GNG 1103

Design Project User and Product Manual

EcoTech User and Product Manual

Submitted by:

EcoTech, Group C8

Claire Christensen, 300188934

Isaiah Philip, 300189093

Tye Segal-Kawano, 300195291

Yufei Zhao, 300193345

Yang Chen, 300187517

April 18, 2021

University of Ottawa

Table of Contents

List of Figures

---

1. Introduction

The purpose of this deliverable is to outline the application EcoTech built to help Ottawa households sort their waste items. At the beginning of this project, we were introduced to our client Mitch Bouchard who identified the misplacement of waste items which was leading to excess amounts of waste ending up in landfills. Miss sorting waste items into recycling or compost contamination has been known to negatively affect the environment. Thus, our client needed an effective method to properly sort recyclable materials into their appropriate bins. The following report is a user manual to our solution, EcoTech, which is an application that is able to help others properly dispose of their waste.

2. Overview

Roughly 52% of waste taken to landfills in the Ottawa area was miss sorted and should have ended up in a different bin so it could be properly taken care of.[1] These misplaced items are filling up landfills and harming the environment.

A need existed for households to pay closer attention and become more knowledgeable about the type of items they are throwing away. This was done by creating an AR based app to help its users sort their items and also to become more aware of different items they may come across. This solution had to be simple, easy to use, low cost, and able to follow the Ottawa Waste Management guidelines. To set our solution out from the rest we created an app with two ways for the user to find their item, as well included a game to test users on what they know.

2.1 Final Product

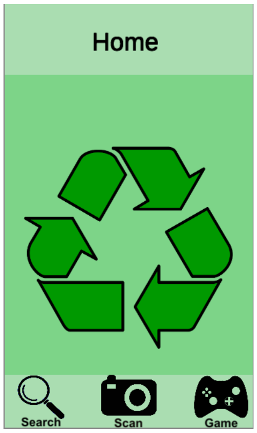
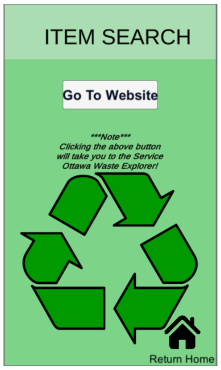


Figure 1. Home Screen

In Figure 1, a home screen was created using Unity. This screen includes the recycling logo, a title at the top, and three buttons: Search, Scan, and Home, which takes the user to different screens of the app.

---

[1] https://ottawa.ca/en/garbage-and-recycling/solid-waste-data-and-reports



Note
Clicking the above button will take you to the Service Ottawa Waste Explorer!

Figure 2. Search Screen

Figure 2 is the home search screen, this screen has a button that opens the City of Ottawa Waste Explorer in a browser. There is a warning to let the user know that the button will open the website, as well it has a title to tell users which portion of the app they are in. There is also a home button in the bottom right corner which will redirect the user to the home screen, Figure 1.

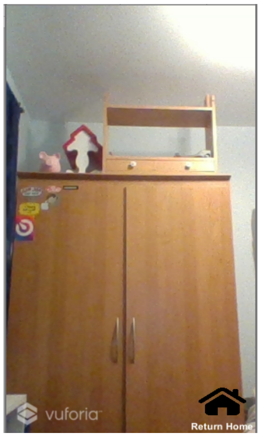


Figure 3. Scan View Without Item

Figure 3 is the view of when the scan button is clicked from figure 1 and an item is not recognized. When the camera doesn't recognize an item shown the camera will remain open with an optional home button in the bottom right to redirect users to the home screen seen in figure 1.

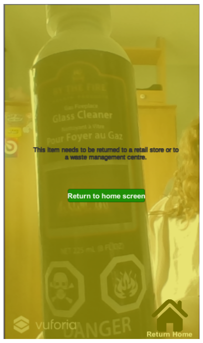

This item needs to be returned to a retail store or to a waste management centre.

Figure 4. Scan View With Recognized Item

The above figure is a screenshot of when the scanning component recognizes an item. Depending on the item the colour of the screen will change between yellow, blue, and black. Each item will have a tailored message on how to properly sort the item. This screen also has an option to return to the main screen by clicking the green 'Return Home' button in the middle of the screen.

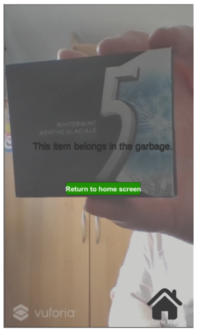

This item belongs in the garbage.

Figure 5. Another Example of the Scan View

Figure 4 is another screenshot of a different item being recognized, as mentioned the colour of the background changed for the colour of which bin the item should be sorted in, has a message on how to throw the item away, as well as a return home button.

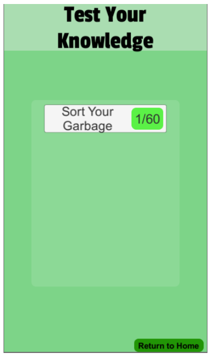


Figure 6. Game Home Screen

Figure 5 is the screen that the user is redirected to when they click the ‘Game’ button from figure 1. This screen contains two main buttons: 1. ‘Return to Home’ which takes the user back to the home screen and 2. ‘Sort Your Garbage’ which when clicked starts the game. Your previous attempt scores will be kept as seen in the green square beside the ‘Sort Your Garbage’.

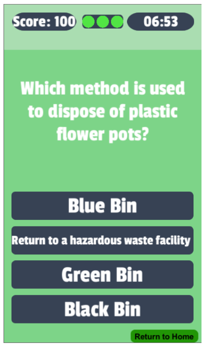
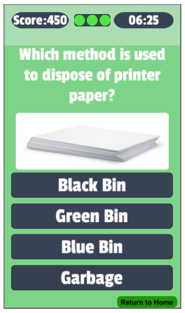

Which method is used to dispose of plastic flower pots?

Figure 7. In-Game Without Image

Figure 6 is a screenshot of when the game is started. A question appears with four multiple choice options and a return home button which will take the user to the main screen. When a question is answered it will flash green for right and red for wrong, and then a new question will appear. The user has 7 minutes to answer 60 questions based on the City of Ottawa waste guidelines, the timer is counting down in the top right corner. A score will be kept in the top left corner that increases when questions are answered correctly by 50 points, but does not when the question is answered incorrectly. The user will be given three lives at the beginning of the game, and will lose one life for each wrongly answered question.

Once the game is completed or all lives are lost a pop up will appear that prompts the user to retry the game, but the user can return to the home screen.



Which method is used to dispose of printer paper?

Figure 8. In-Game with Image

As seen in figure 7 some questions are accompanied with an image to help the user visualize objects.

2.2 Cautions and Warnings

This app requires a functional camera to use the scanning(AR) portion of the app, without a functional camera when the scanning option is selected it will open but only a black screen will appear. There are also no warnings or caution pop-ups in the app that will appear if something goes wrong. Lastly, permission may be required from the City of Ottawa to use its website in the app, if you are looking to publish it under your name.

3. Getting Started

3.1 App

This app had multiple steps to achieve the current version of it which will now be described. The user should have a recently made computer and smartphone to create and test the app.

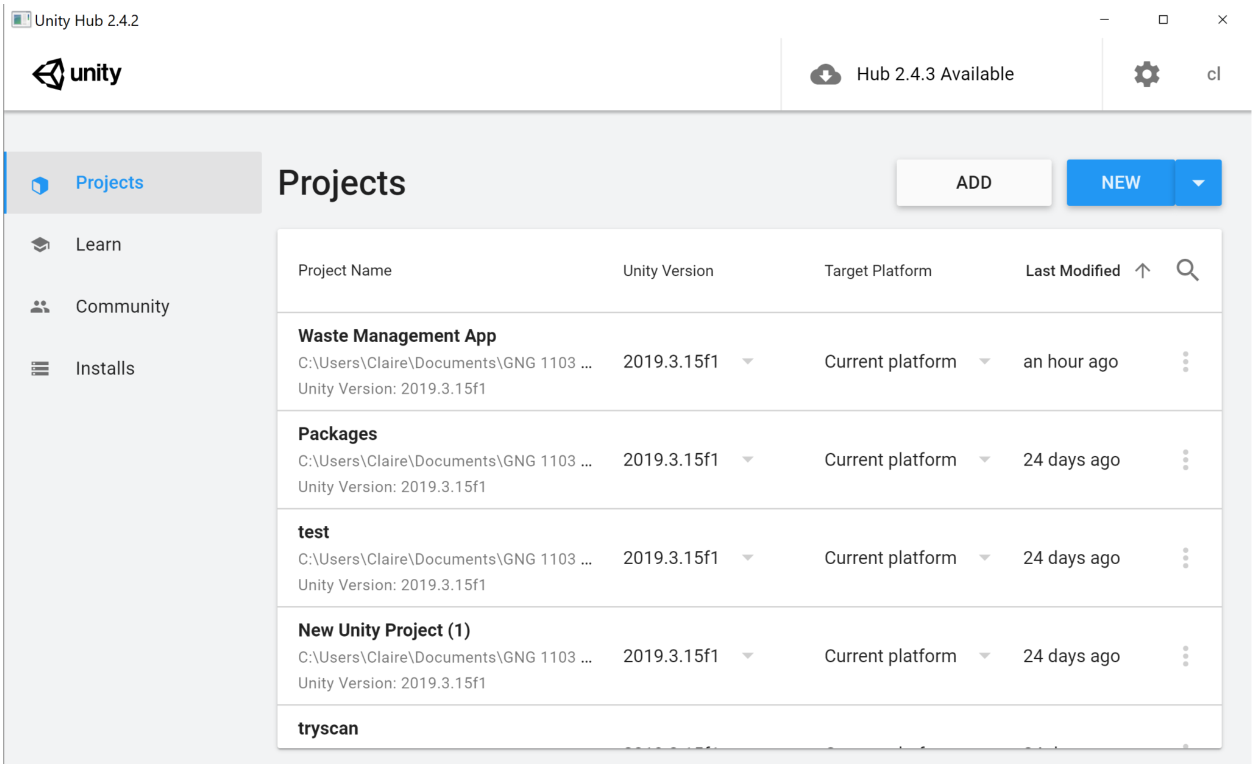
3.2 Preparing to Build the App

The app was created using Unity, Vuforia, and Microsoft Visual Studio. To build this app Unity and Microsoft Visual Studio must be installed on the builders computer. The instructions on how to download these application can be found in the following links:

- https://docs.unity3d.com/560/Documentation/Manual/InstallingUnity.html
- https://visualstudio.microsoft.com/downloads/

The user should download the latest version of these applications to their computer.

Once Unity is installed the user should open Unity(figure 9) which should show the user their projects, which should be empty. From here, the user should click on installs and then the three buttons in the top right corner of the latest version of Unity(figure 10).



Project Name	Unity Version	Target Platform	Last Modified
Waste Management App C:\Users\Claire\Documents\GNG 1103 ... Unity Version: 2019.3.15f1	2019.3.15f1	Current platform	an hour ago
Packages C:\Users\Claire\Documents\GNG 1103 ... Unity Version: 2019.3.15f1	2019.3.15f1	Current platform	24 days ago
test C:\Users\Claire\Documents\GNG 1103 ... Unity Version: 2019.3.15f1	2019.3.15f1	Current platform	24 days ago
New Unity Project (1) C:\Users\Claire\Documents\GNG 1103 ... Unity Version: 2019.3.15f1	2019.3.15f1	Current platform	24 days ago
tryscan			

Figure 9. Unity Open

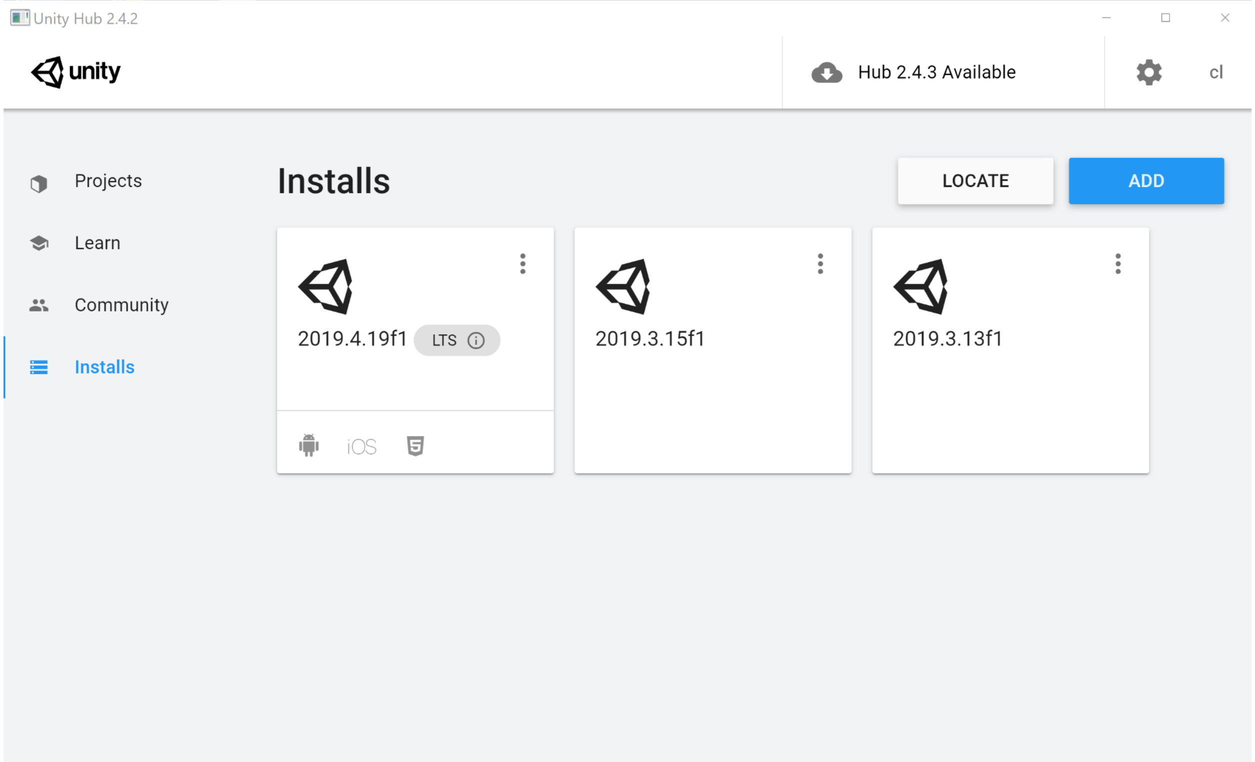
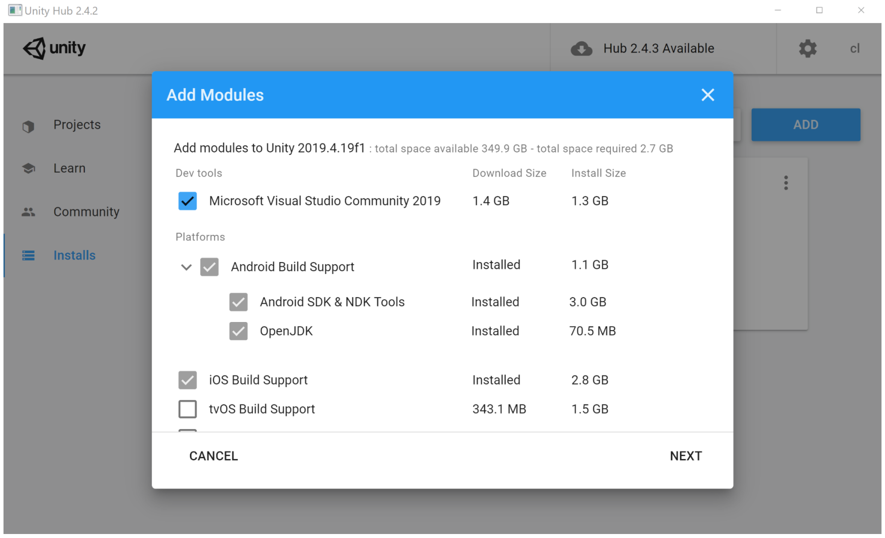


Figure 10. Unity Installs

After selecting the three dots, a drop down will appear. Click on ‘add Modules’, then make sure Microsoft Visual Studio is selected under ‘Dev tools’, and under ‘Platforms’ you select ‘Android Build Support’ and

‘IOS Build Support’(figure 11). Then click ‘Next’ and follow the prompts to install the modules to your Unity.



Add Modules

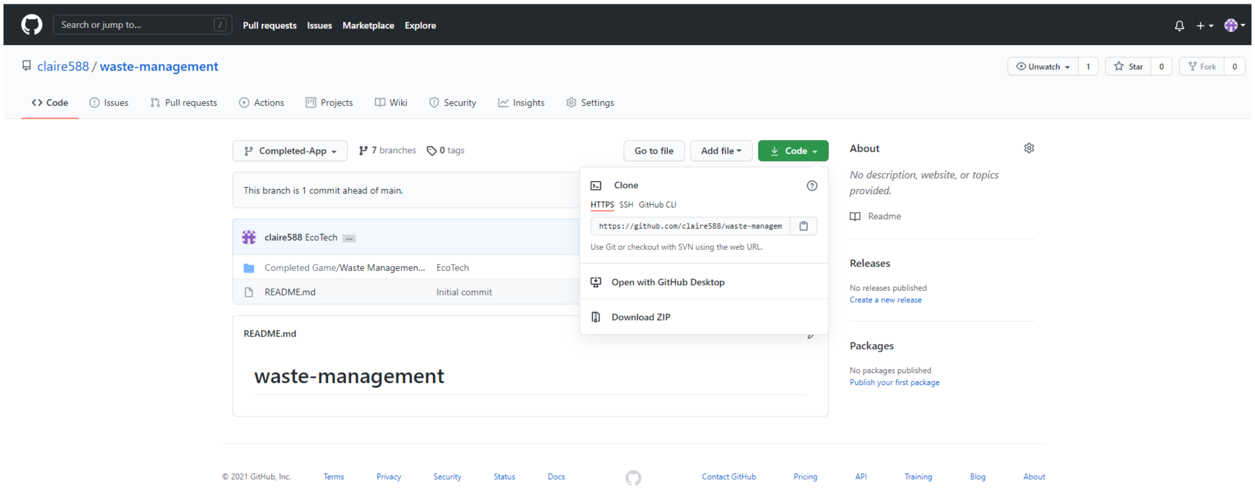
Add modules to Unity 2019.4.19f1 : total space available 349.9 GB - total space required 2.7 GB

Dev tools	Download Size	Install Size
☑ Microsoft Visual Studio Community 2019	1.4 GB	1.3 GB
Platforms		
☑ Android Build Support	Installed	1.1 GB
☑ Android SDK & NDK Tools	Installed	3.0 GB
☑ OpenJDK	Installed	70.5 MB
☑ iOS Build Support	Installed	2.8 GB
☐ tvOS Build Support	343.1 MB	1.5 GB

Figure 11. Modules to Add

3.3 Downloading the App & Adding the App to Unity

To get access to the apps files you will need access to the internet and the computer that Unity has been installed on. Github will be used to download the app’s files and Unity Hub will be used to open those files. To download the app, the user will use the following link to open Github in their browser, https://github.com/claire588/waste-management/tree/Completed-App. This will open to our repository, you will want to download the files under ‘Completed-App’(figure 12).



This branch is 1 commit ahead of main.

claire588 EcoTech

Completed Game/Waste Managemen...	EcoTech
README.md	Initial commit

`https://github.com/claire588/waste-managem`

Use Git or checkout with SVN using the web URL.

No description, website, or topics provided.

No releases published

No packages published

README.md

waste-management

Figure 12. Github Link

To download the files click on the green ‘Code’ button. A drop down menu will appear and you will want to click ‘download ZIP’. Select where you want to download the zipped files, you will need to know where they are located. Once they are done downloading, unzip the files by opening your file explorer and locate the files. Open the zipped file and there should be a unzipped one called ‘waste-management-Completed-App’, unzip this file by moving it to another section of your file explorer.

Now, open Unity Hub. Once open click the ‘ADD’ button in the top right corner. It will open your file explorer where you can locate the file you just unzipped. You will want to double click the file to open it further to a file named ‘Completed Game’, by double clicking that file you will be left with a folder named ‘Waste Management App’ click the folder once then click select folder at the bottom right of the screen. Once the files are done and Unity is now open(figure 13).

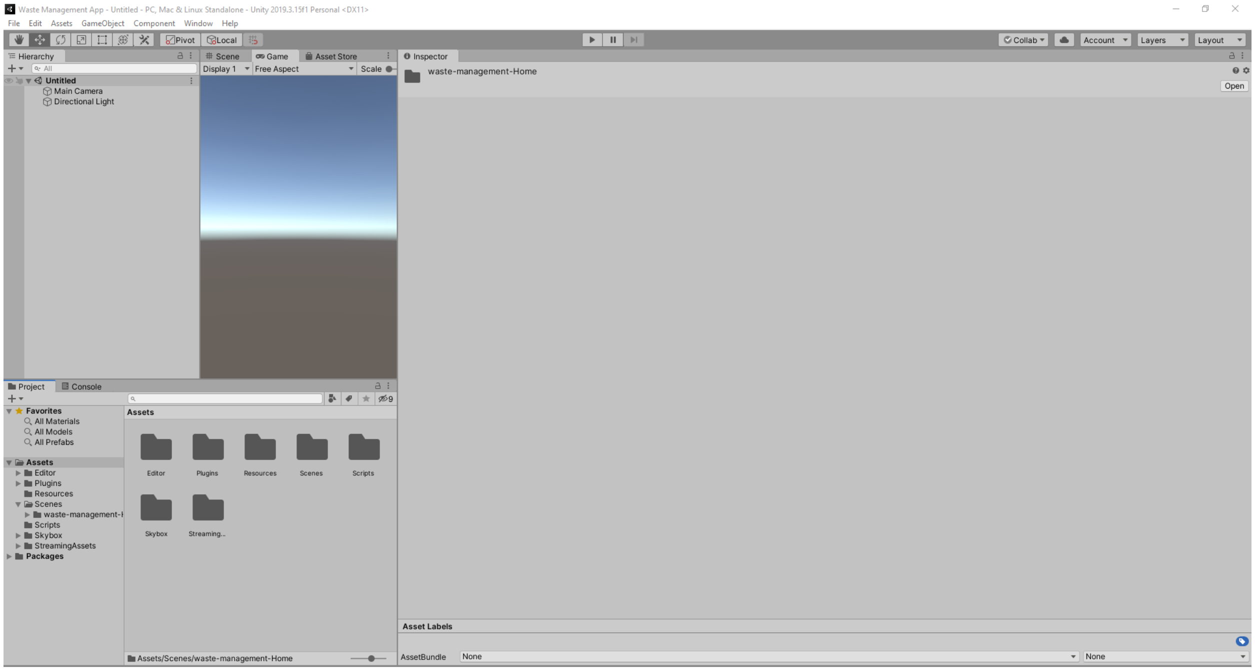


Figure 13. Waste App Open in Unity

Finally, click the folder named ‘Scenes’, then open the folder ‘waste-management-Home’, you will now see all the scenes, code files, and images used in the creation of the app(figure 14), to change to one of the scenes in the app simply double click one of the scenes.

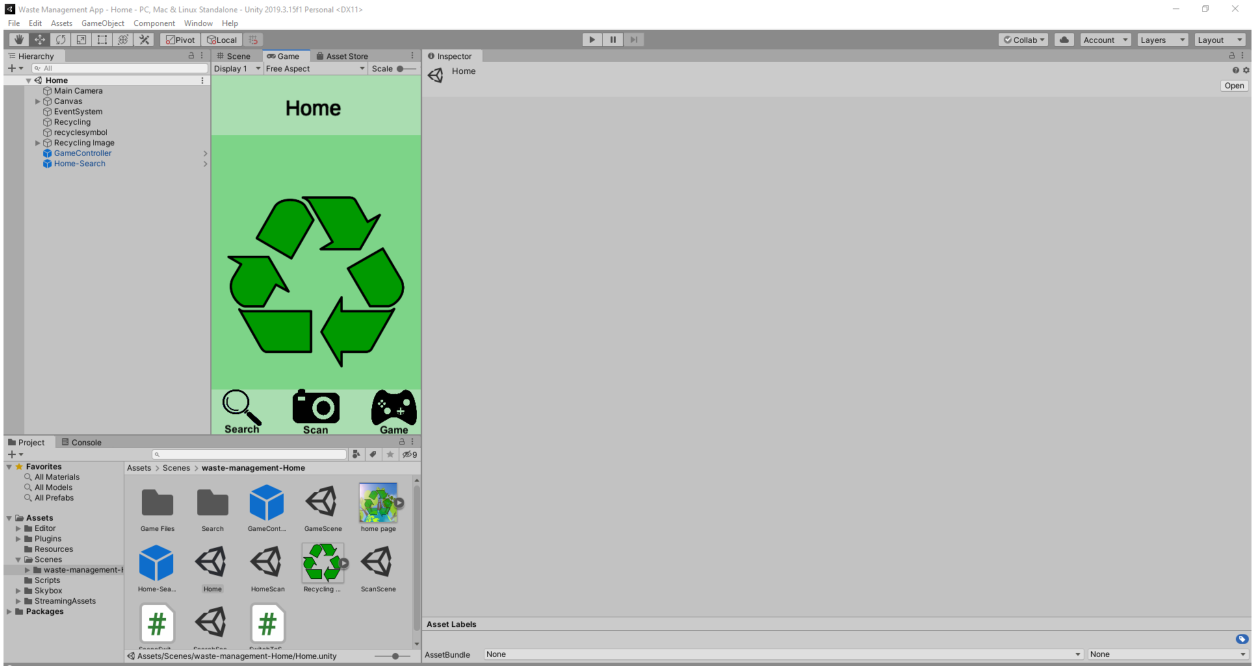
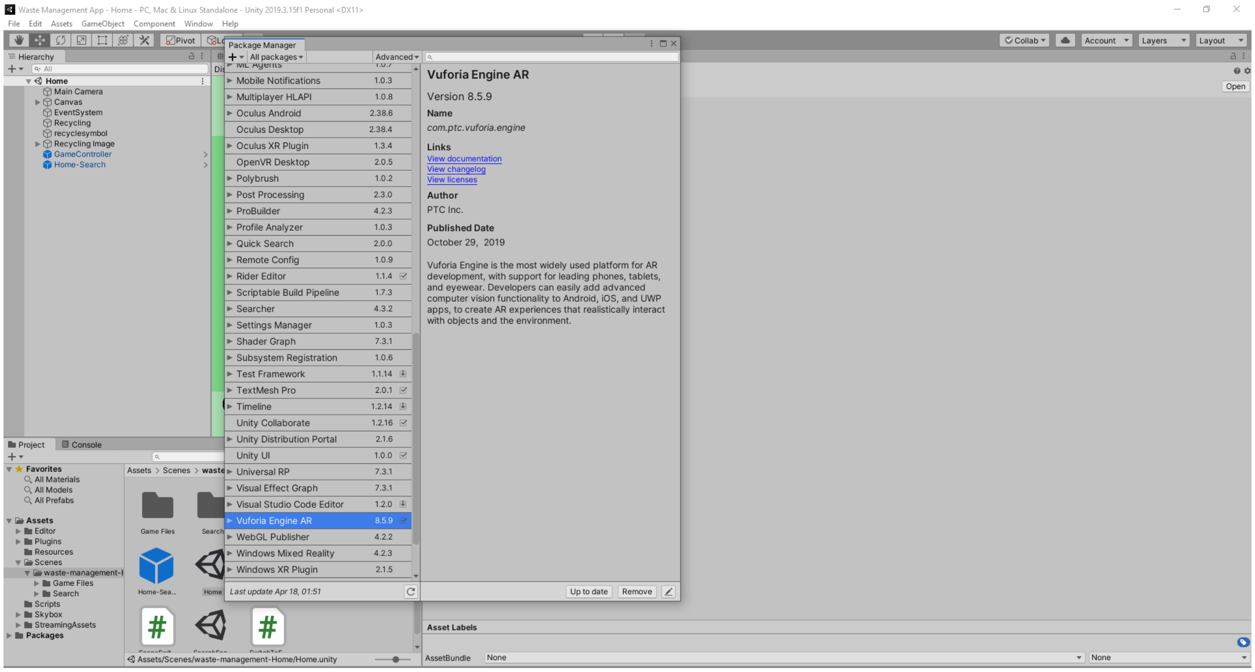


Figure 14. App Files in Unity

All app scenes should be fully functional when downloaded and can be tested by clicking on the triangular play button in the top middle of the screen, and clicking through the app.

If the scan scene is not functional or Unity won’t let you “play” then the Vuforia Plugin will need to be added. To do this select ‘Windows’ at the top, then ‘Windows Manager’ from the drop down. After a new window should open(figure 15), scroll on the left of the window till ‘Vuforia Engine AR’ appears, click it and install it to your Unity.



Vuforia Engine AR

Version 8.5.9

Name
com.ptc.vuforia.engine

Author
PTC Inc.

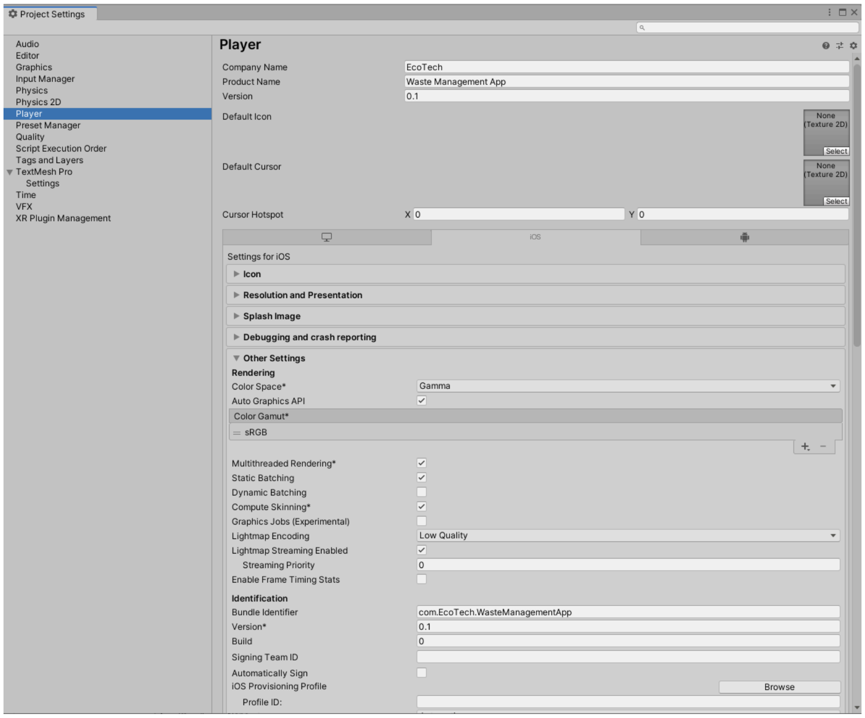
Published Date
October 29, 2019

Vuforia Engine is the most widely used platform for AR development, with support for leading phones, tablets, and eyewear. Developers can easily add advanced computer vision functionality to Android, iOS, and UWP apps, to create AR experiences that realistically interact with objects and the environment.

Figure 15. Installing Vuforia Engine

3.4 Preparing the App for Use

Now that the app is downloaded, you can prepare the settings so you can export it to use. By going to the top left corner of the screen you can see 'Edit' click on that and a drop down will appear. Now click on 'Project Settings …' near the bottom of the list. A new window will open(figure 16).



Company Name: EcoTech
Product Name: Waste Management App
Version: 0.1
Default Icon: None (Texture 2D)
Default Cursor: None (Texture 2D)
Cursor Hotspot: X 0 Y 0

Settings for iOS

Other Settings

Rendering
Color Space*: Gamma
Auto Graphics API: ☑
Color Gamut*: sRGB
Multithreaded Rendering*: ☑
Static Batching: ☑
Dynamic Batching: ☐
Compute Skinning*: ☑
Graphics Jobs (Experimental): ☐
Lightmap Encoding: Low Quality
Lightmap Streaming Enabled: ☑
Streaming Priority: 0
Enable Frame Timing Stats: ☐

Identification
Bundle Identifier: com.EcoTech.WasteManagementApp
Version*: 0.1
Build: 0
Signing Team ID:
Automatically Sign: ☐
iOS Provisioning Profile
Profile ID:

Figure 16. Project Settings Window

Click on the 'Player' on the left, then select either IOS or Android in the 'Other Settings'.

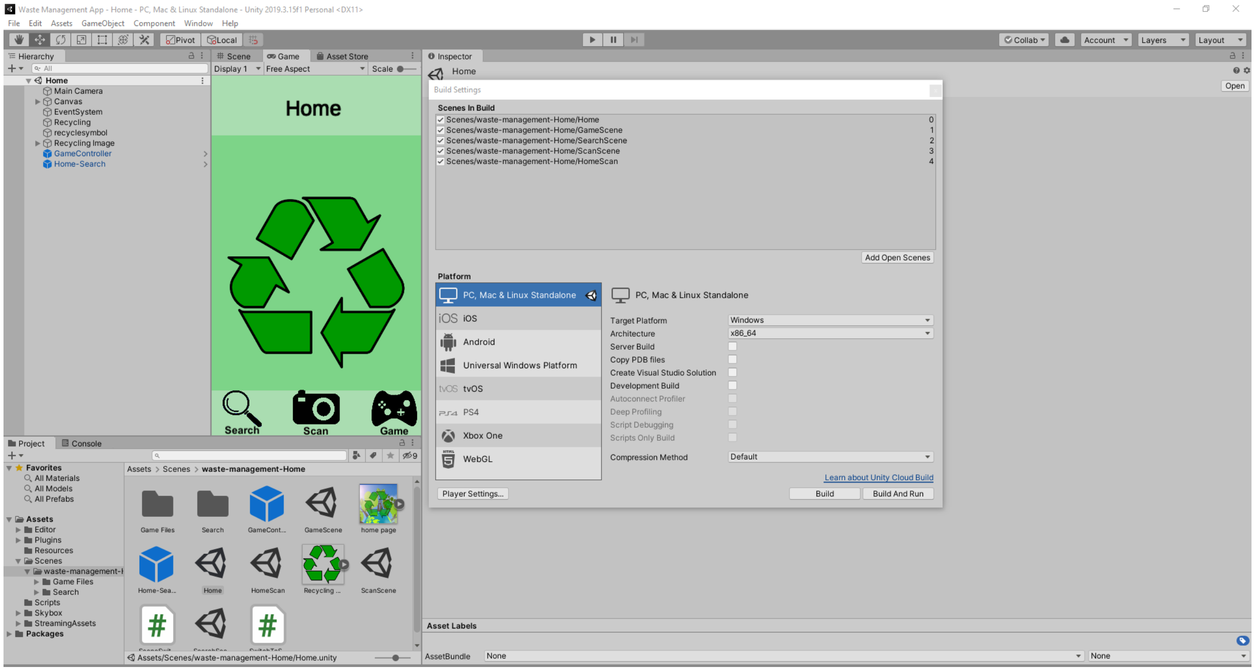
IOS Settings:

1. You can change the 'Company Name' to whatever fits your needs.
2. Next you will scroll down to the 'Requires ARKit support' and do the following:
    a. Change the 'Target minimum IOS Version' to 11.0
3. Then go to 'Architecture' and select 'ARM64' from the drop down.

Android Settings:

1. You can change the 'Company Name' to whatever fits your needs.
2. Uncheck the 'Multithreaded Rendering' check box.
3. Under 'Graphics APIs' make sure both 'Vulkan' and 'OpenGLES3' are added.
4. After ARcore is only supported on Android 7.0 or later. So under the 'Minimum API Level' select Android 7.0 'Nougat' using the drop down.

After closing the settings window, click on 'File' at the top left of the main screen, then open 'Build Settings'. Make sure all the scenes are under the 'Scenes in Build' if one is missing simply open the missing scene in the main Unity window, and then return to the build window and 'Add Open Scenes'. Now, you can choose whether you want to build the app for Android or IOS by switching 'Platforms', then you can build out the game by clicking 'Build and Run'(figure 17).



Build Settings

Scenes In Build

☑ Scenes/waste-management-Home/Home 0
☑ Scenes/waste-management-Home/GameScene 1
☑ Scenes/waste-management-Home/SearchScene 2
☑ Scenes/waste-management-Home/ScanScene 3
☑ Scenes/waste-management-Home/HomeScan 4

Platform: PC, Mac & Linux Standalone

Target Platform: Windows
Architecture: x86_64
Server Build ☐
Copy PDB files ☐
Create Visual Studio Solution ☐
Development Build ☐
Autoconnect Profiler ☐
Deep Profiling ☐
Script Debugging ☐
Scripts Only Build ☐
Compression Method: Default

Figure 17. Build Settings

3.5 User Access Considerations

This application is designed for a user with a mobile smartdevice, specifically smartphones, and has a basic understanding of the current waste management system being used. The app is designed to have a simple, easy to navigate layout for users of all ages to use. Some restrictions are that the user's phone must be running on a more recent version, Android 7.0 and IOS 11.0. As well the device should have a camera or a main part of the application will not be functional for the user.

3.6 Accessing the System

There is no account, password, or any other form of user login system to use the app. Simply opening the app will take the user directly to the home screen where they can use the three bottom buttons(figure 18) to navigate through the screens to where they would like to go.

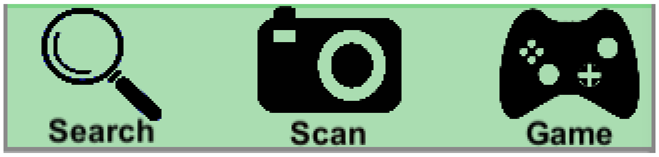


Figure 18. Application's Button Controls

3.7 System Organization and Navigation

The systems organization and navigation mainly revolves around the three icons at the bottom of the screen which represents each of the different subsystems. In order to access each subsystem, the user must simply tap the icon of the subsystem they are attempting to access. Furthermore, each subsystem has some form of a note or set of instruction that should be followed for proper usage of the application.

3.8 Exiting the System

To exit the application it differs depending on which platform you're using. When the application is running on a smart device you can return to the home page then click on your phones home button, swipe up, or any other way your device closes applications. If the application is running on the computer using Unity make sure to stop play but clicking on the play button to stop, simply close Unity with the 'X' in the top right corner.

4. Using the System

The following subsections provide detailed, step-by-step instructions on how to use the various functions or features of EcoTech.

4.1 Home Screen

When the application is opened, the user will be greeted with the home screen. The only purpose of this screen is to redirect the user to the different subsystems of the application. At the bottom of the screen there are three buttons that will take the user through the app.

4.1.1 Search Button, Left Button

This button takes the user to the search screen. In this new scene the user will be welcomed with a title of which subsystem they entered, a 'Go to Website' button, a home button, and a warning of what the main buttons function is. The warning lets the user know what will happen when the search button is pressed.

4.1.1.1 Go to Website Button

This button opens the users browser and takes them to the City of Ottawa Waste Explorer website. Once the website is opened the user has to type the name of the item they are looking for and click search to find it.

4.1.1.2 Return Home Button

The button in the bottom right corner will redirect the user back to the home screen when pressed.

4.1.2 Scan Button, Middle Button

This button will switch scenes and open the user's camera. The screen will only show what the user's camera can see. When an item the system recognizes is shown a panel will appear in the a specific colour to which bin the user needs to throw the item in, and a message on how to properly throw the item out will appear with the option to return to the home screen. If the user will like to scan another item, they should remove the other item from the camera's view and now place the new item they would like to look for in the view of the camera.

4.1.2.1 Return Home Button

The scanning scene contains two different return home buttons that redirect the user to the home screen. The first one stays in the bottom right corner of the screen and can be clicked at any time. When the panel appears to let the user know what item it is, another return button appears that is green and in the middle of the screen.

4.1.3 Game Button, Right Button

This button redirects the user to the game home screen, where there are two buttons. One to start the game, 'Sort Your Garbage', in the middle of the screen and the second in the bottom right to let the user return to the home screen.

4.1.3.1 'Sort Your Garbage' Button

Once the button to start the game is pressed, the application will start the game. Four main buttons will appear that can be clicked to answer the question. As well the return button at the bottom right so the user can exit the game at any time before finishing the game.

5. Troubleshooting & Support

5.1 Error Messages or Behaviours

EcoTech in itself does not have any error messages that will appear when using the application.

If an item is not recognized when its being scanned try switching the position of the item a few times, if the item is still not being recognized then it probably is not in the database, you can check items that are in it by opening Unity and selecting the scan scene, on the left there should be a list of items with their names(figure 19).

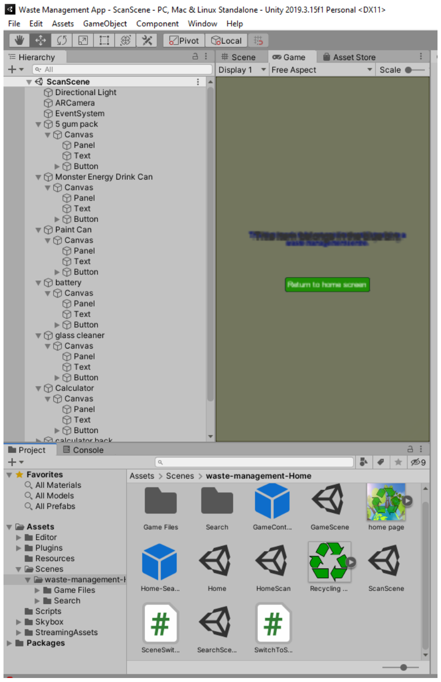
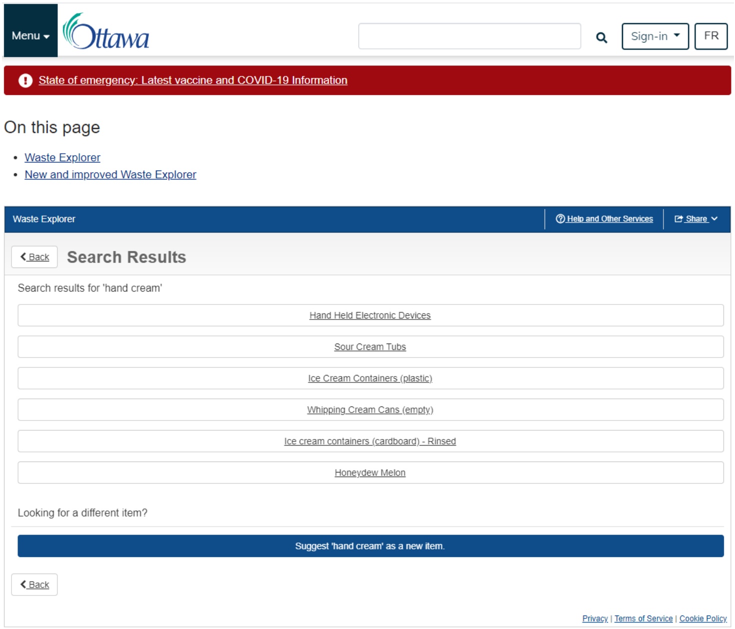


Figure 19. Items in Database

If when searching for an item in the Ottawa Waste Explorer and an item is not recognized, it will try and give you suggestions on what they think you are referring to. If these are not what you're looking for there is a button at the bottom where you can suggest as a new item(figure 20).



State of emergency: Latest vaccine and COVID-19 Information

On this page

- Waste Explorer
- New and improved Waste Explorer

Waste Explorer

Search Results

Search results for 'hand cream'

- Hand Held Electronic Devices
- Sour Cream Tubs
- Ice Cream Containers (plastic)
- Whipping Cream Cans (empty)
- Ice cream containers (cardboard) - Rinsed
- Honeydew Melon

Looking for a different item?

Suggest 'hand cream' as a new item.

Figure 20. Waste Explorer When Item is Unknown

5.2 Maintenance

There is no maintenance that is needed to avoid failure while using this prototype. But to keep the application relevant, items should be added to the scanning database so it will be up to date with new product designs.

5.3 Support

If issues arise or there are any complications with the application, the user can contact the EcoTech developers through email or by contacting the Unity support team. The names and emails addresses of our team members are Claire Christensen at cchri039@uottawa.ca, Tye Segal-Kawano at tsega096@uottawa.ca, Isaiah Philip at iphil098@uottawa.ca, Yufei Zhao at yzhao312@uottawa.ca, and Yang Chen at ychen703@uottawa.ca. For help regarding scanning contact Yufei Zhao, help with the game contact Claire Christensen, and help regarding the search page contact Isaiah Philip. Otherwise for general information or questions any member of the team is happy to help. As well, Unity has a support team found on their website in the following link, https://unity.com/support-services, where you can contact them through chat or email if there are any questions our team cannot answer.

6. Product Documentation

This section will contain how each subsystem of the EcoTech application was built for the final design. This will outline the equipment used and the total costs to create this app.

6.1 App

6.1.1 BOM(Bill of Materials)

Item Name	Purpose	Price (CAD$)
GitHub	Used to assist with coding and sharing files	Free
Unity Add-ons	Used for layout and letter fonts	Free
Google Console Account	To publish the app for users to download off Google Play	$32.27
	Total Cost:	$32.27

6.1.2 Equipment List

Item	Item Description	Unit Price	Amount

Unity	Software to create an application	Free	$0
Vuforia	AR software and database builder	Free	$0
		Total Cost	$0

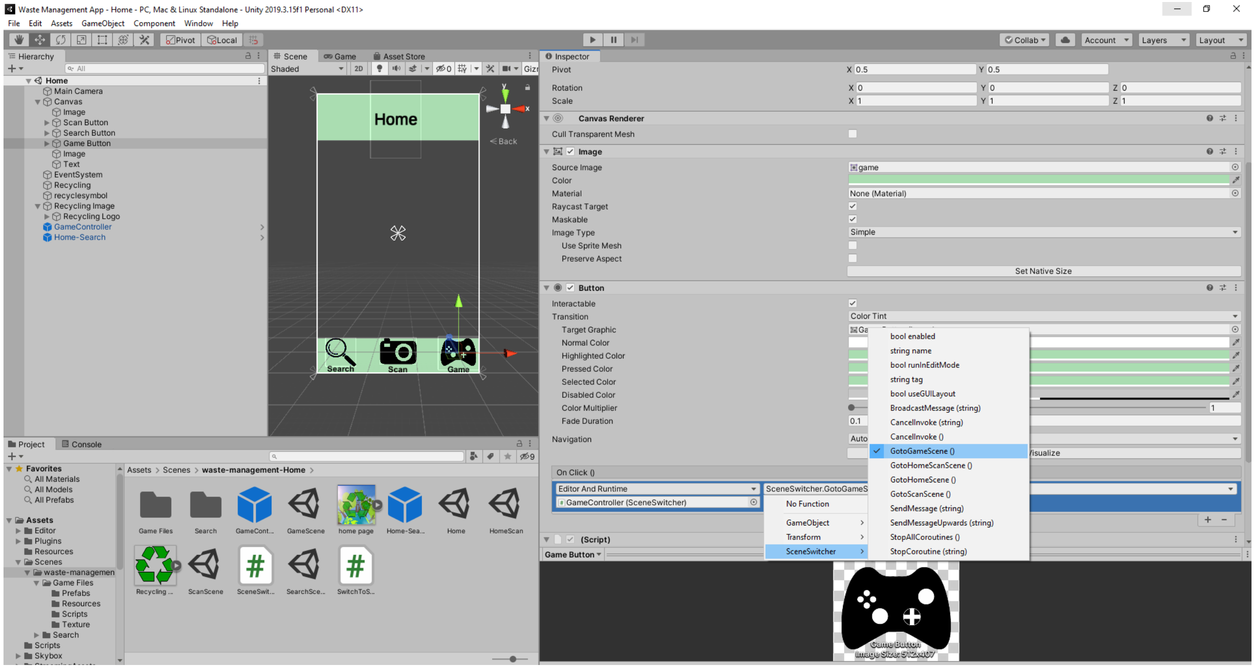
Personal laptops and phones were also used which cost around $1500 to $2000, this price will range depending on version, year, and company of the models you buy.

6.1.3 Instructions

All files, including C# scripts, referenced in this section will be a part of the files that can be downloaded using Github, as mentioned before. After opening the app files in Unity all colours, and placements of objects can be changed by clicking on the object you would like to change on the left hand side, and then adjusting its details in the inspector window on the right hand side.

6.1.3.2 Home Screen

In the files that were downloaded, the buttons on the home screen were images downloaded and turned into Sprite(2D and UI) formats. Then each button was given the C# script GameController (SceneSwitcher) then a scene was selected from the drop down. The selected scene is the one it will switch to when the button is clicked. (Figure 21)

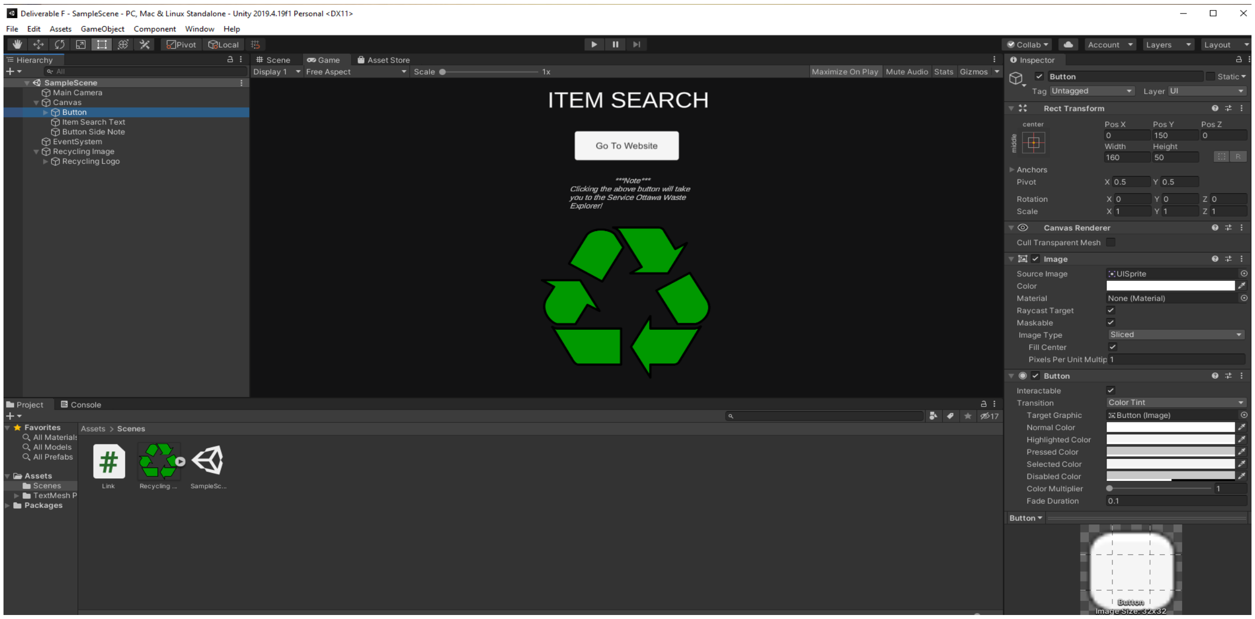


Pivot X 0.5 Y 0.5
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
Cull Transparent Mesh ☐
Source Image: game
Material: None (Material)
Raycast Target ☑
Maskable ☑
Image Type: Simple
Use Sprite Mesh ☐
Preserve Aspect ☐
Interactable ☑
Transition: Color Tint
Color Multiplier: 1
Fade Duration: 0.1
Navigation: Auto
On Click ()
Editor And Runtime
GameController (SceneSwitcher)
SceneSwitcher.GotoGameS

Figure 21. Button Activity

6.1.3.2 Search Screen

To create the search scene, a canvas was added with multiple objects being added to it. Unlike the other subsystems, no specific background was required. In order to reduce the empty space and improve the overall aesthetic of the overall layout of the subsystem, a recycling logo was added to the center of the screen. This was done by selecting the appropriate logo and adding it to the assets folder and then dragging this asset onto the added image in the canvas of the hierarchy window. The image was then scaled and placed in the desired position (Figure 22). Additionally, two sets of text were added to the scene in the hierarchy window. The first set of text simply titled the subsystem and the second set of text provided the user with a warning that clicking the above button will take them to their default web browser (Figure 22).



ITEM SEARCH

Go To Website

Note
Clicking the above button will take you to the Service Ottawa Waste Explorer!

Button
Tag: Untagged
Layer: UI

Rect Transform
center / middle
Pos X: 0 Pos Y: 150 Pos Z: 0
Width: 160 Height: 50
Pivot: X 0.5 Y 0.5
Rotation: X 0 Y 0 Z 0
Scale: X 1 Y 1 Z 1

Canvas Renderer
Cull Transparent Mesh: ☐

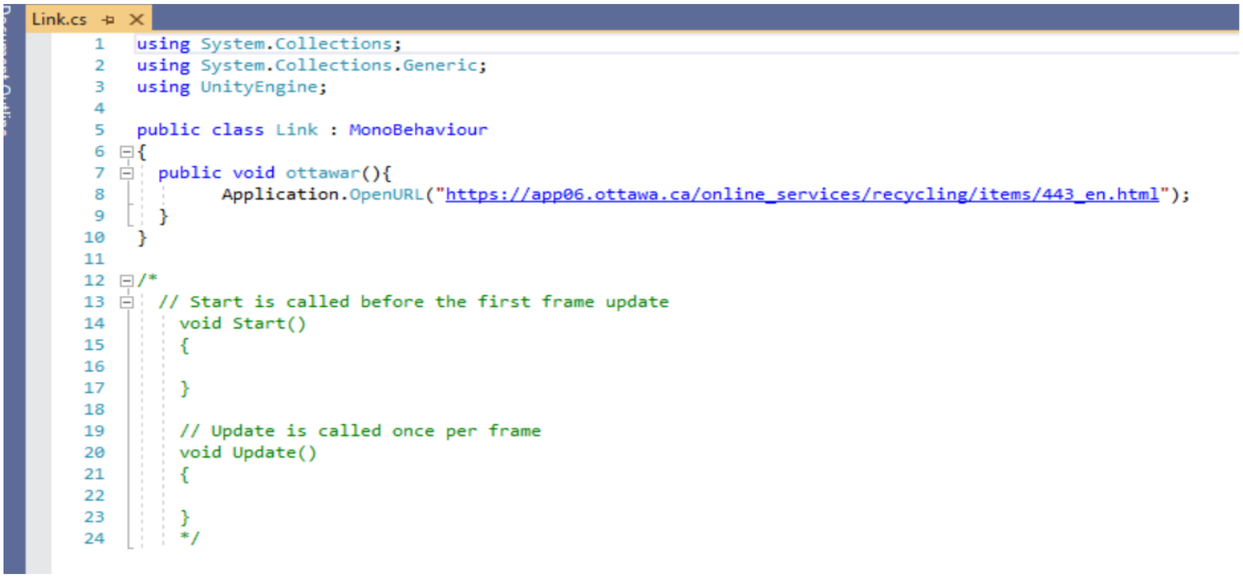
Image
Source Image: UISprite
Material: None (Material)
Raycast Target: ☑
Maskable: ☑
Image Type: Sliced
Fill Center: ☑
Pixels Per Unit Multip: 1

Button
Interactable: ☑
Transition: Color Tint
Target Graphic: Button (Image)
Normal Color
Highlighted Color
Pressed Color
Selected Color
Disabled Color
Color Multiplier: 1
Fade Duration: 0.1

Button
Image Size: 32x32

Figure 22: Layout of Search Scene

Following the image, a button was then added in the hierarchy window with a main purpose to direct the user to the Service Ottawa web page. After the button was created it was then scaled and placed in an appropriate location in the scene (Figure 22). The code for the button was then created by creating a new C# script in the assets folder where the following code was written (Figure 23).



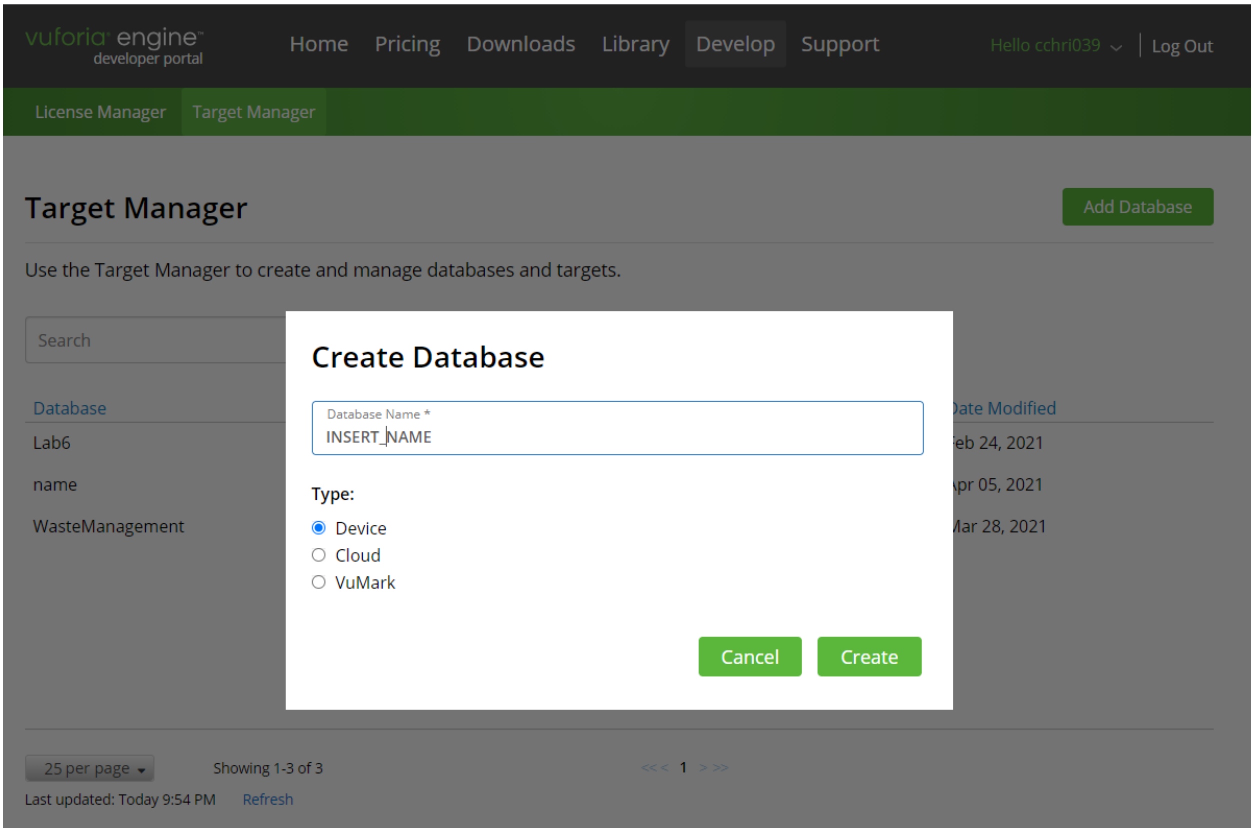
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Link : MonoBehaviour
{
  public void ottawar(){
        Application.OpenURL("https://app06.ottawa.ca/online_services/recycling/items/443_en.html");
  }
}

/*
  // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {

    }
    */
```

Figure 23: Code for Search Button

Once the code was created, the C# code asset was then dragged onto the button in the hierarchy window in order to obtain its full functionality.

6.1.3.3 Scan Screen

The scanning portion all started by creating a Vuforia account, https://developer.vuforia.com/, after an account is created select the ‘Develop’ section at the top of the screen. Next click ‘Target Manager’, now add a new database. Name the database and select type ‘Device’(Figure 24).



Target Manager

Use the Target Manager to create and manage databases and targets.

Database	ate Modified
Lab6	eb 24, 2021
name	pr 05, 2021
WasteManagement	lar 28, 2021

Create Database

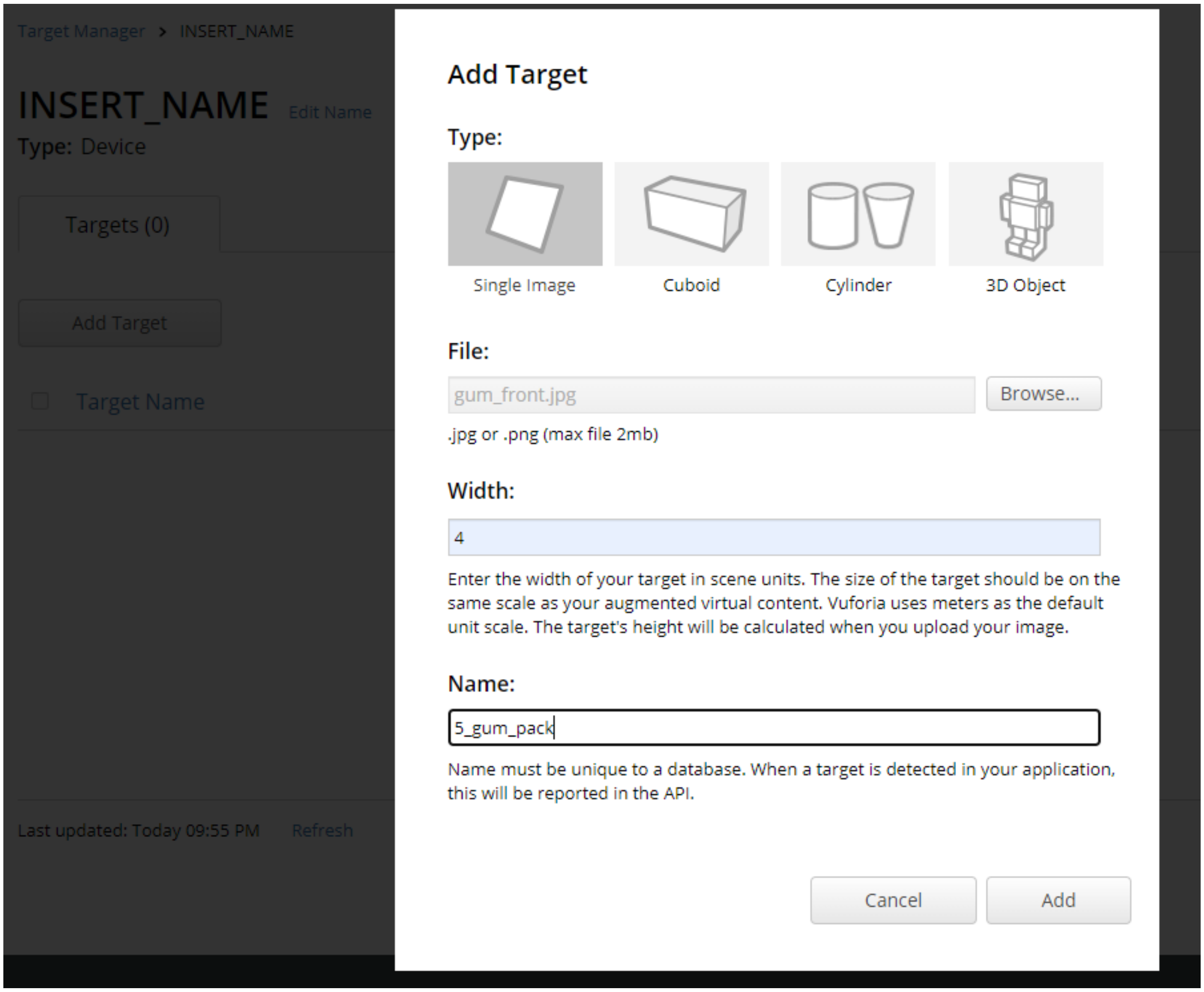
Database Name * INSERT_NAME

Type:

● Device
○ Cloud
○ VuMark

Figure 24. Adding a Database Vuforia

Select the database you just created and add a target, for this you will need a photo of an item. Select a single image, choose which file you want to upload, set width to 4, and name the image(Figure 25).



Type: Device

Last updated: Today 09:55 PM

Add Target

Type: Single Image

File: gum_front.jpg

.jpg or .png (max file 2mb)

Width: 4

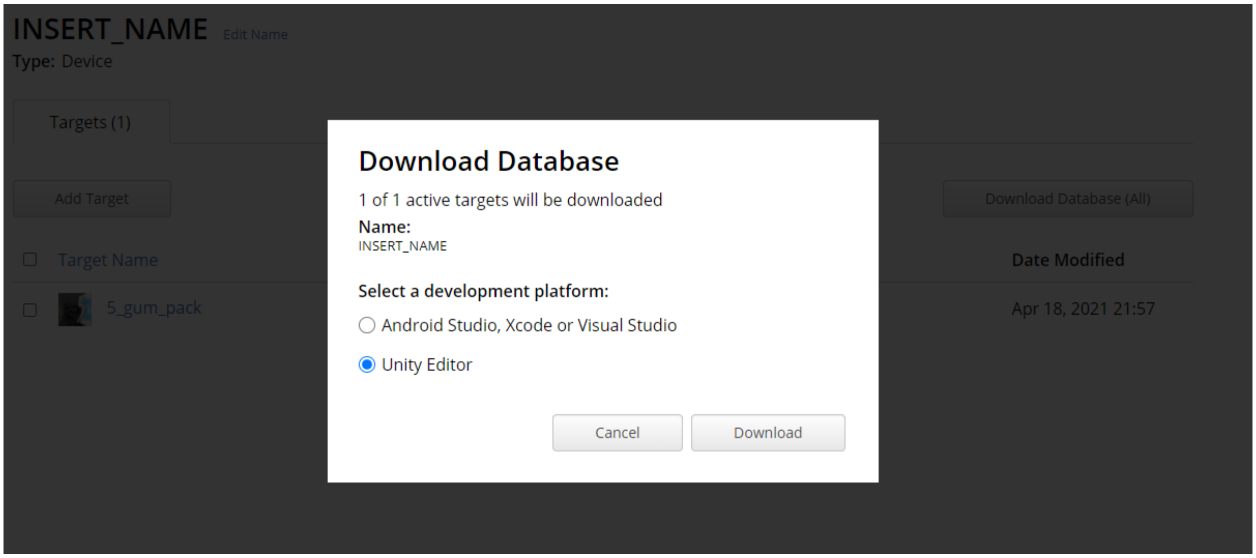
Enter the width of your target in scene units. The size of the target should be on the same scale as your augmented virtual content. Vuforia uses meters as the default unit scale. The target's height will be calculated when you upload your image.

Name: 5_gum_pack

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

Figure 25. Adding a Target Vuforia

Make sure your items have a rating of 4 or greater, and add as many items as you need. Once your finished adding to the database download the database, and make sure to select Unity Editor as the development platform(Figure 26).



Download Database

1 of 1 active targets will be downloaded

Name:
INSERT_NAME

Select a development platform:

○ Android Studio, Xcode or Visual Studio

● Unity Editor

Figure 26. Downloading the Database

Next, reopen Unity into the project you are working on. Make sure Vuforia is added to your Unity by selecting the Window, and selecting it on the drop down. Now, right click the hierarchy, and on the drop down menu hover over Vuforia Engine till another drop down menu appears, then select AR Camera. Then again do the same process but instead of adding an AR Camera, add an image. Click on the image you just created in the hierarchy and in the inspector window add the database you downloaded, by clicking on Database. Then choose which image you want to use by changing the image target right underneath(Figure 27).

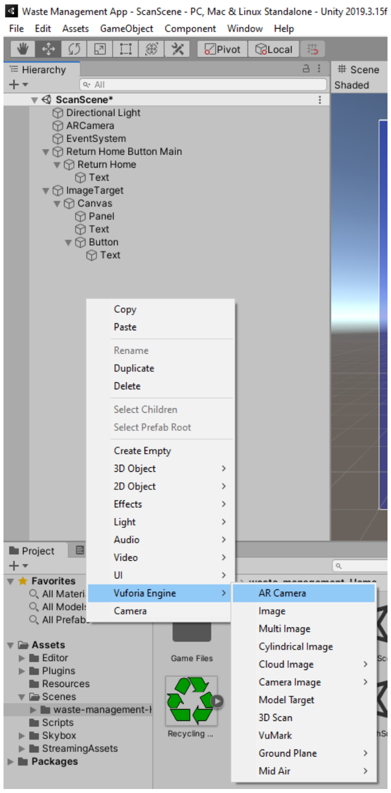
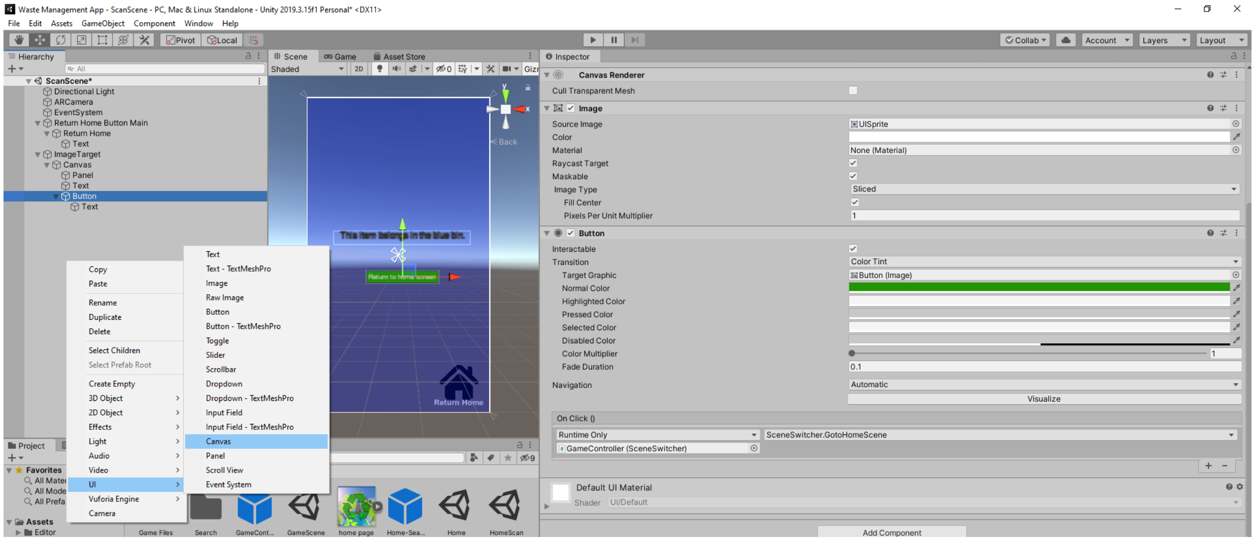


Figure 27. Adding Vuforia to Unity

Once that’s done create a canvas and place a panel, text box, and button, and place them all within the image target you had created. Now change the panel colour to the colour of the bin the item goes in. Then in the text box you created type a message to tell the user on where they should place the item. Lastly, add the text ‘Return Home’ to the created button and link the C# script SceneSwitch and change the scene to ‘HomeScene’. Lastly, to finish the scan screen add a separate button outside of the image target, and follow the same instructions as the button on the canvas, linking the button to the home screen and calling it the return home button(Figure 28).


Waste Management App - ScanScene - PC, Mac & Linux Standalone - Unity 2019.3.15f1 Personal* <DX11>

Canvas Renderer
Cull Transparent Mesh ☐

Image
Source Image: UISprite
Material: None (Material)
Raycast Target ☑
Maskable ☑
Image Type: Sliced
Fill Center ☑
Pixels Per Unit Multiplier: 1

Button
Interactable ☑
Transition: Color Tint
Target Graphic: Button (Image)
Color Multiplier: 1
Fade Duration: 0.1
Navigation: Automatic

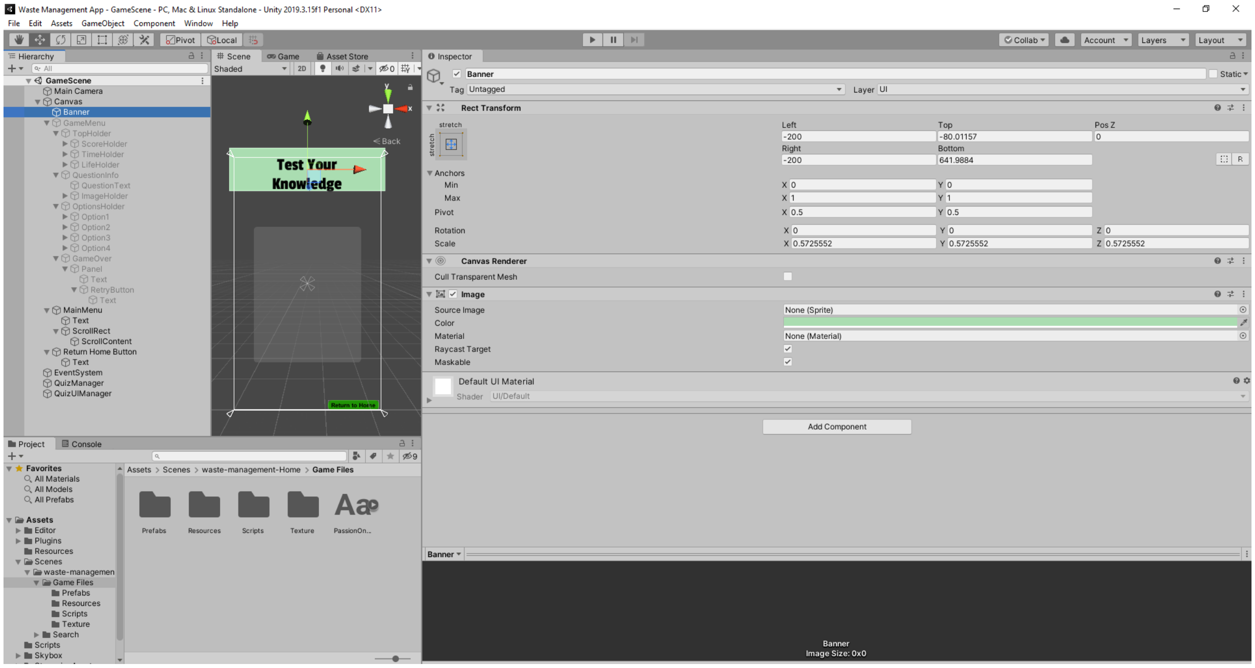
On Click ()
Runtime Only — SceneSwitcher.GotoHomeScene
GameController (SceneSwitcher)

Default UI Material
Shader: UI/Default

Figure 28. Adding Pop-up Display

6.1.3.4 Game Screen

To create the game scene a canvas was created with multiple objects added to it. To start building the game download the game files from the EcoTech Github repository called ‘Game’ or ‘Completed-App’. Once that is done, open the files in Unity explained in an earlier part of this report. Once the files are open it should look like Figure 29, with multiple different labeled objects along the side. Once an object is clicked its inspection details will be shown on the right in the inspector window.

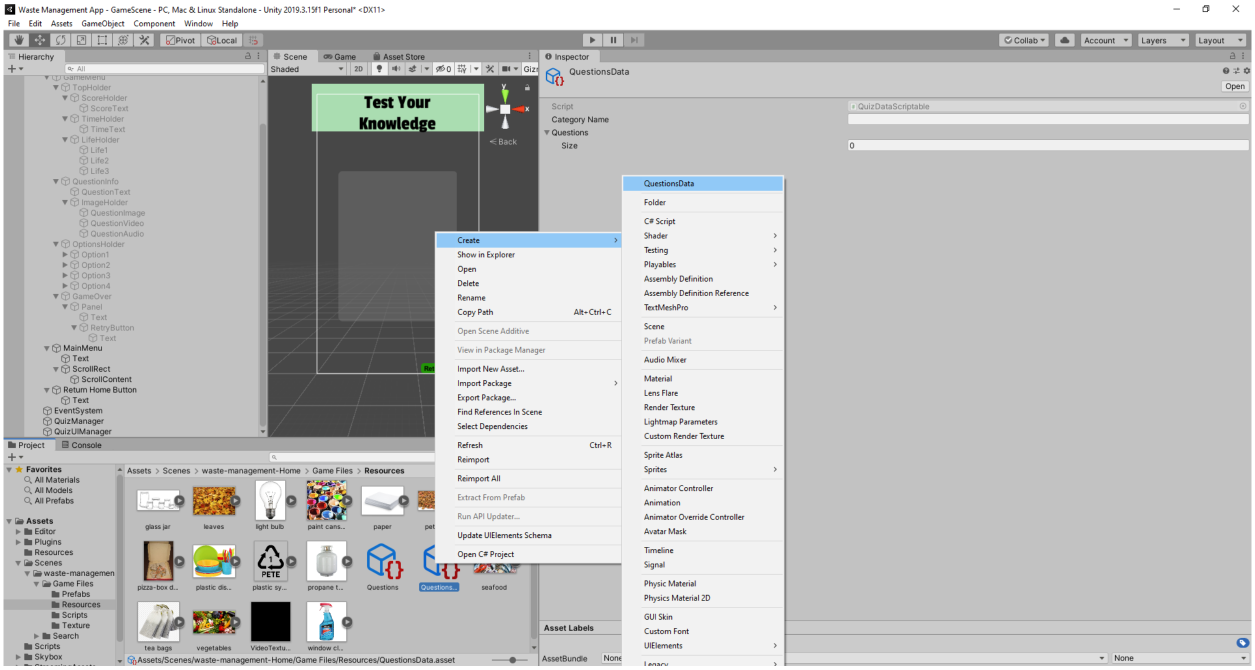


Banner
Tag Untagged Layer UI
Left -200 Top -80.01157 Pos Z 0
Right -200 Bottom 641.9884
Anchors Min X 0 Y 0
Max X 1 Y 1
Pivot X 0.5 Y 0.5
Rotation X 0 Y 0 Z 0
Scale X 0.5725552 Y 0.5725552 Z 0.5725552
Source Image None (Sprite)
Material None (Material)
Raycast Target ☑
Maskable ☑
Default UI Material
Shader UI/Default

Figure 29. Game Files

The files from Github already come with everything already to go, colours and sizes are of fonts, objects, and buttons are all at the correct size. They can be changed by selecting the part you would like to change, then you can adjust the size in ‘Rect Transform’ and colour in ‘Image’. All fonts should be preset to ‘PassionOne-Regular’ but can be changed by selecting the font under ‘Character’.

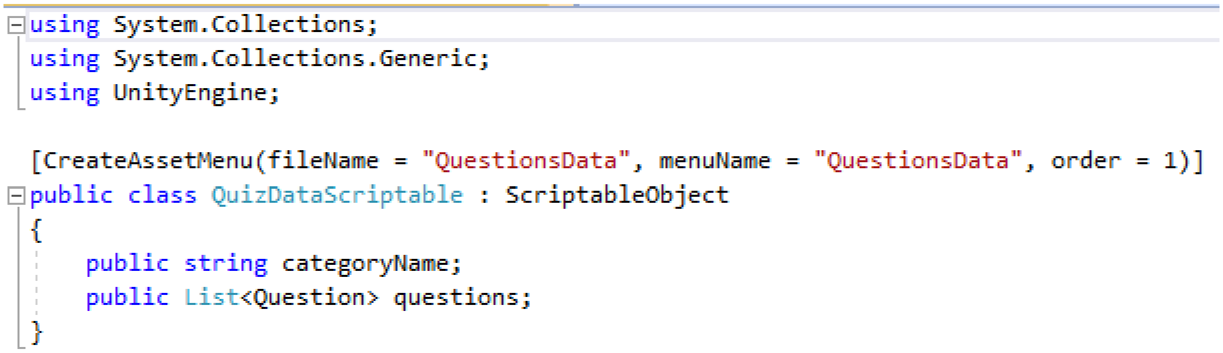
The question list can be found in the Resources section, or a new one can be created by right clicking the project window and creating a new ‘QuestionData’ found near the top(Figure 30). After type a ‘Category Name’ to call your questions and add the C# script named ‘QuizDataScriptable’(Figure 31).



Script QuizDataScriptable
Category Name
Questions
Size 0

Assets/Scenes/waste-management-Home/Game Files/Resources/QuestionsData.asset

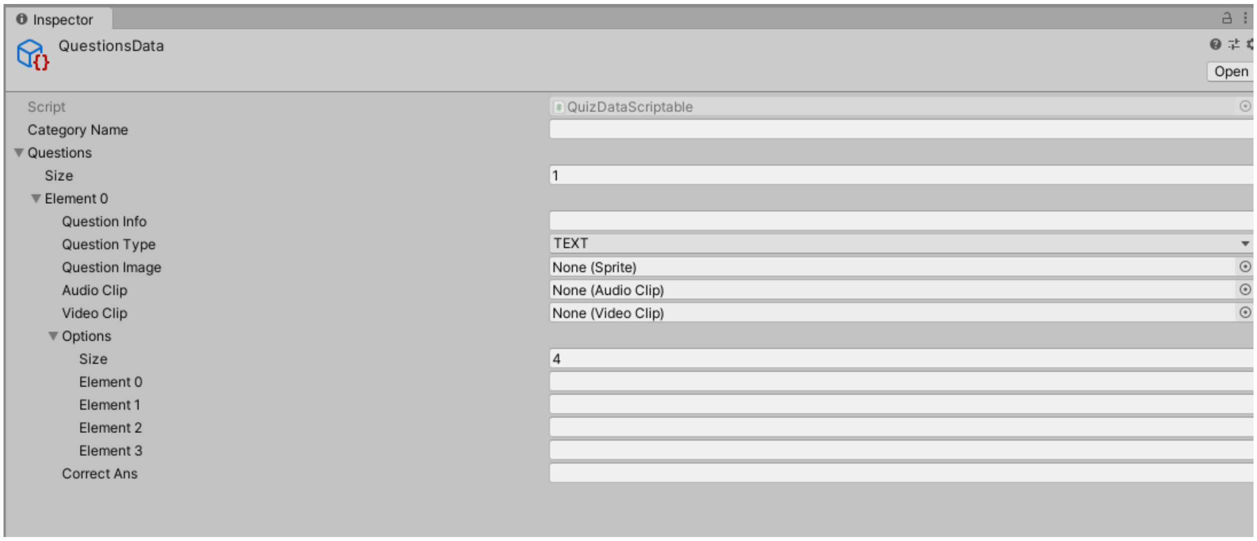
Figure 30. Question Data



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

[CreateAssetMenu(fileName = "QuestionsData", menuName = "QuestionsData", order = 1)]
public class QuizDataScriptable : ScriptableObject
{
    public string categoryName;
    public List<Question> questions;
}
```

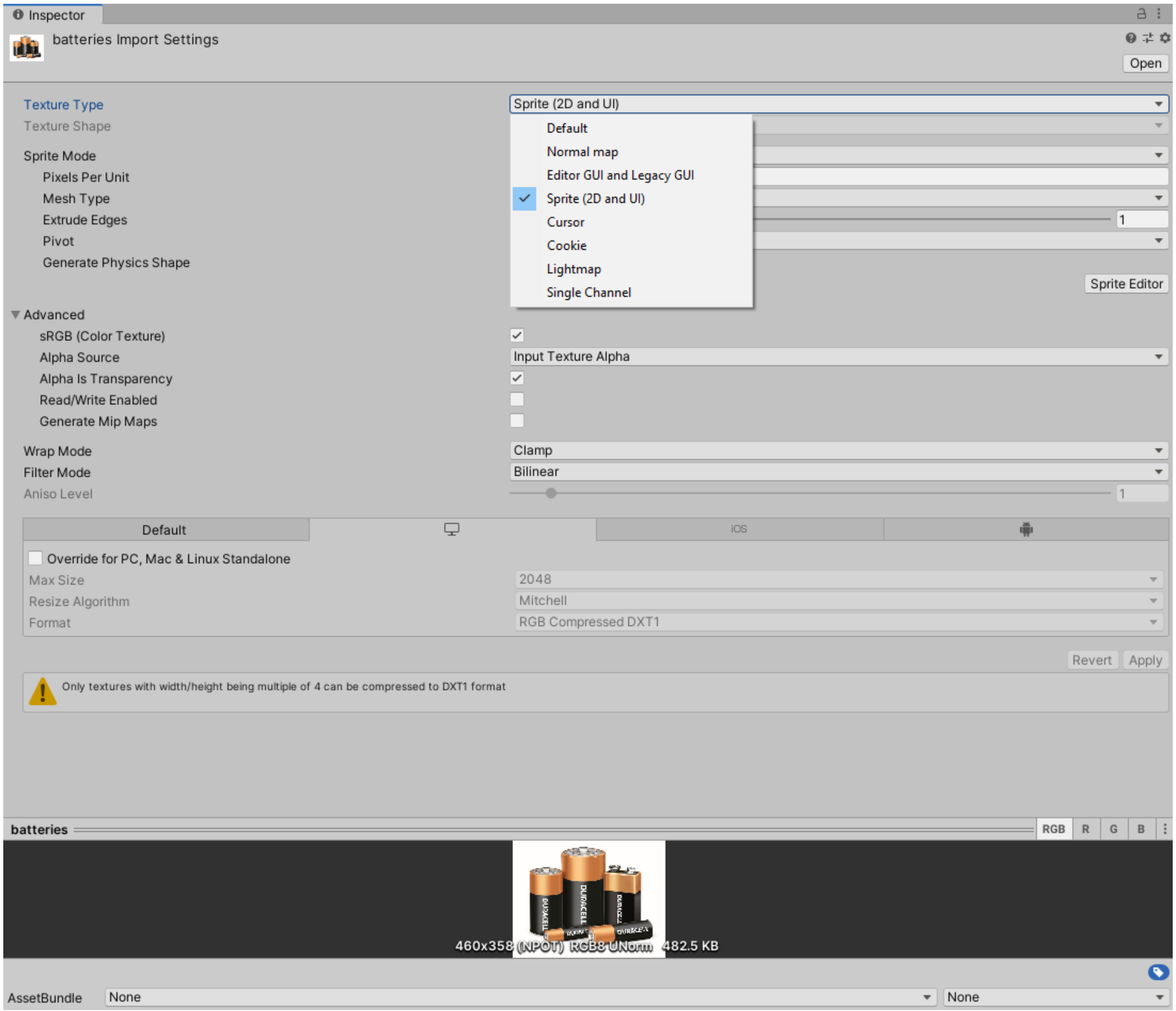
Figure 31. QuizDataScriptable Code

After this set your size to the number of questions you would like to have in your database. For each question type the question in ‘Question Info’, select your question type if you want a video, image, or just a text question. Type in the number of options your question should have and fill out the element slots with possible answers for the question. Type the correct answer in the ‘Correct Ans’ section and make sure it matches with at least one of the options elements(Figure 32). If an image is being added to a question import the image into Unity and change the texture type to ‘Sprite (2D and UI) and then click apply(Figure 33).



Inspector
QuestionsData
Script: QuizDataScriptable
Category Name:
Questions
Size: 1
Element 0
Question Info:
Question Type: TEXT
Question Image: None (Sprite)
Audio Clip: None (Audio Clip)
Video Clip: None (Video Clip)
Options
Size: 4
Element 0:
Element 1:
Element 2:
Element 3:
Correct Ans:

Figure 32. Question Options



batteries Import Settings

Texture Type	Sprite (2D and UI)
Texture Shape	
Sprite Mode	
Pixels Per Unit	
Mesh Type	
Extrude Edges	1
Pivot	
Generate Physics Shape	
Advanced	
sRGB (Color Texture)	☑
Alpha Source	Input Texture Alpha
Alpha Is Transparency	☑
Read/Write Enabled	☐
Generate Mip Maps	☐
Wrap Mode	Clamp
Filter Mode	Bilinear
Aniso Level	1

Default

☐ Override for PC, Mac & Linux Standalone

Max Size	2048
Resize Algorithm	Mitchell
Format	RGB Compressed DXT1

Only textures with width/height being multiple of 4 can be compressed to DXT1 format

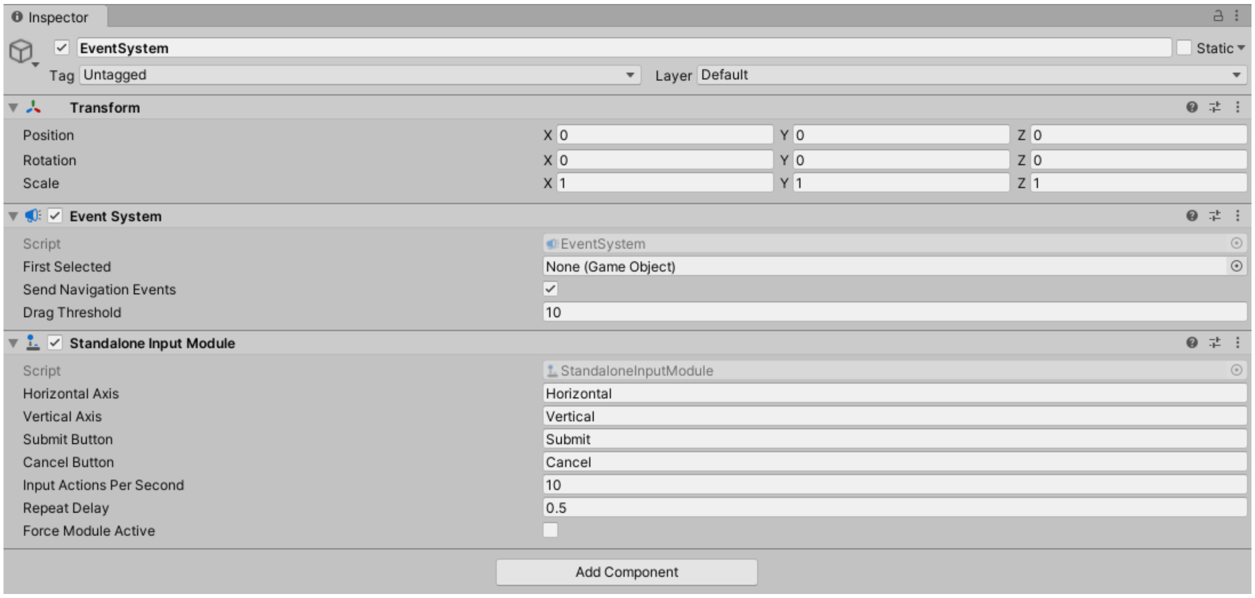
batteries

460x358 (NPOT) RGB8 UNorm 482.5 KB

AssetBundle None None

Figure 33. Changing Image to Correct Format

Now that you have questions and a setup for them, there are 3 objects used to run the game. The first one EventSystem uses the 'StandaloneInputModule' and 'EventSystem' scripts to regulate when things happen, delays between when a button is pressed and the next event that follows(Figure 34). In QuizManager it uses the objects from QuizUIManager and QuizManager scripts, then adds on top of them how many different games can be changed by size number, and how much time the game is given to be completed in seconds(Figure 35). Lastly QuizUIManager uses the CategoryBtn, QuizManager, and ShuffleList scripts. These allow the game to know which questions it should be usings, that it should be using the QuizManager object, and that it should shuffle the questions every time the game is started(Figure 36).



☑ EventSystem
Tag Untagged
Layer Default

Transform

Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

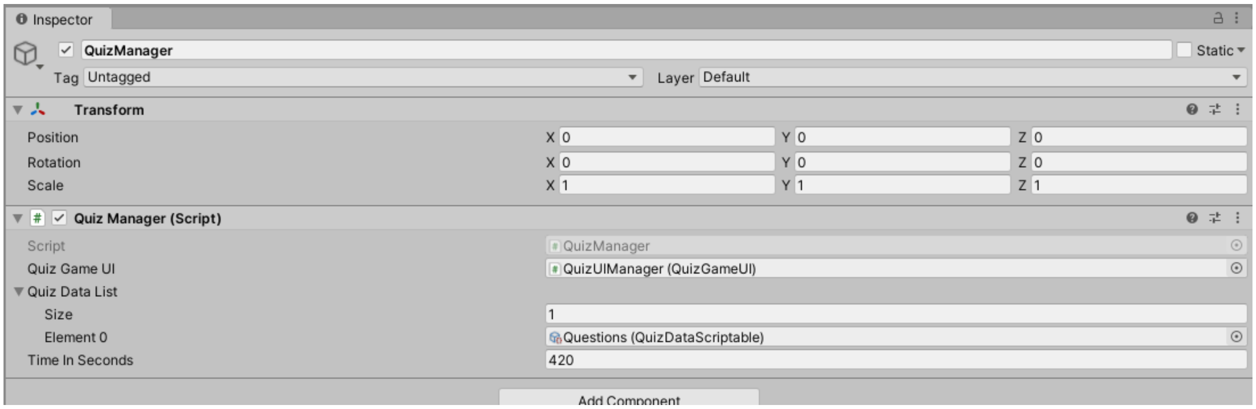
☑ **Event System**

Script EventSystem
First Selected None (Game Object)
Send Navigation Events ☑
Drag Threshold 10

☑ **Standalone Input Module**

Script StandaloneInputModule
Horizontal Axis Horizontal
Vertical Axis Vertical
Submit Button Submit
Cancel Button Cancel
Input Actions Per Second 10
Repeat Delay 0.5
Force Module Active ☐

Figure 34. EventSystem



☑ QuizManager
Tag Untagged
Layer Default

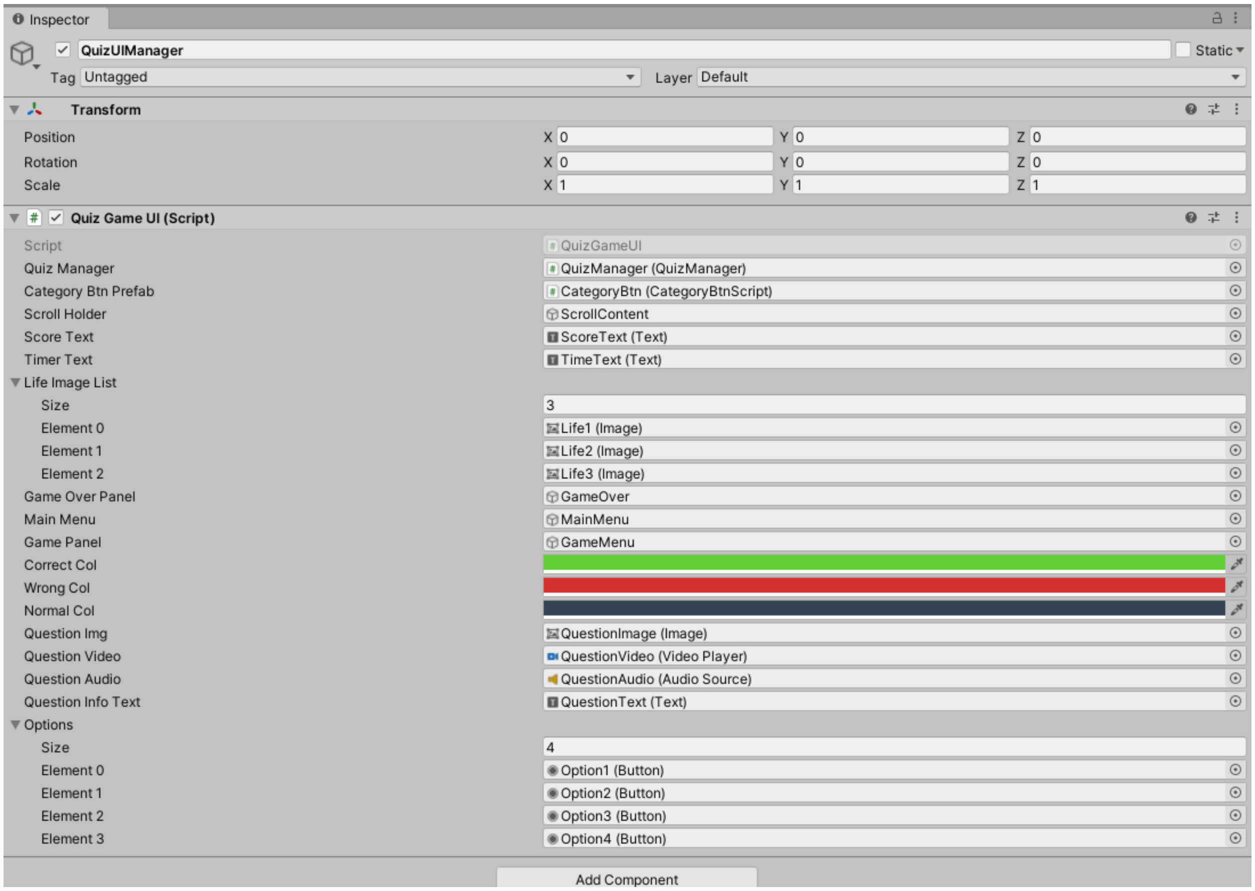
Transform

Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

☑ **Quiz Manager (Script)**

Script QuizManager
Quiz Game UI QuizUIManager (QuizGameUI)
Quiz Data List
Size 1
Element 0 Questions (QuizDataScriptable)
Time In Seconds 420

Figure 35. QuizManager



☑ QuizUIManager

Tag Untagged Layer Default

Transform

Position	X 0	Y 0	Z 0
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

☑ **Quiz Game UI (Script)**

Script	QuizGameUI
Quiz Manager	QuizManager (QuizManager)
Category Btn Prefab	CategoryBtn (CategoryBtnScript)
Scroll Holder	ScrollContent
Score Text	ScoreText (Text)
Timer Text	TimeText (Text)
Life Image List	
Size	3
Element 0	Life1 (Image)
Element 1	Life2 (Image)
Element 2	Life3 (Image)
Game Over Panel	GameOver
Main Menu	MainMenu
Game Panel	GameMenu
Correct Col	🟩
Wrong Col	🟥
Normal Col	■
Question Img	QuestionImage (Image)
Question Video	QuestionVideo (Video Player)
Question Audio	QuestionAudio (Audio Source)
Question Info Text	QuestionText (Text)
Options	
Size	4
Element 0	Option1 (Button)
Element 1	Option2 (Button)
Element 2	Option3 (Button)
Element 3	Option4 (Button)

Figure 36. QuizUIManager

6. Testing & Validation

Testing was done by “playing” the app in Unity. We used the Unity option to run the application from our computers to test the application. We tested all pages, buttons, items in the database, and any other subfunctions. Unfortunately, we were not able to test the application on a phone as there was incompatibility between our phones and our computers, the tester all had IOS running devices and windows running computers. This restricted some of our testing.

Testing the home screen was pretty straightforward. It was tested on layout/design, colour scheme, and functionality of the three buttons at the bottom. This was done as a pass or fail test which each button was pressed and if it advanced to the next scene then it passed, and if it did not then it failed. When testing we found that the majority of the testers liked the layout and colours of the final prototype, which took the team till the last prototype to get. As well all three buttons passed as they were able to switch to their designated scenes.

Testing the search scene was also straightforward. When testing it was also based on the pass/fail method. If the ‘To Website’ button opened the Ottawa Waste Explorer the test passed, which it did. Next, was the second button in that scene which is the return to home. This button passed during the second prototype as it was able to return the user to the home screen when pressed.

The scan scene was not only tested on the pass/fail scale for buttons, but also the correctness and accuracy of its recognition. The return home buttons on every canvas for each item when it was recognized had to pass, which they did in the final prototype when they were all capable of returning the user to the main screen. As well each item was scanned a number of times in different lighting and from different angles to ensure that the AR camera was able to recognize the item. After the item passed these tests by the panel appearing the item passed. Lastly, the home button was passed in the second prototype as it was able to return the user to the main screen.

Lastly, the game was all based on the pass/fail scale. All buttons were tested to ensure that the application knew which answers were correct and which were wrong. After this a few run throughs were done to ensure that when the time ran out that the application showed the game complete pop-up, which passed its test in the second prototype. The home button was passed during the first prototype as it was able to redirect the user to the home screen.

The results from all the tests done during the testing phases were passed before the final prototype was completed.

7. Conclusions & Recommendations for Future Work

Overall, due to unforeseen circumstances and the timeline to complete the application that the EcoTech team faced, the final prototype was a little different from what we envisioned to make for our client, but overall we are happy with the results.

We've learned there is a lot of hard work involved in putting an application together and the testing process is not always easy. The scanning features database is not as large as we would have hoped for the final design, adding more items and improving the app's scanning capabilities would be a main focus in continuing the application in further developments. As well, partnering with the City of Ottawa to gain access to its Waste Explorer database, so the search subfunction of the application could be within the app and not an external website would be another big feature that we would have liked to add if we had more time.

Some other things to think about incorporating would be multiple languages, as not all Ottawa residences main language is English. Possibly adding a social portion of the app and allowing the users to go into head to head competitions when playing the game may be another thing to think about adding in the future.

8. Bibliography

Public Works and Environmental Services Dept. (2019, September 09). Solid waste – data and reports. Retrieved April 19, 2021, from https://ottawa.ca/en/garbage-and-recycling/solid-waste-data-and-reports

Appendices

9. Appendix I: Design Files

This document is the final deliverable of the project for the EcoTech application. None of the deliverables leading up to this deliverable is referenced; they are still relevant to the creation and completion of this application.

All final deliverables and files mentioned above can be found on theses two websites:

- MakerRepo: https://makerepo.com/claire66/852.gng1103c8ecotech
- Github: https://github.com/claire588/waste-management/tree/Completed-App

Document Name	Location	Issuance Date
Deliverable B	https://makerepo.com/claire66/852.gng1103c8ecotech	January 31, 2021
Deliverable C	https://makerepo.com/claire66/852.gng1103c8ecotech	February 7, 2021
Deliverable D	https://makerepo.com/claire66/852.gng1103c8ecotech	February 21, 2021
Deliverable E	https://makerepo.com/claire66/852.gng1103c8ecotech	February 28, 2021
Deliverable F	https://makerepo.com/claire66/852.gng1103c8ecotech	March 7, 2021
Deliverable G	https://makerepo.com/claire66/852.gng1103c8ecotech	March 21, 2021
Deliverable H	https://makerepo.com/claire66/852.gng1103c8ecotech	March 28, 2021

EcoTech App files	https://github.com/claire588/waste-management/tree/Completed-App	April 4, 2021