

# Asthma App

Manar Ajhar  
manarajhar@hotmail.com

Dr. Suha Al-Naimi  
sal-naimi@sharjah.ac.ae

October 7, 2024

## **Abstract**

[Text Here]

## **1 Introduction**

[Text Here]

## 2 Methodology and Procedures

[Text Here]

### 2.1 Programming and Mathematical Terms

There are several components and aspects in the field of programming that we need to discuss before heading to the methodology:

- **Class:** An asset that contains special properties and represents a specific object in a system or program.
- **Game Engine:** A computer program that is used to develop video games.
- **Inheritance:** The process of a class to derive the properties from another class to have all its properties and other more. For example, if “Class B” inherits from “Class A”, then “Class B” is a sub-class of “Class A”, and “Class B” gets access to all (or most) of the properties of “Class A”, while also has its own. Inheritance represents an “is a” relationship.
- **Singleton:** A class that can only have one instance of it.
- **Library:** A collection of prewritten code that serves a unique purpose to help programmers enhance their work tasks and projects [1].
- **Script:** A computer file that contains code written in a specific programming language. Some scripts contain only one class, while others can have as many classes as possible.
- **Encryption:** The process of converting a file in a readable format to an unreadable format to prevent users from reading and tampering with the information.
- **Decryption:** The process of converting a file in an unreadable format back to its original (readable) format.
- **Axis:** A long, straight, and infinite line that represents a specific direction. In 3D graphics, there are three axes (each in their negative and positive directions, respectively):
  1. X-Axis: Left and Right
  2. Y-Axis: Down and Up
  3. Z-Axis: Backwards and Forward
- **Transform:** The geometrical measurement of an object in a 3D environment. There are three components of a transform:
  - Position
  - Rotation
  - Size

## 2.2 Unity

The program that the team agreed to build this endeavor with is Unity [2]. It is the main program used to develop the application. It is a game engine that has a friendly user interface and allows developers to flexibly program their games with as many utilizations as possible. The version that was used for this project was “2021.3.5f1”. In Unity, the programming language that was used was C#. C# is an object-oriented programming language, which means we can create classes and assign them to certain objects.

The target platforms that were desired for this game are smartphones and tablets. They allow Augmented Reality (AR) technologies, having virtual scenes and objects “blend in” with the physical world. Smartphones also have touch as their input method, which can simulate actions that happen every day, such as picking-up objects, pressing buttons, or shaking objects.

However, other tools were used to complete the application. For example, Photoshop was used in this project to design badges, buttons, icons, and other UI elements. It also helped change the color/hue of UI elements. The version of Photoshop that was used is “20.0.4”. Another software that was used is Blender. It was used to make simple modifications on 3D models and assets and make some 3D animations.

## 2.3 AR Package

The AR package that is implemented for the project is Google’s ARCore package. The reason that this package is chosen for this project include:

1. It is compatible with iPhones and Android devices, which are the most dominant devices in their market<sup>1</sup>
2. Unity can let the user download the package natively from it
3. ARCore has the ability to let detect physical surfaces, such as floors, walls, ceilings, and tables. Therefore, no specific pattern is required, which are similar to traditional AR methods

Because we are using an AR environment, there needs to be special components to run the application in AR, such as AR cameras and sessions. These are assets that are within Unity. In the phone, when the camera detects a physical surface, an on-screen button will appear to tell the player to start the game. When the button is pressed the virtual scene will be placed on the very spot that the camera picked.

## 2.4 Loading and Saving

Loading and saving are very important when opening an application to avoid losing progress from any session. It is done using a “. json” file. It stores information in

---

<sup>1</sup>As of this document, according to Google’s website, in terms of iPhones, ARCore supports up to iPhone 13 [3].

text-form. There is a C# script that read from the file and writes to it for loading and saving, respectively. It also encrypts the file to avoid users from easily tempering with the data. Each savable data has at least two variables: 1. Only one key variable and 2. at least one value variable. The key is basically an ID, which is a unique string that contains 32 characters. It can be generated via C#. The value can be of any type. Therefore, we needed to program what value do we want to save from what asset, and the asset needs a key as a reference to know where the value should be stored. The process of loading is:

1. Decrypt the file
2. Read the keys and values from it;
3. Convert the values from text-form to the correct type
4. Find the variable that has its ID match the current one used in the search and assign its value to the variable
5. Keep doing this for all the data
6. Encrypt the file again

The process of saving is:

1. Decrypt the file
2. Read the keys and values from the classes
3. Convert the values from their current form to text
4. Write the information into the “.json” file
5. Keep doing this for all the data
6. Encrypt the “.json” file again

The information that we are saving are:

- The badges earned
- The answers on the action plan
- The settings
- The type of user

## 2.5 Programming and Designing the Action Plan

The Action Plan is a survey that prompts the user to ask questions about their status with asthma. It also has some questions about their personal status such as birthday, name, and gender. The overall number of questions are twenty. However, each question has a different type of answers:

1. Integer Answers

2. Float Answers
3. String Answers
4. Enum Answers
5. Answer with Two Strings
6. Date (DD/MM/YYYY Format, all integers)

Eventually, only if all questions are answered, the user will be able to make a PDF form of the plan. Also, they are saved, and if the user accesses the action plan again, the input fields will be filled with the previous answers.

### **2.5.1 Enums**

There were two types enum questions: 1. The color of the inhaler; and 2. The user's gender. These are to ensure that the user do not provide various responses to questions that only need a specific answer. Visually, the answer of enum questions is presented in a dropdown menu. When the user taps on the menu, a dropdown box will appear below it, listing the possible choices based on the enum. The user can tap on a specific answer or accept the already selected answer.

### **2.5.2 Two Strings**

There was only one question that needed two strings, which is the doctor's contact methods: phone number and e-mail. The phone number must be a string because if it is an integer (or any numerical variable type), the leading zeroes in the sequence will be discarded since they would be treated as numbers. Besides, phone numbers do not provide any mathematical purpose. Visually, there are two input field, each prompting for a contact method. For the phone number, only numerical values can be added. So, in a smartphone, when the input field for the phone number is selected, the on-screen number-pad will appear, rather than the whole on-screen keyboard.

### **2.5.3 Date**

There are three questions that ask for the date: 1. The birthdate of the user; 2. The expiry date of the medication; and 3. The next appointment with the user's doctor. There are three input fields: day, month, and year, and each one is an integer. The former two only accepts two digits, while the latter accepts four.

### **2.5.4 The Action Plan's PDF Form**

All the answers of the Action Plan can be presented in a PDF. This was done using the iTextSharp library [4], which is compatible with C# and Unity. There is a button called teal button called "Print PDF" in the action plan's UI, which is responsible for creating the PDF file. When the file is created, the very first thing the is printed is the date and time of printing and the time zone of the device's current location.

Below it, a table is printed showing the personal information of the user, which are 1. Name, 2. Date of birth, 3. Age, and 4. Gender. The age is automatically calculated by the difference between the date of printing and the date of birth. The others are taken from the user's input, from the app. After this, a list of all the questions and their answers (Other than the personal information) are printed across the document. As for the date and time formats:

- All the dates in the PDF are in the following format: “MMMM dd, yyyy (dddd)”, for example “September 02, 2024 (Monday)”
- The time format used is in 12-hour format, for example “7:05 pm”. Also, it measured the time zone by using “UTC” time zone as a base. For example, “UTC+4:00” in the time zone of the United Arab Emirates, or “UTC+1:59” in Croatia.<sup>2</sup>

## 2.6 The Exhibition

The exhibition is a unique experience that users can try. It is similar to seeing exhibits in a museum, and if there is a specific object or display that they want to know more about, they press the corresponding number that is displayed next to the object on their small digital companion. The exhibits that we are showing in the exhibition are the types and components of inhalers, as well as a whole assembled inhaler.

### 2.6.1 Preparing the Exhibition

In order to make the exhibition visually compelling in a 3D space (Even in AR), some math skills are required. The exhibits are displayed by being spread across in a perfect circle. To make the circle, we need to get the direction of each exhibit and choose a magnitude.

**Getting the Direction** The first variable that we will need is the angle of the current exhibit, which is done by the following equation:  $\theta_i = (\frac{i * 360}{n})$ .<sup>3</sup> In the equation:

- “ $i$ ” is the current index (Must be a whole number and greater than or equal to 0)
- “ $\theta_i$ ” is the angle of the current index (The angle is in degrees). A full circle is 360 degrees
- “ $n$ ” is the total number of exhibits, so we can divide the circle into “ $n$ ” equal parts (Must be a whole number and greater than 0)

---

<sup>2</sup>Manar has tested printing the action plan in Croatia in late July of 2024. The time zone was accurate.

<sup>3</sup>“ $\theta$ ” is a Greek symbol, pronounced “Theta”. It is a variable used in geometry to represent angles.

Once we get the angle, we need to get its sine and cosine values to establish the direction in the z-axis and the x-axis, respectively:

- $dir_{i,x} = \cos(\theta_i * \frac{\pi}{180})$
- $dir_{i,z} = \sin(\theta_i * \frac{\pi}{180})$

In Unity, the standard unit for angles are radians. Thus, we need to convert the angle from degrees to radians before getting the sine and cosine values, by multiplying it with the fraction ( $\frac{\pi}{180}$ ).<sup>4</sup> “ $dir_{i,y}$ ” is equal to zero, since we do not want to spread the exhibits across the y-axis, regardless of the value of “ $i$ ”. “ $dir_i$ ” is a Vector3 variable.

**Completing the Display** Lasty, we multiply the values of “ $dir_{i,x}$ ” and “ $dir_{i,z}$ ” with the radius to determine the position of the current exhibit (in meters, which is Unity’s default unit for length/distance measurement). Therefore, the final function is: “ $pos_i = dir_i * r$ ”, where “ $r$ ” is a positive float, which represents the magnitude (or radius) of the circle. It will tell how far the exhibits should be from the center. We repeat the process for all exhibits, and with each next iteration, “ $i$ ” is increased by one until it equals the value of “ $n$ ”. We set the radius to be 500. Like “ $dir_i$ ”, “ $pos_i$ ” is a Vector3 variable.

## 2.6.2 Showcasing the Exhibition

In Unity, there is a technique called “Raycasting”, that is used to make the exhibition work. A raycast is a straight line that starts from one point and continues infinitely (by default) until it detects an object in the way. It is, conceptually, similar to a laser beam. At the exhibition, the raycasting line/beam starts from the center of the user’s camera. There is a special UI for the exhibition, which has the following components:

- A red circle, which is in the center of the screen, which is the same position as the starting point of the raycasting line
- A back button
- A microphone button which appears only when an exhibit is hit by the line and disappears otherwise

All exhibits have one script in common, called “ExhibitionObjectScript”. When the user places the center of his/her camera at an object, the raycasting line will detect if it contains the mentioned script. If so, the object will be highlighted in a specific color, the red circle will turn green, the name of the object will be displayed, and the microphone button will appear below. When the microphone button is pressed, Dr. Salem will discuss the highlighted object.

---

<sup>4</sup>“ $\pi$ ” is a Greek symbol, pronounced “Pi”. It is a constant value, which equals to 3.141592

## 3 Evaluation

[Text Here]

## 4 Conclusion and Future Work

[Text Here]

## 5 Definitions

- Cosine: The ratio of the adjacent side of a corner in a right triangle to the hypotenuse of the triangle:  $\cos(\theta) = \frac{\text{adjacent}}{\text{hypotenuse}}$ .
- Sine: The ratio of the opposite side of a corner in a right triangle to the hypotenuse of the triangle:  $\sin(\theta) = \frac{\text{opposite}}{\text{hypotenuse}}$ .
- Raycasting: using a straight line that starts from one point and continues infinitely (by default) until it detects an object in the way.
- Canvas: A UI element that contains other elements, including text, images, buttons, or other canvases.
  - On-Screen Canvas: A canvas that is always fixed in the display's screen.
  - World Space Canvas: A canvas that exist in a world with other objects.

### 5.1 Types of Programming Variables

- Enumerator: A variable type in programming that has a certain number of values and can only be assigned to one value.
- Integer: A whole number, such as: 1, 3, -8, 100, 2006.
- Float: A decimal number, such as: 1.0, 5.2, -10.33, 7604.099.
- String: A text or sequence of characters, such as: "Hello World, 12381 42!".
- Vector2: A collection/array of two float variables. The first one is named 'x', and the second one is named 'y'.
- Vector3: A collection/array of three float variables. In addition to the two variables in Vector2, the third one is named 'z'.

## 6 Acronyms

- Enum: Enumerator
- AR: Augmented Reality



- UI: User Interface

## References

- [1] *What is a Programming Library? A Beginner's Guide*. Accessed on: October 4, 2024. Jan. 2023. URL: <https://careerfoundry.com/en/blog/web-development/programming-library-guide/#what-is-a-programming-library>.
- [2] *Unity*. URL: <https://unity.com/>.
- [3] *ARCore supported devices*. Accessed on: October 4, 2024. Jan. 2023. URL: <https://developers.google.com/ar/devices>.
- [4] *iTextSharp*. URL: <https://itextpdf.com/products/itextsharp>.