

**CZ4001: Virtual and Augmented Reality  
AY 18/19 Semester 2**

**MemoryAR – A Memory Training AR Game**

**Project Report**

**Student Name: Sim Long Siang**

**Matriculation Number: U1522053H**

**18 April 2019**

# Report Objective

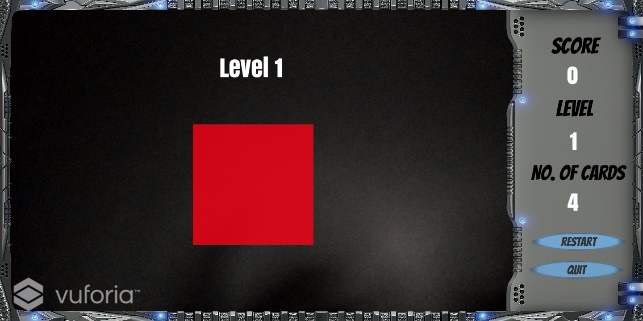
The objective of this report is to document the game construct and features of MemoryAR : A Memory Training Game, which is part of the assignment requirement of CZ4001 Virtual and Augmented Reality. In this report, the author will explain the thought process behind the creation of MemoryAR, and some AR features that is critical to the game application. The report will also discuss possible future development of the application.

With Augmented Reality (AR) technology getting more widespread and prevalent, developers are creating it for different sectors, like entertainment, education, game and many more. In addition, the exposure of newer technologies to younger generation increases the demand for AR-related application. Therefore, MemoryAR is designed with the target audience towards the young.

**MemoryAR – A Memory Training Game**



MemoryAR is a single player marker-based memory training game for the young. Created using Unity and Vuforia as the ARtoolkit, the application aims to train and improve a user’s brain memory through a series of flashing images, and user is required to remember the order of those images. In this first version, poker cards are used as the marker for the game.

Upon entering the game, instruction of the game are flashed to the user. Next, red or black boxes will be flashed in random for an interval of 1 second. After the flashing is completed, instruction to scan the red or black card will be shown. The red card is represented using the 8 of Hearts while the black card is represented using the 8 of Spades.

As the cards are scanned, the respective 3D graphics will be rendered and appeared on the screen. At the same time, the game will be able to detect if the card scanned is in correct order. If the card scanned is in wrong order, the game stops and prompts the user to restart. If the card scanned is in correct order, the score will be added (1 point for each correct card). If the whole sequence is scanned correctly, the game will move on to the next level, increasing the difficulty by adding 1 more count to the total flashed image.

**Game Construct of MemoryAR**

The game mechanics of MemoryAR consists of several functions as shown below. Each function calls another function upon the completion of their individual task. This creates a procedural loop as the game progresses.

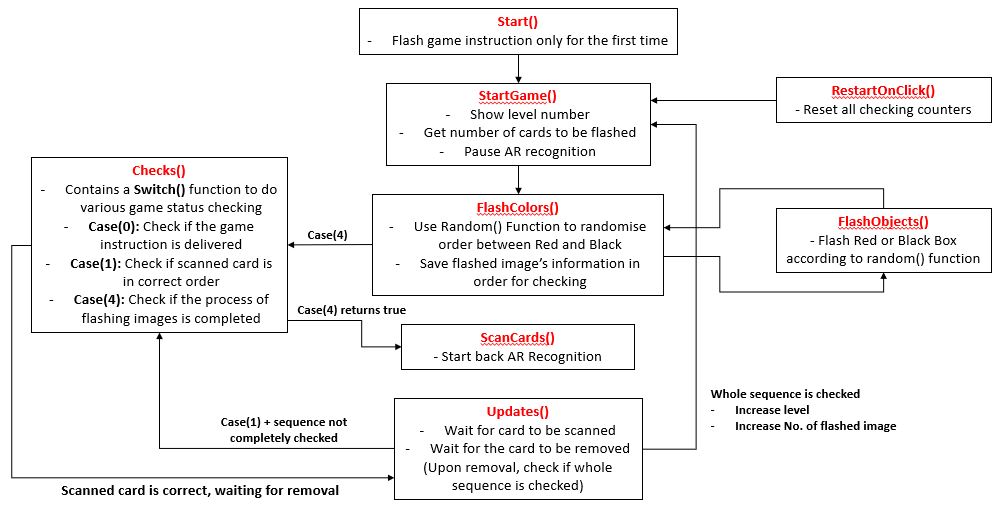


Figure . Function flow of MemoryAR

There are 3 main code features that make this application possible:

1. **Invoke() and Invokerepeating() functions**

Invoke() and Invokerepeating() functions allow the calling of other functions after a set period of seconds. In addition, Invokerepeating() acts as a replacement for the while() function, with a terminating function, CancelInvoke(). With that in mind, MemoryAR uses Invokerepeating() to implement the flashing of the images, controlling the speed that it flashes. Also, it uses Invoke() to call other functions, making the procedural aspect of the code happen through coordination of timing each function runs between each other.

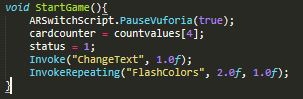


Figure . Using InvokeRepeating() function to flash colored box after 2 seconds, in 1 second interval.

1. **Manipulating AR detection**

During the process of flashing the sequence of images, AR recognition and detection needs to be paused to avoid user accidentally scanned the image target, causing runtime error as the checking of cards happens after the flashing image process. Thus, a static method **PauseVuforia(bool what)** house the stopping and starting function of AR recognition, namely **imageTracker.Stop()** and **imageTracker.Start().** This implementation allows the stopping and starting functions to be called by passing in a Boolean value to indicate the status of AR detection.

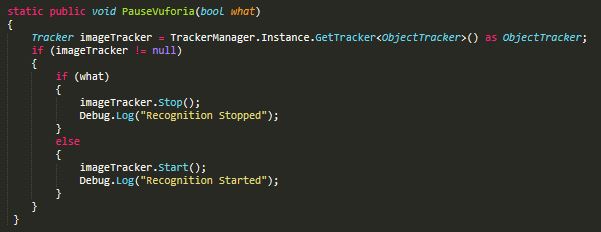
****

Figure . AR Script that starts or stops the detection/recognition.

1. **Getting image target’s name**

In order to check if the scanned card is correct, the application has to be able to determine what the scanned card is. Vuforia has a DefaultTrackableEventHandler script that generates the image target name. Thus, extracting this variable will allow the main game script to perform correctness check using If-Else() function.

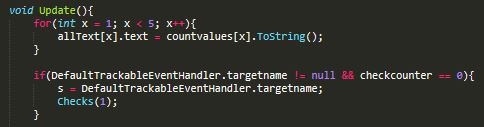
****

Figure . Using DefaultTrackableEventHandler.targetname to get the image target name

**Future Development**

The creation of MemoryAR opens a wide range of function for the masses. In the education sector, MemoryAR can be a visual teaching tool, such as introducing new words and getting them exposed to those word through memory training. This can be achieved by replacing the flashing images with words, and assigning each card used to a word for the game to work. Also, each card can render a 3D model that represents the word assigned. In short, the idea of a dynamic marker (markers that renders different object on different setting) unlocks the possibility of learning words through visual.

In other sectors, MemoryAR can be used as:

* an aid to help elderly with failing memory,
* Color-blind testing,
* Learning assessment tool (flashing questions and asking user to scan correct answers)
* Spelling Bee competition (flashing letters in a word in random order, and scanning card to determine correct letter order)

In conclusion, the creation of MemoryAR is to help create or improve solution for existing problems. The capability of AR technology is one that should be heavily explored and implemented to make the world better.