desired targets (bar codes for this project), select “Download Database (All)” to download the target database. The Database should be saved as a Unity Package file.



The screenshot shows the Target Manager interface for a project named 'ARCamera'. The interface includes a header with the project name and an 'Edit Name' link. Below the header, there is a 'Type: Device' label and a 'Targets (7)' indicator. A table lists seven targets, each with a checkbox, a barcode icon, a name, a type, a rating, a status, and a date modified. A 'Download Database (All)' button is circled in red in the top right corner. At the bottom, there is a 'Last updated: Today 08:56 AM' label and a 'Refresh' button.

<input type="checkbox"/>	Target Name	Type	Rating ⓘ	Status ▾	Date Modified
<input type="checkbox"/>	 METAL_CAN_BAR_CODE	Single Image	★★★★☆	Active	Mar 28, 2021 11:09
<input type="checkbox"/>	 BAR_CODE_BAKING_POWDER_C...	Single Image	★★★★☆	Active	Mar 28, 2021 11:08
<input type="checkbox"/>	 TEXTBOOK	Single Image	★★★★★	Active	Mar 28, 2021 11:06
<input type="checkbox"/>	 KLEENEX_BOX	Single Image	★★★★★	Active	Mar 28, 2021 10:56
<input type="checkbox"/>	 GarbageCode1	Single Image	★★★★★	Active	Mar 06, 2021 11:27
<input type="checkbox"/>	 GoldFishBox1	Single Image	★★★★☆	Active	Mar 06, 2021 10:52
<input type="checkbox"/>	 BublyCanCode	Single Image	★★★★★	Active	Mar 04, 2021 15:07

Figure 128: Downloading the Target Database

6.1.3.3.4. Implementing Vuforia Image Targets into Unity

The targets are imported into Unity Project by clicking on Import Custom Package under the Asset tab in the Unity project, then selecting the target database. After selecting a Unity Package, a menu will appear displaying the files in the Unity Package. To avoid conflicts it may be necessary to deselect objects in the Unity Package file that have been downloaded in previous packages. To finish importing select import.

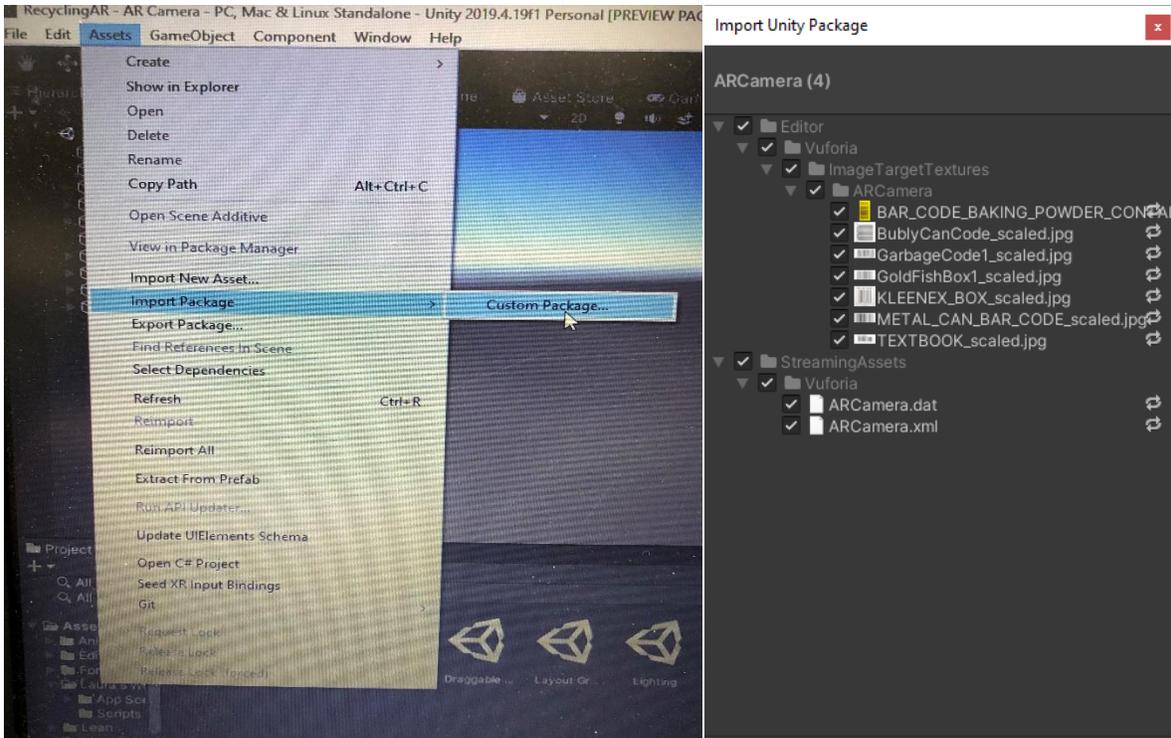


Figure 129: Importing Custom Package

To create an image target in the AR scene, right click in the hierarchy tab then select image under the Vuforia section.

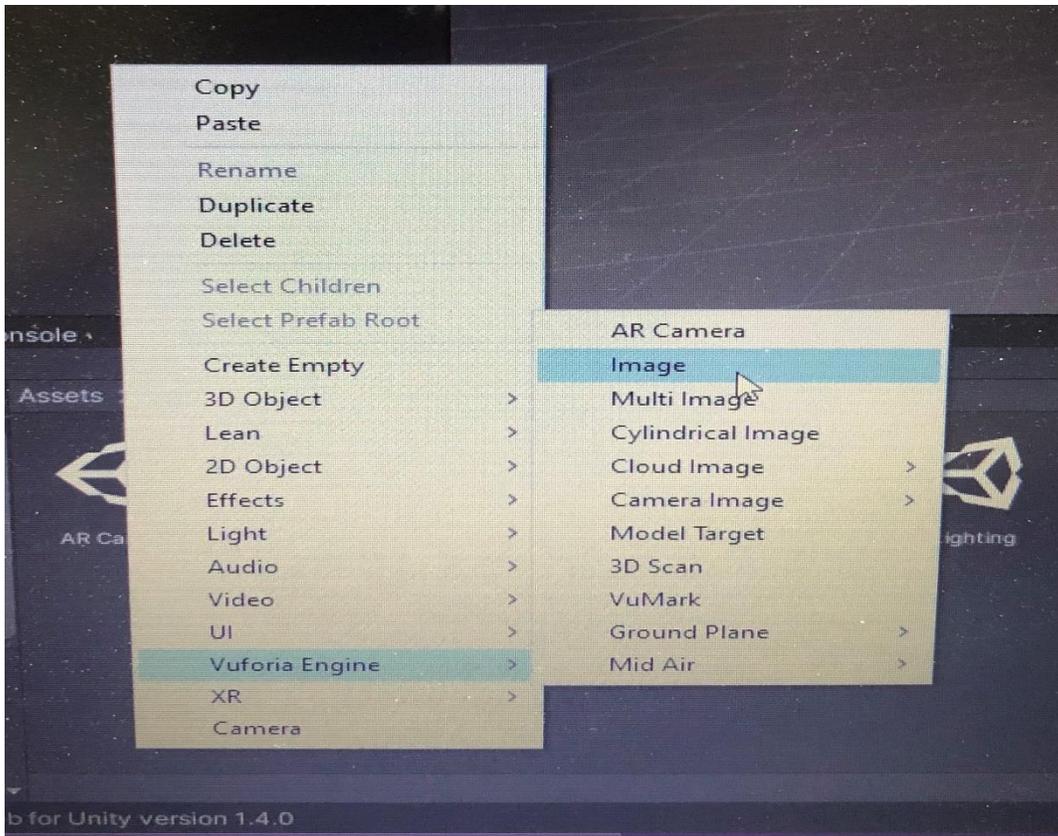


Figure 130: Creating Image Target in Unity Scene

Once an image target object has been added to the Unity scene, the image target can be chosen in the inspector tab of the image target object. To change the image target, select the desired database image target in the inspector tab.

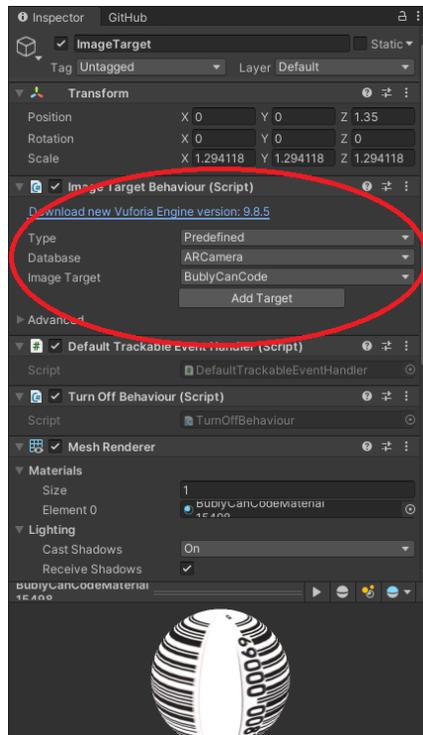


Figure 131: Target Image Inspector Tab

6.1.3.3.5. Creating Bin Notices

The bin notices were placed on a canvas so that they would be easy for the user to read. To create a canvas right click on the hierarchy while the image target is selected, then click on canvas under the UI section.

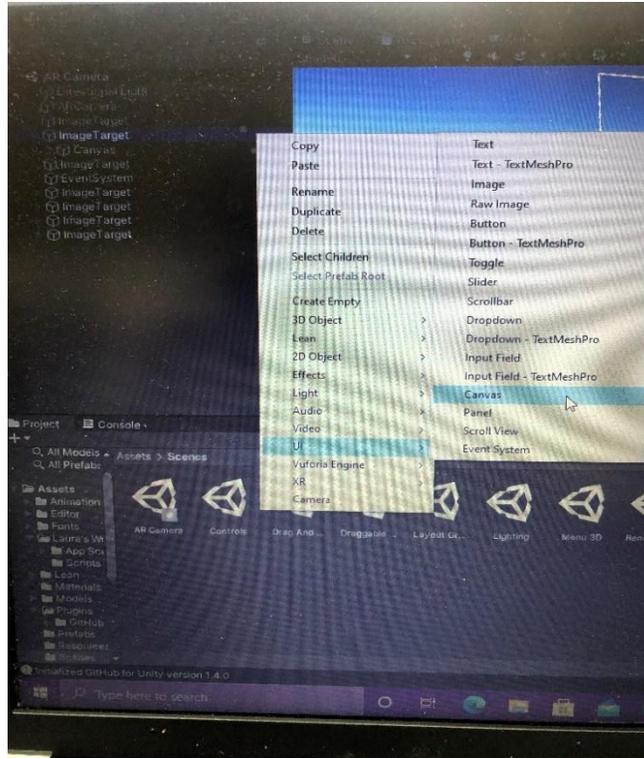


Figure 132: Adding a Canvas

To add the image to the canvas right click the hierarchy while the canvas is selected, then click on image under the UI section.

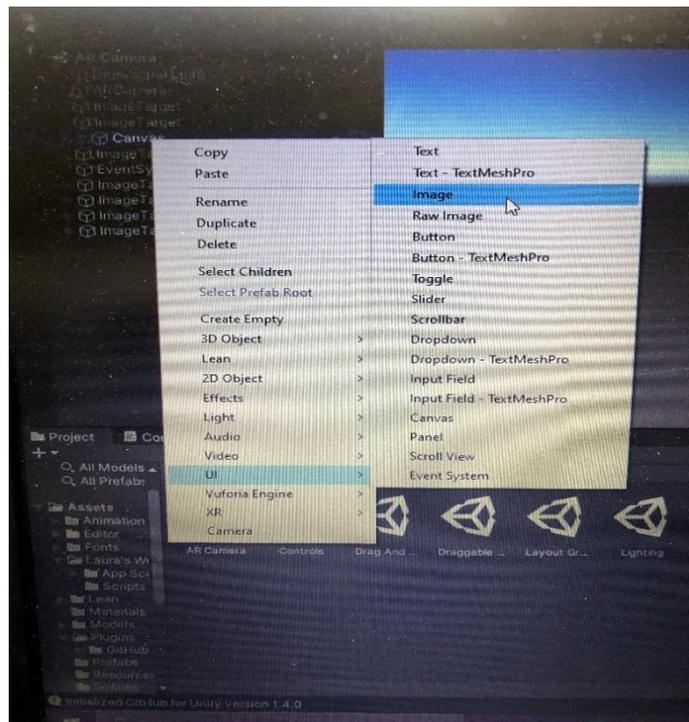


Figure 133: Adding an Image to Canvas

To select a source image, click on the source image in the inspector window while the image is highlighted in the hierarchy. The bin scene images were used as the bin notice images. The image must be a 2D sprite image type.

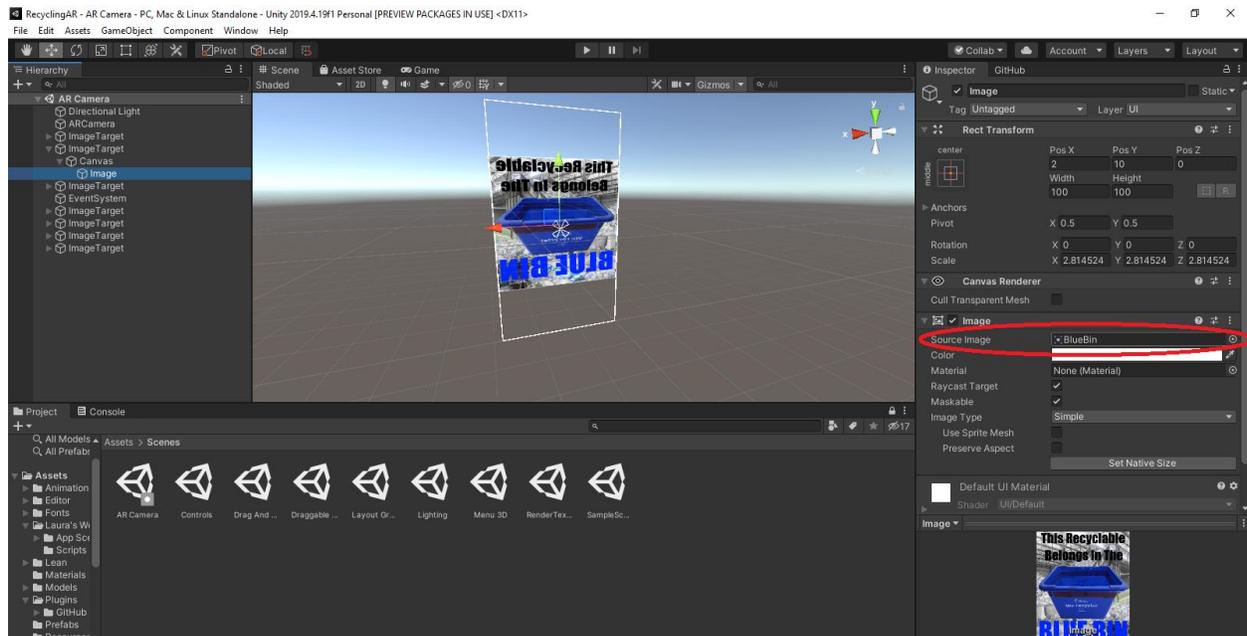


Figure 134: Setting a Source Image

Once The image has been added to the canvas, center the image using the tools in Unity.

6.1.4. Testing & Validation

Testing has been done on the app and website throughout the many prototypes. The testing has been done by various users and their feedback has been recorded and used. The feedback from the testing has been used to develop later prototypes and come up with new improvements to the app and website.

List of tests:

- Can the app scan barcodes and provide recycling instructions with accuracy?
- Can the manual sorting mechanic guide the user to the correct item?
- Can the app be accessed on all platforms? If not, which ones are available?
- Can the app recognize items without bar codes?
- Can the search mechanic return the specified items with instructions?
- Is the website searchable? Does it improve ease of use?
- How are the aesthetics of the product?

List of feedback:

- Scanner is quick and convenient
- Scanner is dependent on lighting conditions
- Scanner is not usable if barcode is not present on item

- Search bar needs autofill
- Manual sorting needs to be easier to use
- Manual sorting needs audio queues when buttons are tapped
- Needs better colour theme
- Scroll bar would be useful
- Add method of inputting barcode numbers into app
- Improve aesthetics of app
- Add image recognition of recyclable, not just barcode scanner
- Thought that the background of the notice image was a little busy
- Would prefer a neutral background on the notice image
- Have a way to return to both the general plastics page and the manual sorting page from the individual plastic categories pages
- Should have a way to return to the main menu from the different manual sorting pages
- When gibberish is typed into the search bar, it still maintains that what you are typing is correct. Should indicate the nothing with that lettering is found in database
- Questioned the use of the word AR – people may not know what it means
- Enter button on keyboard does not make the search bar work
- How would you address spelling mistakes? Can you address spelling mistakes?
- After error message about item not in database (needs to be fixed) the database list of items below does not always update
- The database search list below search bar should include everything like what the user is typing – not just a single item – and should fill the entire screen
- Also pointed out that a drop list would be ideal
- Items in search menu below should sort in alphabetical order
- When typing in gibberish the list does not remove items so has items that do not match the input
- The suggestions should not begin until after 2 characters have been typed into the search bar
- On the manual search pages, the different components of the items should link to their own recycling instructions (Ex. From the jar page should be a way to easily find out how to recycle the lid)

List of changes made due to feedback:

- Aesthetics improved
- Colour theme improved
- Backgrounds improved
- Improved font, typeface and colours of text and title
- Added icons to the title screen
- Changed colours and background design of main menu
- Manual sorting improved to sort items without barcodes
- Error message for search bar added if text inputted into the search bar is not in database

- Autofill added for search bar when items are being typed in

6.2. Website

6.2.1. Bill of Materials

Table 5: Bill of Materials of Website

Item Name	Purpose	Price
Wix Website Builder	Build the Website	Free
Wix Search Bar	Implement a search bar into the website	Free
SEO Google	Make the website searchable	Free
	Total Final Price:	FREE

6.2.2. Equipment List

Table 6: Equipment List of Website

Equipment	Use
Wix	Software used to build the website
Zee Maps	Software used to create interactive maps

6.2.3. Instructions

To build the website, Wix Website Builder was used. This software comes with many features already implemented so it is very easy to create any type of website. Once a format has been chosen, a selection of tools on the left-hand side of the screen will be used to add elements and pages (figure 135). The ‘+’ button can be used to add any text boxes, images, buttons and many more.

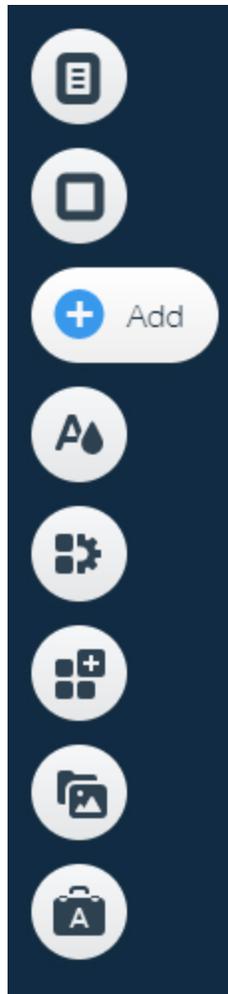


Figure 135: Left Side Editor Navigation Bar

When buttons are added they can be linked to other pages of the website by choosing another page when setting the button link (figure 136).

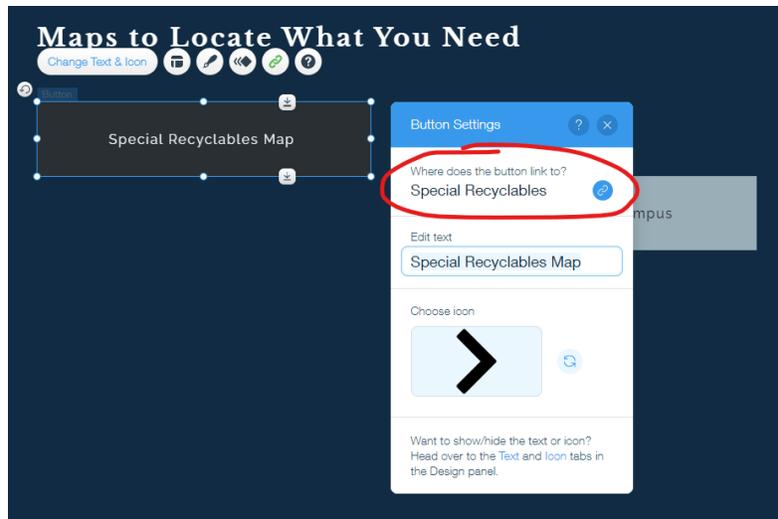


Figure 136: Button Linked to Another Page

External websites can also be embedded into the pages of the website and this feature was used to implement the interactive maps for Special Recyclables and Waste Bin Locations. The external website used to create the interactive maps is called Zee Maps (<https://www.zeemaps.com/>) and it is very easy to use. After creating a new map, adding a description and the details, locations can be added through an address or by selecting a specific location on the map. To add the location to the map, the name, details, and description must be filled out, and then submitted. To embed the maps on the website, click ‘Print or Share’ in the top left navigation bar, select ‘Publish in Website’ and select the desired functions when embedded and copy the code (figure 137).

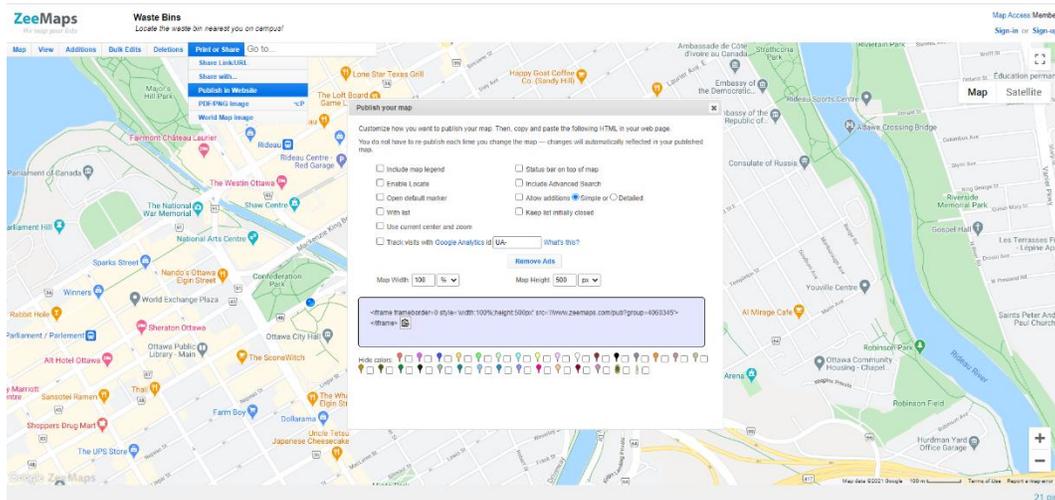


Figure 137: Zee Maps Embedding Process

After copying the HTML code, add an embed feature using the left navigation bar (figure 138) the Wix website and paste the code into the HTML text box (figure 139).

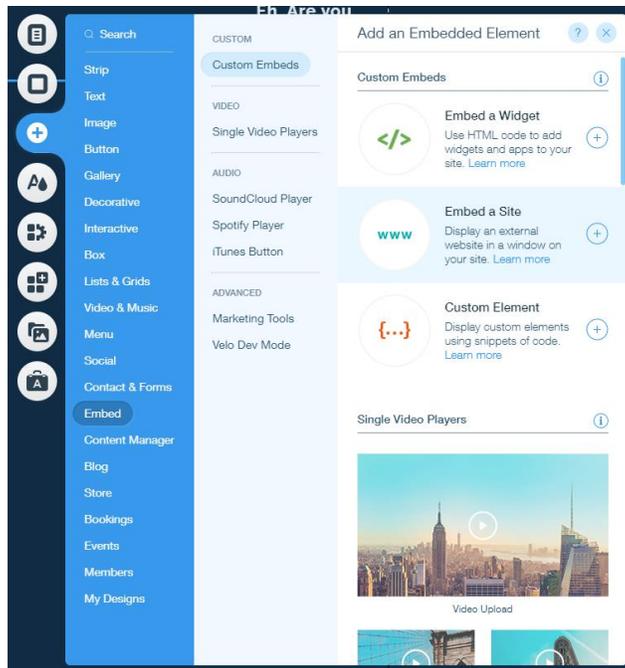


Figure 138: Embed Feature in Left Navigation Bar

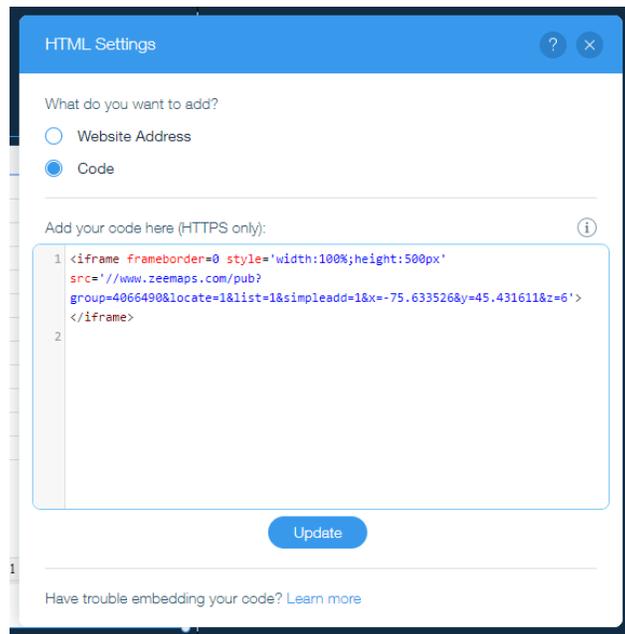


Figure 139: HTML Code for Special Recycling Locations Map

To create the Sort the Recycling game on the game page, an external website <https://www.classtools.net/vortex/> and information that was found for the manual sorting pages was used. It was very straightforward and just needed 4 categories and as many items to be placed into these categories as desired (figure 140). The game was then linked to a button that was added to the game page.

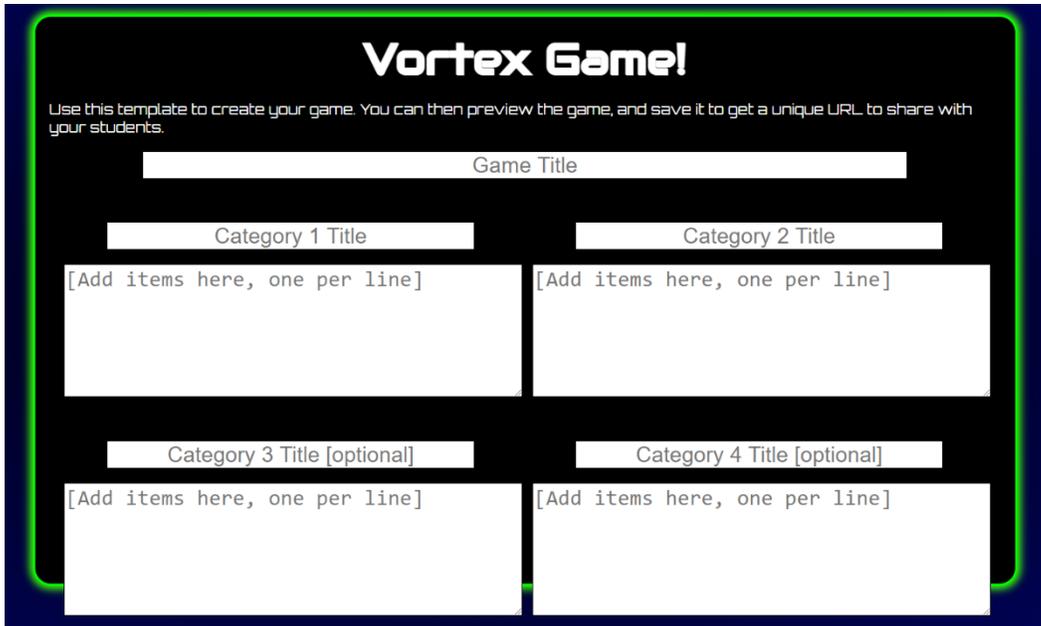


Figure 140: Creating the Sort the Recycling Game

The Education and News page required some research before it could be completed. Four websites were found, and four buttons were inputted (figure 141) into the page each linked to individual external resources and named to match the source (figure 142).

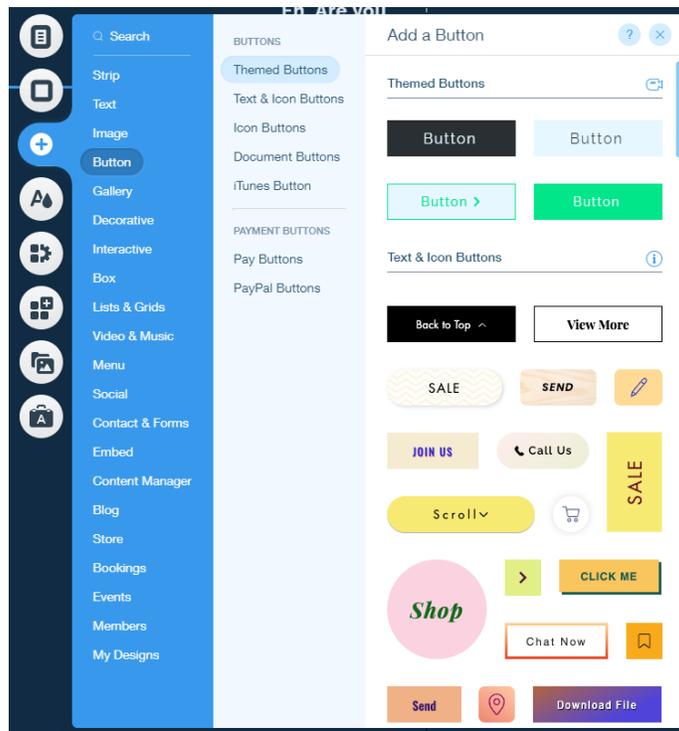


Figure 141: Adding Buttons

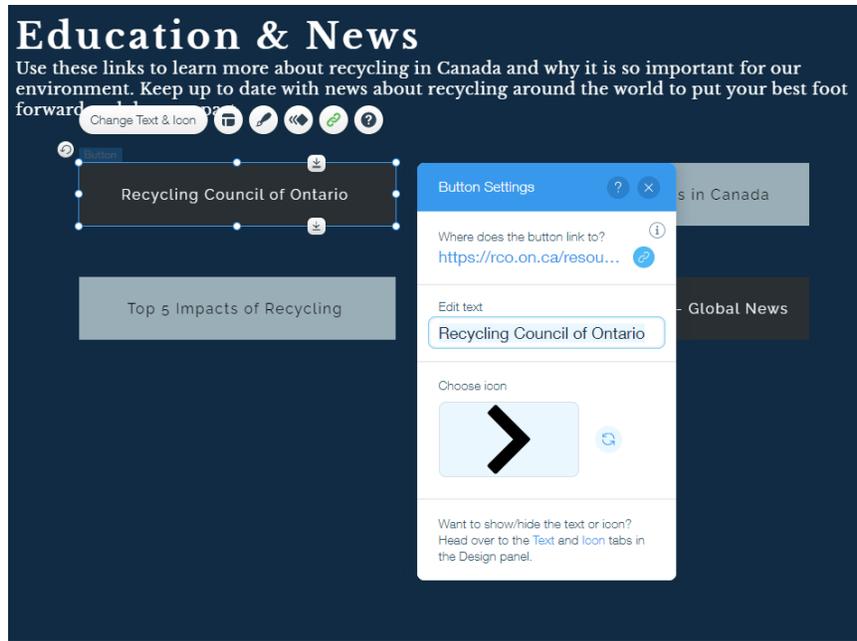


Figure 142: Linking the Button to External Websites

The discussion forum and members page can be inputted through the 'Add Apps' feature (figure 143). Both are added together when the Wix Forum App is selected, and it automatically creates the admin profile and first forum introductory post.

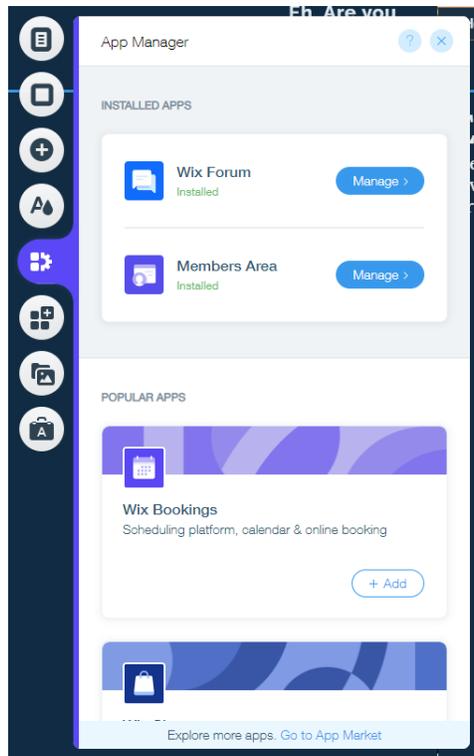


Figure 143: App Manager - Wix Forum

6.2.4. Testing & Validation

Testing was performed on each prototype and the final design by allowing users to use the product and simultaneously give feedback. Prototype 2 was also shown to the client to receive feedback for improvements that should be implemented before the final design presentation. The feedback that was received over the three prototypes are:

- Larger manual sorting database
- More special recyclable locations and information such as operating hours
- More details about washing plastics before recycling
- Different colours of waste bins in Ottawa
- Information about disposing of garden waste and clothing
- Addition of a search bar
- The map on the website should also have normal recycling bins
- Add instructions about how to throw away garbage – bags or no bags, garbage waste bundles, etc.
- Make information about #6 plastics more clear
- Add information about how to recycle mattresses, small appliances, and wood scraps
- Add information about users being able to add locations on the maps themselves

This feedback was then used to further develop the prototypes and final design. Once the final design was completed, it was tested with another small group of users, and the implementation of the previous feedback was noted and praised.

7. Conclusions and Recommendations for Future Work

The solution that was created to make recycling easier and more accurate for the Ottawa region consisted of an app and a website. The app was created in Unity and the website was created in the Wix Website Builder. After three prototypes and the final design, there was a functioning solution and future work to improve the product was planned.

7.1. App

One of the recommendations for future work is to publish the app on the App Store and Google Play Store. In order to publish the app on the App Store, many things must be done. For example, you need to enroll in the Apple Developer Program which costs 99\$ every year. The app must also be submitted for review to ensure that the app follows the App Stores guidelines. Setting your app's page such as setting the price, describing the app, advertising the app, and adding other details to your app's store page must also be done. Although it is not very easy, publishing an app on the App Store is very realistic and a good future goal for our app since it will allow people to download our app from the App Store if they use Apple devices. The same concepts apply to the Google Play Store and publishing the app on the Google Play Store is also a good future goal since it will allow Android users to download our app.

Other recommendations include adding a French option for people who speak mostly French and are not as fluent in English. This is especially useful since Ottawa is in Canada and Canada has French as a national language as well and French speakers are likely to use this app.

Some final improvements to the app could be expanding the database for the AR scanner, the manual sorting and the search bar. Adding a drop-down menu for the search bar like many search bars will increase the user-friendliness. Improving the aesthetics of the app, especially by fixing the screen ratio would also be good. A social system that lets you interact with other users of the app would make it more appealing and interesting. Finally, some recycling games should be added to the app to make it more entertaining, especially for children.

7.2. Website

Improvements on the website will continue to occur regularly to ensure that it is up to date with all relevant information, and more features will be added. Some further additions that will be made to the website in the future is adding more games and links to the Education and News page and expanding the Waste Bin Locations map past the University of Ottawa campus to cover all of Ottawa. Another important change for the future would be the connection between the website and the app. Once the app can be downloaded off of the App Store/Google Play Store, the link to download will be added to the website's home page to promote the use of both platforms.