

Vocabulary Game Using Augmented Reality—Expressing Elements in Virtual World with Objects in Real World

Kong Yang, Haga Hirohide

Doshisha University, Kyoto, Japan

Email: ykong@ishss10.doshisha.ac.jp, hhaga@mail.doshisha.ac.jp

Received December 2014

Abstract

The purpose of this study is to express elements in virtual world with logical constraints using objects in real world with physical constraints. As an example to achieve this purpose, this paper presents an English vocabulary game utilizing game engine Unity and programming language C# along with ARTool Kit, a software library for building Augmented Reality (AR) applications. This game, designed for the purpose of education by aiming at helping students memorize English vocabulary, can be classified as a serious game. Watching the question provided by the game program on the screen, which designates alphabets player can use to assemble into a word, the player arranges the cardboard boxes with alphabets on it in the real world. Through the camera, the AR program detects the arrangement and position of the markers which are the alphabets on the boxes. Then, the game program judges whether the arrangement is the correct English word. This game expresses the game objects in the game program using cardboard boxes in the real world.

Keywords

Serious Game, Augmented Reality, Vocabulary Game, AR

1. Introduction

Until the digital games became the mainstream of gaming, games were designed and played in the physical world using real-world objects. Recently, the mainstream digital games focus the player's attention on the screen of computers or smart devices. However, there seems to be an increasing interest in games with a focus outside the screen. In recent years, with the growing globalization, Japanese government proposed enhancing English-language education in elementary schools by making it an official subject for fifth and sixth graders. Younger and younger students will need to study English. As the power of vocabularies is one of the foundations in the process of studying English or any foreign languages, we devised a way that students can memorize vocabularies by playing games. There are abundant of vocabulary games, but most of them are limited on screen, that is, in the virtual world. This paper proposes a vocabulary game using Augmented Reality which allows players play with the objects in the real world. This novel game systems bring both features of physical gaming and computer gaming. In other words, this game system connects the real world with physical interaction to the virtual world through the game space developed with game engine Unity.

2. Game Design

This game is designed as a gaming style that player plays with the real world objects in front of the computer or smart devices with a built-in camera. On the devices' screen, the game program presents the player the question, which indicates the letters players can use to assemble into a word, as shown in **Figure 1**.

The letters are presented in alphabetical order by sorting letters in an existing English word. If the letters are presented randomly, there is a chance that no word can be assembled using the listed letters. With the method we presented, all of the questions have a correct answer. We prepared 3 levels in this game: 3 to 4 alphabets are presented in the "easy" mode; 5 to 6 alphabets in the "normal" mode; 7 to 8 alphabets in the "hard" mode. In the first version of this game system, we prepared 10 words for each level. The words are listed in **Table 1**.

The game begins in the normal mode. If the player keeps answering the questions correctly, the game moves to the hard mode. Otherwise, the game moves to the easy mode. In the real world, we prepared 26 pieces of paper with printed alphabets. All the paper, which act as marker, are attached on the cardboard boxes. **Figure 2** shows the example of markers and **Figure 3** shows the markers on the cardboard boxes.



Figure 1. Question



Figure 2. MarkerA (left) and Marker B (right).

Table 1. Word list.

Easy	Normal	Hard
BAG	CHAIR	PICTURE
ROSE	PENCIL	HOSPITAL
BAT	WATCH	KITCHEN
FISH	PIANO	COUNTRY
DOG	ORANGE	DAUGHTER
DESK	FRUIT	TUESDAY
MILK	PLANE	THURSDAY
BOAT	RACKET	CAREFUL
BIRD	BREAD	SUBJECT
CUP	TABLE	YOURSELF

Player arranges the cardboard boxes next to one another in front of the camera. The boxes, along with their positions, are recaptured seamlessly in the game space. In the game space, it is possible to judge if the boxes are putting together in the real world by simply checking their counterparts in the game space. Setting a fictional game object on each left and right edge of the marker as shown in **Figure 4**, the game program checks if a game object on the right edge of a marker contacts with the other game object on the left edge of the other marker.

We placed 26 pairs of fictional game objects (blue and red cubes) to overlay left and right virtual boundaries on the cardboard boxes in the real world. The initial state of the game space are shown in **Figure 5**.

Red and blue cubes are left and right virtual boundaries of the boxes of 26 letters. Collision judgment program are embedded in the game object on the right edge of each marker using On Trigger Stay and On Trigger Exit function supported in Unity. Names of all the game objects are assigned as the corresponding alphabets. If a game object contacts with the other, that is, the next letter player arranged, the name of both game objects are stored. Word composition program then sort the stored letters according to their position in the game space. Taking cognizance of the fact that camera reverses left and right on the screen, we used x coordinate of the virtual objects in the game space to recognize the position of the cardboard boxes in the real word. With these data of coordinates, we were able to compose the single letters into the word in the order that the player arranged. The examples of the results of the composed word (Original Composed Word) with this algorithm are shown in **Table 2**.



Figure 3. Cardboard boxes.

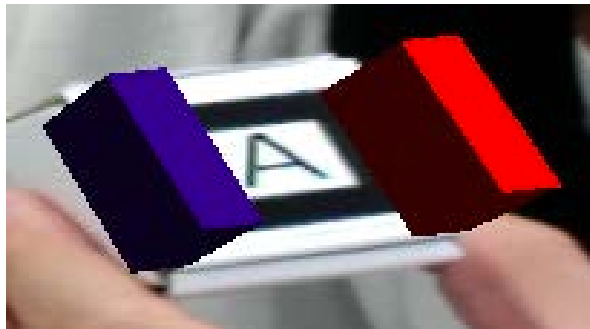


Figure 4 Game objects on the marker.

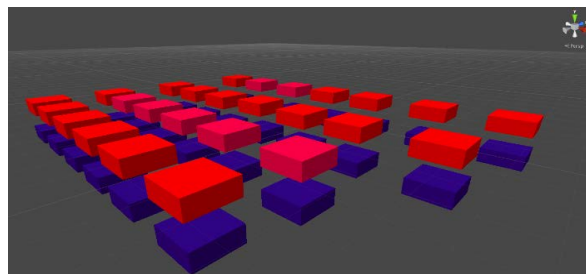


Figure 5. The game space.

Then, we define a new String variable that store the final result of the composed word (Final Composed Word) based on the original one with repeated letters. We extract the first letter, the even-numbered letter except the first and the final one, and the final letter from the Original Composed Word. Then store it to the Final Composed Word. The example of the Final Composed Word are listed in **Table 3**.

This program outputs the current partial word based on the detected row of boxes in real time. Finally, when the length of the Final Composed Word equals the length of the word presented in the question, the word judgment program checks its correctness. **Figure 6** shows the screen shot of an example of a correct word. **Figure 7** shows the screen shot of an example of wrong word. If the answer is wrong, the correct answer is shown. In either case, the meaning of the word is presented. And the player can tap the screen or click the mouse to move to the next question.

3. Augmented Reality

Augmented reality is a live, direct or indirect, view of a physical, real world environment whose elements are augmented by computer-generated sensory input such as sound, video, graphics or GPS data. It is related to a more general concept called mediated reality, in which a view of reality is modified (possibly even diminished rather than augmented) by a computer. As a result, the technology functions by enhancing one's current perception of reality [1]. In this game system, we choose ARToolKit, a computer tracking library for creation of strong augmented reality applications that overlay virtual imagery on the real world. To do this, it uses video tracking capabilities that calculate the real camera position and orientation relative to square physical markers in real time. Once the real camera position is known a virtual camera can be positioned at the same point and 3D computer graphics models drawn exactly overlaid on the real marker. So ARToolKit solves two of the key problems in Augmented Reality: viewpoint tracking and virtual object interaction. ARToolKit was originally developed by Hirokazu Kato of Nara Institute of science and Technology in 1999, and was released by the University of Washington HIT lab. [1]. The ARToolKit tracking works as follows:

1. The camera captures video of the real world and sends it to the computer.
2. Software on the computer searches through each video frame for any square shapes.
3. If a square is found, the software uses some mathematics to calculate the position of the camera relative to the black square.
4. Once the position of the camera is known a computer graphics model is drawn from that same position.
5. This model is drawn on top of the video of the real world and so appears stuck on the square marker.
6. The final output is shown back in the handheld display, so when the user looks through the display they see graphics overlaid on the real world.

ARToolKit is able to perform this camera tracking in real time, ensuring that the virtual objects always appear overlaid on the tracking markers [2]. **Figure 8** summarizes how ARToolKit tracking works.

Table 2. Original composed word.

1 letter	NONE
2 letters	AB
3 letters	ABBC
4 letters	ABBCCD
5 letters	ABBCCDDE

Table 3. Final composed word.

1 letter	NONE
2 letters	AB
3 letters	ABC
4 letters	ABCD
5 letters	ABCDE



Figure 6. Correct word.

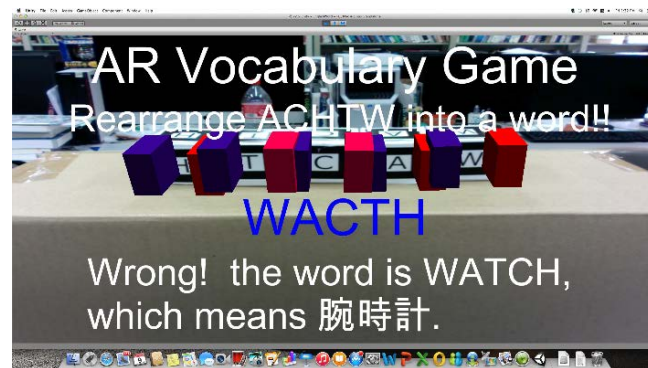


Figure 7. Wrong word.

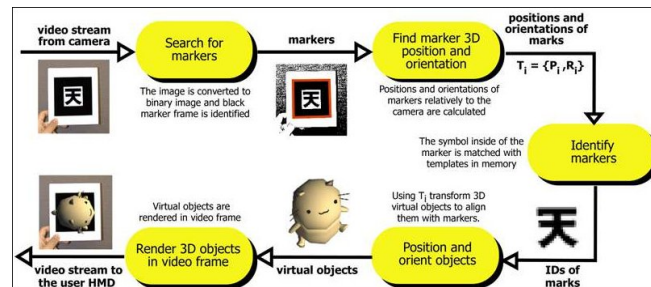


Figure 8. Basic principles of ARToolKit [2].

We choose ARToolKit because it supports multi-markers, because we can design the markers for ourselves, and because the virtual imagery in the game space changes its position when the marker moves in the real world. Other software library for building AR applications, such as Vuforia, provided by Qualcomm Connected Experiences, run in the different way which instead of the virtual imagery, the main camera moves when the marker moves in the real world.

4. Prototype Development

Figure 9 shows the development environment, on which this game system was implemented and examined.

Figure 10 shows the system configuration of this game system.

For the purpose of reducing the misrecognition of letters, only the letters which are designated in the question are on standby as markers. Compared with the algorithm which all of the 26 letters are active for recognition, the misrecognition rate decreased considerably. Also, we had a problem that symmetrical markers such as letter I, X and O, tend to be tracked upside-down, which causes the malfunction during the collision judgment. We solved this problem by adding two little square in the bottom of such markers, as shown in Figure 11.

Operation System	Mac OS version 10.9.5
Development platform	Unity 4.3.4 [3]
Camera	iMac built-in iSight
Programming language	C#

Figure 9. Development environment.

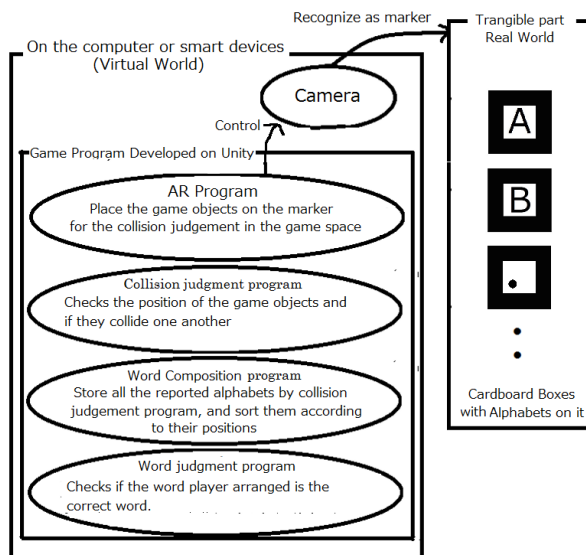


Figure 10. system configuration.

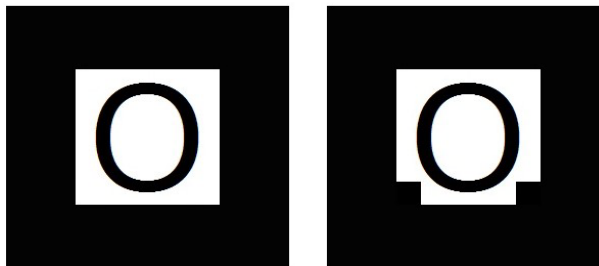


Figure 11. Marker O (Left: original, Right: improved).

5. Conclusion and Discussion

This game provides seamless connection between physical world and virtual world. In most AR games, the player controls the virtual game objects on the screen, while the markers remain stationary. With the idea of moving the markers, players are able to have a novel game experience as this paper presented.

However, there are some limitations in this game system. First, all alphabets can be used only once. Words are not supported in this game if the same alphabets appear more than once in the word. Second, the number of questions can be raised enormously if the program loads English words from online dictionary APIs.

This game system can be easily applied to vocabulary games in other languages by simply changing the markers and the word list.

References

- [1] Carleton Olegario M. Ximo, Ed. (2012) ARToolKit. Ject Press, USA, UK, Germany, 1-4.
- [2] <http://www.hitl.washington.edu/artoolkit/documentation/index.html>
- [3] <http://unity3d.com>