AR Cooking - Gyozas

Joelle Helgert*
FH OOE, Campus Hagenberg

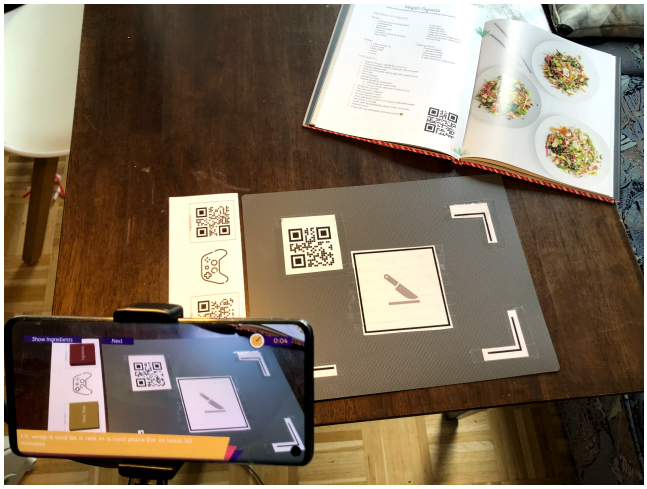

Figure 1: Picture of the cooking set-up

ABSTRACT

The application AR Cooking shall help inexperienced chefs to create tasty dishes. For this purpose, it uses Augmented Reality to provide additional hints, like how to chop an onion finely. A special cutting board which is enhanced with markers is needed, to use this application. Also, a set of buttons can be printed to enhance the experience. The currently provided prototype only includes the recipe for Gyozas.

Index Terms: Human-centered computing—Interaction Paradigms—Mixed / augmented reality; Human-centered computing—Interaction Paradigms—Graphical user interface;

1 IDEA

Cooking is an essential skill, which is needed on a daily basis. Unfortunately, many dishes are complicated and, therefore, many people nourish themselves with fast food or other simple meals. This application will use modern technologies to simplify cooking and help those people. A smartphone, attached to the table as visible in figure 1, is currently used, but later a Head-Mounted Display (HMD) will be used. Here it is vital to use an optical see-through HMD, such as the hololens. Otherwise cutting might be dangerous due to the nature of non-see-through HMDs where there is a short delay in the video feed and a disconnect from the three dimensional space. The application provides the user with information on what ingredients are needed and what steps need to be done. Additionally, it augments elements on the ingredients to show how they are prepared best.

*e-mail: joellehelgert@gmail.com

2 APPLICATION

After starting the app in landscape mode, the user enters a screen to scan the QR code of the recipe. When the QR code is detected, the user can start the cooking process. The current AR Cooking version only supports one recipe; the gyoza recipe by Bianca Zapatka [7]. This recipe was chosen because gyozas feature various complicated steps which could benefit from more detailed explanations. The augmented features are cutting an onion and shaping the dough. Here it would have also been helpful to augment the folding of the gyoza. This is not implemented due to the complexity of the augmentation and animation task. As general features, the user can always open the ingredients list during the cooking and the step by step instructions for the recipe. Since some of the steps are time relevant, a timer is also implemented. The timer visualises the time left and rings after the time is over. The application is developed with the game engine Unity [5] and the AR engine Vuforia [6]. Some of the graphics are from the website flaticon [2–4], and the sound is from freesoundslibrary [1]. In the following the implemented parts are explained in detail.

2.1 Data Reading

Character-Separated values(CSV)-files are used to store the recipe data. For one recipe, two kinds of files are needed. One file keeps the ingredients and its amounts, whereas the other file describes the different recipe steps. Next to the step description and its number, its type is stored. Currently, the following step types are available.

- Headings: These are used for the different parts of the recipe. For example, preparing the dough or filling and pan-frying the gyozas.
- Basic Step: A standard step with no augmentation.
- Timer: A time-sensitive step which starts a timer.
- Onion: Shows additional instructions for cutting an onion
- Circles: Augmentation to visualise how the dough needs to be cut out.

After scanning the recipe code, the path to the CSV-files is set. When the cooking is started, the recipe ingredients and instructions are loaded.

2.2 Start screen

The start screen is just a prototype with basic features. It allows the user to scan recipe codes and shows the belonging recipe title. Later this screen shall also provide a settings section where the user can change the sound settings and define their cooking skill level.

2.3 Visualisation of recipe

The central part of the application is the visualisation of the recipe, consisting of one Unity scene and several C# scripts. It provides a primary user interface (UI) for the ingredients, instructions and the timer. Additionally, it has a tiny next and an ingredients button as fallback buttons if the augmented controls do not work or are not in reach. The different UI-Elements are hidden as long as they are not needed. The same applies to the special elements since they are already placed in the scene and only activated during runtime.

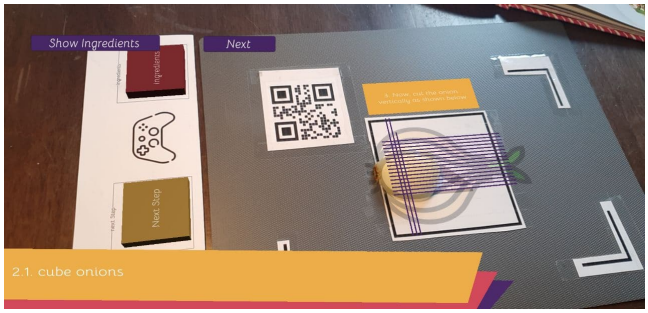

Figure 2: Visualisation for chopping an onion

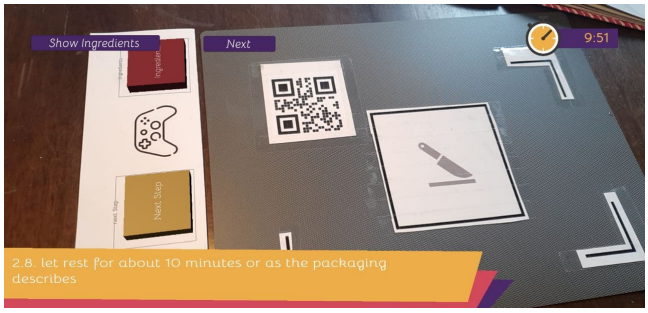

Figure 3: Visualisation of the timer

2.3.1 Cutting onions

Cutting onions needs to follow a few rules to chop it finely. First, it is cut horizontally, leaving out approximately 2mm at the onion's stem so the onion doesn't fall apart during the subsequent vertical cutting. To show this in the application, the onion's cutting lines appear line by line and do not cut to the end. Since this may also seem to be a misplacing of the lines, the instruction text also tells users to stop shortly before the onion's root.

The whole augmentation is done with simple Unity animations which are started after each other. After both sets of lines, horizontal and vertical, are drawn, the augmentation is visible for five more seconds and then disappears. The results can be seen in figure 2 below.

2.3.2 Timer

When it comes to cooking some steps are time relevant. This means that some ingredients shall either boil, rest, brown or stay in the oven for a certain amount of time. If the chef is busy, this is sometimes forgotten; hence, a timer is integrated, which rings after time is up. The timer is visible the whole time in the upper right corner, as visible in figure 3.

2.3.3 Dough shaping

For preparing gyozas, the dough needs to be cut into circles, stuffed with the filling and folded. Cooking them the first time it is challenging to know how big the circles need to be to stuff and fold them easily. For this reason, circles in the right size are augmented on the dough to provide the user with a size they can orientate on.

2.4 Problems

During the implementation, a few complications were faced. In the initial approach, the start screen should scan different known targets and then look up the correct recipe and its path in a dictionary. This approach would be easier to scale for more recipes later. Unfortunately, this turned out to be more complicated than expected hence the application checks for the gyoza recipe target directly. Another problem was the implementation of the control buttons. At first, the buttons were represented by two individual targets, but sometimes the covering of the target did not trigger the pressing of the button. On other occasions, the target is lost when fully covered, so the combined button area was created to provide additional features. Retrospectively this is the more attractive solution since carrying to single targets would be annoying - especially if later other buttons, such as a back button, were added. Regarding the augmentation of the onion cutting step, timing was the biggest struggle. The cut lines should all appear one after another and the text should change after all horizontal lines were drawn. This is implemented code-based with two for loops, that iterate over the lines and trigger the animation. The problem here is starting each animation after it's predecessor finished. When using C#'s `Sleep` function the entire application pauses which makes it seem like the app has crashed for several seconds. Therefore Unity's `WaitForSeconds` was used instead which needs to be executed in its own function to return an `IEnumerator`. The function is called several times, which is why the text sometimes was not updated correctly. The code is depicted in the listing below. The animation process could probably have also been solved with Unity itself, since Unity can create several more complex animations, but the explained approach worked better for the developer.

```
IEnumerator DrawLines(Animator[] hanimators,
    Animator[] vanimators)
{
    if(!horizontalDone)
    {
        HintText.text = "2. Cut the onion
    horizontally as shown below. Do not cut until
    the end!";
        foreach (Animator anim in hanimators)
        {
            yield return new WaitForSeconds(1);
            anim.Play("line_anim");
        }
        horizontalDone = true;
    }

    if(horizontalDone)
    {
        HintText.text = "3. Now, cut the onion
    vertically as shown below";
        foreach (Animator anim in vanimators)
        {
            yield return new WaitForSeconds(1);
            anim.Play("line_anim");
        }

        yield return new WaitForSeconds(5);

        CutOnion.SetActive(false);
        cutting = false;
    }
}
```

Listing 1: Animation of Cutting lines

Close to the end of the creation process, all augmented elements had some offset. The offset seemed to be quite random, as it was sometimes just a bit on the left and sometimes even out of the field of view. Once no issues in the settings or other placing errors were found, the whole scene was transferred to a new one that solved the issue.

3 Future Improvements

To provide more recipes, some improvements to easily scale the project are needed. Therefore, the start scene should be able to

distinguish between multiple recipes and load the data accordingly. Furthermore, the instructions scene should only provide the primary UI then every feature augmentation is one extra scene, or several prefabs are prepared and instantiated during runtime. During testing, it attracted attention that multiple timers could overwrite each other. This should also be tackled by creating multiple timers, that are labelled with the belonging action. Besides, sometimes action needs to be performed before the timer starts; therefore, it could help to start the timer manually. As the last problem, the distinction of the steps needs to be reworked for several reasons. First, it is not scalable with different recipes and secondly, some augmentations would need more time and should be visible over multiple steps. For example, the step ”rolling out of the dough” should already show that the QR code should not be covered. Now, this is only visualized when the cut-out step starts, which is one step later.

For a real product, the hardware needs to be adapted as well. Here it would be great to have a giant engraved cutting board instead of a paper-based board. Maybe it could also be improved by another scaling like smaller marker-areas to use more space of the cutting board.

4 Conclusion

Providing an AR-cooking application could help untrained chefs to create tasteful dishes. Nevertheless, it still needs some research on the usability in real life. Here one issue could be the cutting process as mentions before, but also the covering of potential risks with augmented elements. For example the dishtowel may be too close to the hotplate, the user does not see the problem because currently the ingredients windows is open and the dishtowel catches fire.

References

[1] Alexander. Bell alarm clock sound effect.
[2] DinosoftLabs. Frame.
[3] Freepik. Stopwatch.
[4] MyNamePong. Onion.
[5] Unity. Unity engine.
[6] Vuforia. Vuforia engine.
[7] B. Zapatka. Vegetable dumlings (vegan gyoza/poststickers).