UNIVERSITY OF TARTU
Institute of Computer Science
Computer Science Curriculum

Hei Chun Shum

Futuclass AR

Master’s Thesis (30 ECTS)

Supervisor(s): Raimond-Hendrik Tunnel, MSc

Kristen Tamm

Tartu 2021

Futuclass AR

Abstract:

In this thesis, an AR educational game called *Futuclass AR* was created. The game is based on the current VR game *Futuclass Hub* — an educational virtual reality (VR) game available on Steam developed by Futuclass OÜ. The *Futuclass AR* is based on the DLC module *Reaction Balancing I*. *Futuclass AR* created in this thesis is for the 2 popular mobile device platforms — Android and iOS. *Futuclass AR* shares the same goal as its VR version which is to provide a new kind of interactive learning platform for improving 7-9th grade chemistry learning through Extended Reality (XR).

During this thesis, the requirements for *Futuclass AR* were created in interaction with the client (Futuclass OÜ). The main contribution was the game asset design, the prototype implementation and gameplay improvement through usability testing evaluation. The most valuable part of this thesis is the game design considerations about using different kinds of AR markers. AR markers have been designed and act as learning materials. They are the *Chemical Cards* — self-printed AR markers with different chemistry-related elements like molecules' name and chemical formulae. The main concept of the game design is to make use of the cards and the mobile screen to provide a fun, card learning game with AR contents.

Two usability testing sessions have been conducted with 13 7th grade pupils and 16 9th grade pupils correspondingly. There was an issue that the image tracking was not working as expected on some of the Android devices but it was unreproducible with our Android devices in internal testing. Due to the limited time scope, only iOS devices are tested for the 2nd usability testing session. Besides the above issue, feedback and some improvement suggestions were collected from the testing sessions. Fixes have been done to enhance the UX and improve the game flow. A mass-scale impact testing will be held in near future.

Thus, the thesis provides a good prototype for Futuclass OÜ to continue developing.

Keywords:

Computer Graphics, Augmented Reality, AR, Marker-based AR, Mobile AR, Educational Game Design

CERCS: P170 Computer science, numerical analysis, systems, control

Futuclass AR

Lühikokkuvõte:

Käesolevas lõputöös loodi hariduslik liitreaalsuse (AR) mäng nimega "*Futuclass AR*". "*Futuclass AR*" liitreaalsusmäng baseerub Futuclass OÜ poolt arendatud haridusliku virtuaalreaalsuse (VR) mängu "*Futuclass Hub*" allalaetavale sisupaketile (DLC) nimega "*Reaction Balancing I*". Käesolevas töös loodud "*Futuclass AR*" on mõeldud kahele populaarsele mobiiliplatvormile — Android ja iOS. "*Futuclass AR*" jagab oma VR versiooniga sama eesmärki, milleks on pakkuda uut interaktiivsed õppeplatvormi 7-9nda klassi keemia õppimiseks laiendatud reaalsuses.

Lõputöö jooksul püstitati suhtluses kliendiga nõuded "*Futuclass AR*" loomiseks. Põhiline töö panus on mänguvarade disainil, prototüüprakenduse implementeerimisel ja kvaliteedi hindamisel läbi kasutatavuse testimise. Kõige väärtuslikum osa töös on mängudisaini kaalutlused ja analüüs erinevate AR markerite kasutamiseks. AR markerid nimega "*kemikali kaardid*" on töös disainitud õppematerjali osana. Need on ise-prinditavad AR markerid, millel on keemia info nagu molekulide nimi ja keemiline valem. Mängudisaini tuum oli nende kaartide ja nutitelefoni ekraani kasutamine lõbusa, kaartidel põhineva ja AR sisuga õppemängu disainiks.

Kaks kasutatavuse testimise sessiooni viidi läbi vastavalt 13 7nda klassi kooliõpilase ja 16 9nda klassi kooliõpilasega. Tuvastati viga, et pildijälgimine ei töötanud vastavalt ootustele mõndade Androidi seadmete puhul, kuid seda viga ei õnnestunud korrata sisetestimise käigus. Töö piiratud skoobi tõttu kasutati ainult iOS seadmeid teises kasutatavuse testimises. Peale selle probleemi koguti testimiste käigus ka muud tagasisidet ning täienduste ettepanekuid. Tehti parandusi, et parandada kasutajakogemust ning mängu voogu. Massiline rakenduse mõju uuring on planeeritud lähitulevikku.

Lõputöö tulemuseks on hea prototüüprakendus ettevõttele Futuclass OÜ edasi arendamiseks.

Võtmesõnad:

Arvutigraafika, liitreaalsus, AR, markeripõhine AR, mobiilne AR, õppemängu disain

CERCS: P170 Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine (automaatjuhtimiteooria)

"That's what games are, in the end. Teachers. Fun is just another word for learning [1]."

(Raph Koster, 2013)

Table of Contents

1. Introduction

Futuclass OÜ[1] is a VR educational technology company based in Tartu. They are developing a VR educational game *Futuclass Hub*[2] for the VR headset platform. Currently, they are focusing on creating an educational game for learning chemistry. The main module available for their VR game — *Reaction Balancing I*[3] is about chemical reaction balancing, which targets the 7th to 9th-grade pupils. They move forward at a very fast pace and hope to create more value in interactive learning with XR technology.

Futuclass OÜ has the idea to build a mobile AR version of this chemical reaction game prototype and hopes to collect feedback to see if there is any educational value in it. *Futuclass AR* is a side project to them and is targeted at the mobile AR market (see Figure 1). The mobile AR market is expected to have significant growth in the coming years [2]. There are multiple supporting factors for exploring the mobile AR market. Firstly, the launches of the ARCore[4] by Google in 2018 and the ARkit[5] by Apple in 2015 have lowered the cost of AR applications development. Secondly, the release of the AR Foundation framework[6] in the cross-platform game engine Unity helps game developers to create AR games for both Android and iOS mobile platforms simultaneously.

Serious games are games that have educational purposes and see entertainment as an added value [3]. They have been found to be more effective in terms of learning and retention than conventional instruction methods like lectures and reading materials [4]. However, the study states that serious games are not more motivating than conventional instruction methods (P. Wouters et al. 2013). Thus, making the game more motivating is one of the challenges.

[1] https://futuclass.com/

[2] https://store.steampowered.com/app/1485370/Futuclass_Hub/

[3] https://store.steampowered.com/app/1485400/Futuclass__Reaction_Balancing_I/

[4] https://developers.google.com/ar

[5] https://developer.apple.com/augmented-reality/

[6] https://unity.com/unity/features/arfoundation

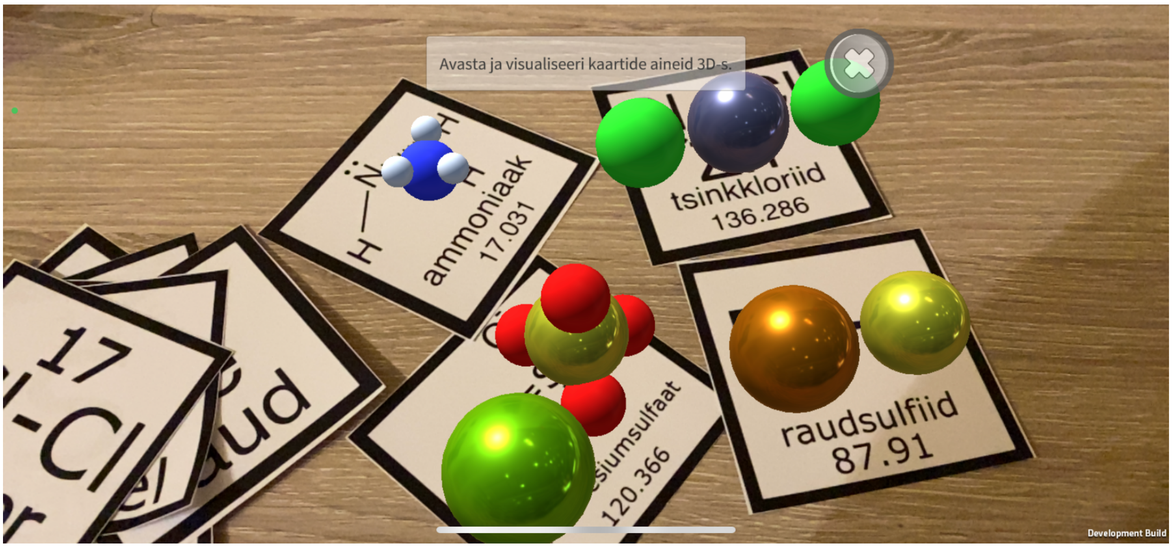

Figure 1. Screenshot of *Futuclass AR* gameplay on an iPhone

In the following chapter the background of *Futuclass Hub - Reaction Balancing I* is first explored to analyse the idea of re-designing it for AR. Existing AR solutions are investigated and the reasons for developing a custom AR application (*Futuclass AR*) are stated. As for the Design chapter, it covers the design principles, AR marker design and the game flow of *Futuclass AR*. Then, in the implementation chapter, the description of the game assets, issues during AR development and their solutions are discussed. After that, the plan and result of internal testing and usability testing are shown in the 5th chapter. Finally, the conclusion will be in the last chapter. In the Appendix, there is the glossary, accompany files (video of the gameplay, link of the repository, AR markers file), screenshots of the game and the version history.

2. Futuclass Hub - Reaction Balancing I

Futuclass Hub is a VR educational game for 7th-9th grade pupils on the VR headset platform. There are different modules, which are focusing on chemistry and physics. In this thesis, we describe the *Reaction Balancing I* module of the game because it is the first paid content released as downloadable content (DLC) for the hub. According to Futuclass, the impact testing result for this module shows that it provides solid educational values. Thus, the top priority is to create a prototype of this module in AR.

Figure 2. Player grabbing a H_2O molecule in *Reaction Balancing I*[7]

The chemical reaction balancing module is a quiz based game with several levels. Each quiz consists of an unbalanced chemical equation. The player needs to choose the right type of molecules and the correct numbers of them to balance the equation. They could complete this by grabbing molecules generated from the spawning zones and move them to the answering zones as shown in Figure 2. Once the player completes the balancing, he/she could press the confirm button to check the answer as shown in Figure 3. If the numbers of the reagents and the products are correct, then a congratulation message will be shown to inform and motivate the player. If the answer is not correct, then the player could retry by modifying the numbers of reagents and products by grabbing more of them to add to the zones or removing them from the zones.

[7] Video of the gameplay in reaction balancing I - Futuclass Hub: https://www.youtube.com/watch?v=2bS3v0r_oR4

Figure 3. Player pressing the confirm button in VR[8]

The chemical equations are divided into 8 levels. Players need to make 3 correct answers in a row in order to advance a level.

The game is designed to have an answer submit button. Players could add or remove molecules freely before submission. The quiz is not timed. It is believed that players could learn how to balance the equation through practical movements of grabbing molecules. With manual verification but not automatic answer checking, players have chances to add and remove molecules with their own paces.

Overall, the *Reaction Balancing I* module is a good example of implementing some of the educational game design principles about learning and exploration developed by Laine et al [5]. They are analysed and discussed in detail in chapter 3. Design. After looking into the reaction balancing in VR, we will look into some existing AR solutions of chemistry education and see why we have to create *Futuclass AR*.

[8] Video of the gameplay in Reaction Balancing I - Futuclass Hub: https://www.youtube.com/watch?v=2bS3v0r_oR4

2.1 Alternative AR Solutions

There were some mobile AR chemistry educational games currently in the market. However, it is found that there are no such games featuring reaction balancing. And, though the existing solutions already provide good chemical molecules' visualisation in AR, they did not create games with the AR content but made use of the AR content as a visualisation sandbox only. Two solutions - *MoleQL*[9] and *Aroloon Chemistry*[10] each with different designs are introduced.

2.1.1 MoleQL

MoleQL is a mobile AR prototype for chemistry education by Cross Motion. It targets pupils at least grade 5th who have not yet started or just have started learning chemistry. Its target group is different from ours. Users could use their phone cameras to scan the printed paper cards to visualise chemical elements and reactions (see Figure 4).

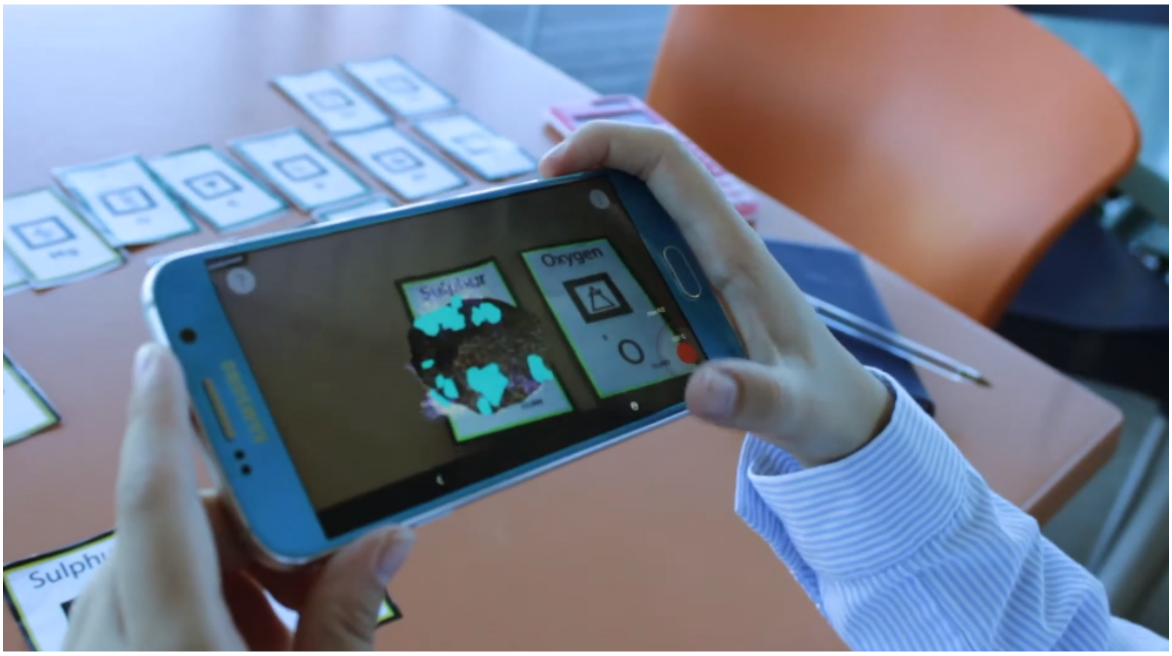

Figure 4. Scanning the AR marker[11]

The main interactive element is the spawning of reactions. The 3D models of chemical elements will be visualised when the device is able to track the corresponding AR marker successfully. The application provides some buttons of different chemical elements for the

9 https://www.crossmotion.org/moleql/

10 http://www.arloon.com/apps/arloon-chemistry/

11 Video of gameplay of MoleQL: https://www.youtube.com/watch?v=h0M_AoqQtFY

user to press. When the user presses the element button, the reaction between the 2 elements will be visualised in AR. Unlike *Reaction Balancing I*, it is not a game with different tasks and levels but more like a visualisation software.

2.1.2 Arloon Chemistry

Arloon Chemistry is one of the AR educational apps created by Arloon[12]. Arloon is a studio focusing on AR educational software development in Spain. *Arloon Chemistry* has demonstrated a design combining the on-screen mobile game with a 3D molecule visualization mode in AR. This application consists of 3 different learning sessions.

The first session is the molecule database. Users can select the chemical compound on the screen and learn about the molecule's 3D structure. Information about the molecule is also displayed on the screen as reading material.

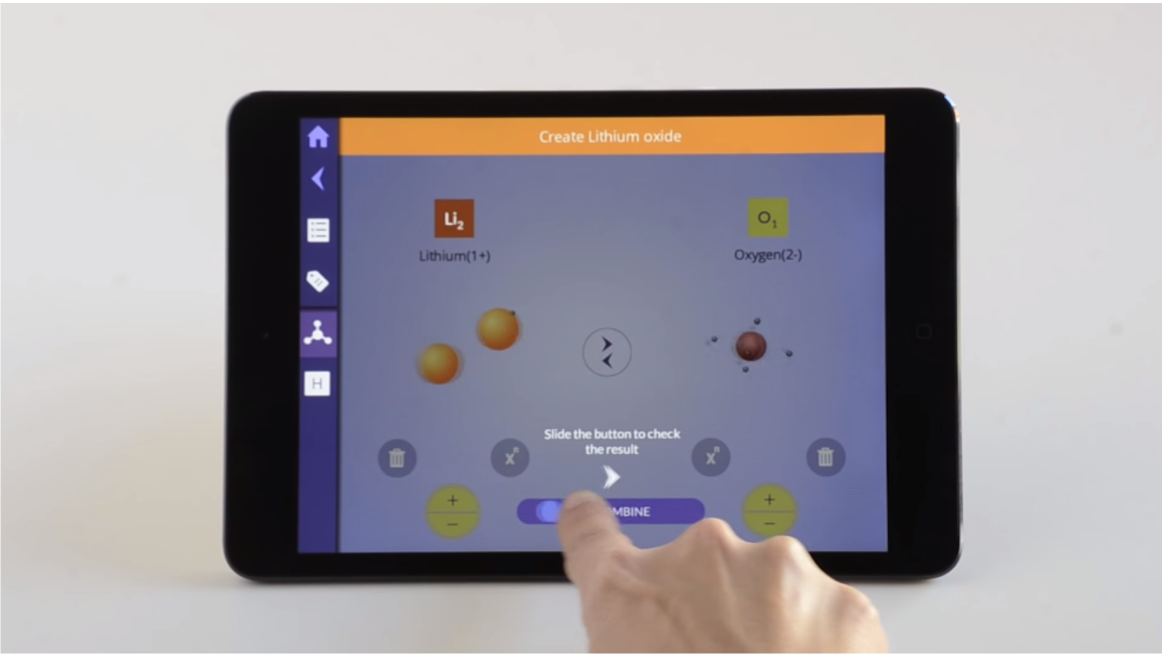

Figure 5. Forming chemical compound[13]

The second one is the making molecule session. Firstly, the application provides an interactive periodic table to users. Users can select an element from the periodic table and then choose a chemical compound that includes the element to start playing the molecule forming game. It is a mobile mini-game where users can add or drop elements by pressing

---

[12] http://www.arloon.com/en/

[13] Video of gameplay of Arloon Chemistry: http://www.arloon.com/apps/arloon-chemistry/

the buttons on the screen (see Figure 5). If the numbers of the elements are correct, then users would be able to get the chemical compound as a result.

Figure 6. Visualisation of a 3D molecule model in AR[14]

As for the AR session, it gives a simple AR visualisation functionality. A generic printed placement card is provided. Users could select what elements they want to visualise by choosing them from the screen. Then, users could point their phone cameras to the placement card. When the phone recognises the printed card, the corresponding 3D molecule model would be displayed in AR on the card.

Both AR solutions demonstrate two different approaches in combining augmented reality and chemistry learning. *MoleQL* made use of a lot of markers in its design with AR contents. However, it is not game-like but much like a visualiser. *Arloon Chemistry* provides better game-like features like forming molecule tasks. However, the tasks and the learning materials are all presented on-screen but only molecule visualisation is on the universal AR marker placement card (see Figure 6). It does not make full use of AR. Additionally, both of them do not provide any tasks about reaction balancing. Thus, our design for *Futuclass AR* is needed. In the next chapter, the design is discussed in detail.

[14] Ibid

3. The Design

Although AR and VR share some similar computer graphics technologies, they are highly different by nature. VR focuses on letting the player immerse into the virtual and isolate them from the real world. In contrast, AR focuses on the interactions between the virtual contents and the real-world environment. In *Reaction Balancing I*, players could be immersed in a virtual laboratory world. With the help of the Oculus hand tracking system[15] in Unity, grabbing atoms with their fingers and moving the molecules is possible inside the VR space. In contrast, AR focuses on the interactions between real-world environments and computer-generated contents. However, the same content and interaction logic from VR could not provide the same experience to players because of the limitations of AR. Thus, the design should be able to make good use of AR and provide sufficient educational value.

3.1 Gamification and Game-based Learning

The design process of this AR learning game *Futuclass AR* is gamification, which is defined as a process of applying game design elements to non-game contexts [6]. In *Futuclass AR*, the chemistry learning process is the main context to be turned into a game. Game elements and mechanics like tasks, points and colourful 3D models were added to the process. Another related term is game-based learning (GBL). They are very similar but different. Game-based learning is using games as part of the learning process. For instance, pupils could learn some stores of ancient Greek philosophers when playing the video game *Assassin's Creed Odyssey*[16].

As a gamified content, *Futuclass AR* was not a game at the very beginning. However, Kapp (2012) notes that the goals of gamification and GBL are basically the same, which are trying to solve a problem, motivate, and promote learning using game-based thinking and techniques. *Futuclass AR* can be considered as a game[17] when the user feels like he/she is playing it.

---

15 https://developer.oculus.com/documentation/unity/unity-handtracking/

16 https://www.ubisoft.com/en-gb/game/assassins-creed/odyssey

17 https://www.merriam-webster.com/dictionary/game

3.2 Design Principles

The chemistry learning process is what should be focused on in this thesis. To design a fun game with sufficient educational value, the design is educational-focused. there are some constructive design principles developed by Laine, T. et al. we should comply with.

Favour simple challenges over complex challenges. It is suggested that if the gaming task is too complex, it will discourage the player. Therefore, it suggests breaking complex challenges into smaller tasks. In the AR learning game, a learning topic like reaction balancing should be broken into different small tasks in the AR format. Thus, players could focus on completing a single task at a time and easily learn how to complete the more complex task.

Provide enough time to solve challenges. Providing sufficient time is helpful for reflection in learning. In *Reaction Balancing I*, the reaction balancing task has no time limit. The same practice should be applied in its AR version.

Freedom of exploration and experimentation. The principle is about creating a game environment for the players to explore in a comfortable manner. It does not restrict the world to be real or virtual. In the AR learning game, if a marker-based AR solution is chosen (see Chapter 3.3) the AR markers could be one of the learning materials. The objective of this is to make players interact with the printed AR markers in the real world while playing the game.

Provide instructions and/or tutorials. This is an important principle, especially for an AR learning game design. The components of user input of mobile AR games not only include the UI components on the screen but also the virtual objects in AR and maybe the physical AR Markers. It is more complex than just a physical oriented game. The players have to learn when and where to interact with the AR contents to complete the game challenges. In this prototype, enough textual instructions would have to be provided to players before each game mode. These could be replaced by infographics or short videos in the future for further improvement.

3.3 Augmented Reality

Augmented Reality (AR) focuses on providing an interactive 3D experience that combines the view of the real world with computer-generated elements [7]. To properly implement the benefits AR could provide in an educational game, a proper design is undoubtedly important.

There are two main types of AR, which are image-based AR and location-based AR. The former one mainly makes use of graphical information from the device's camera. Then, the AR application would display some virtual objects and perform some actions based on the application's design. The latter one uses the locational information, network information and sensor information from the devices to perform simultaneous localization and mapping (SLAM). By using SLAM, an AR object could be shown an accurate GPS coordinate, perform some action and even move to a different coordinate.

For our case, image-based AR is chosen. Geographic data is not relevant since the main content of the game involves chemical reactions and elements. And, we could make use of the AR reference image to make them as educational content.

3.3.1 Image-based AR

With the evolution of smartphones, more and more powerful AR applications have launched into the mobile application market. Image-based AR is separated into two types: marker-based AR and markerless AR. Table 1 shows the differences between them.

Table 1. Marker-based AR vs Markerless AR.

Image-based AR types	Marker-based AR	Markerless AR
Features	- Requires a static marker: 1. black and white marker, 2. detailed image with enough unique features. - By scanning the marker, some AR contents will be triggered (etc 3D models, videos and animations) near the marker.	- No preparation of customized markers. - Plane detection. - Face detection. - By tracking the detected surface, the AR contents could be triggered and placed around the detected area.
Pros	- Easy to trigger. - Markers could be informative.	- More flexible. (no physical markers needed)
Cons	- Need markers.	- Requires higher computing power.

Both approaches have their advantages and disadvantages. Markerless AR is favoured for 3D model visualisation with plane detection since no physical markers are needed (see Figure 7). For our use-case, marker-based AR has been chosen. For AR learning games, physical AR markers could be a part of the learning contents together with the AR content. AR triggered contents like image, audio, video and 3D models could make a good combination with the AR markers. 3D models are frequently used in AR learning games [8].

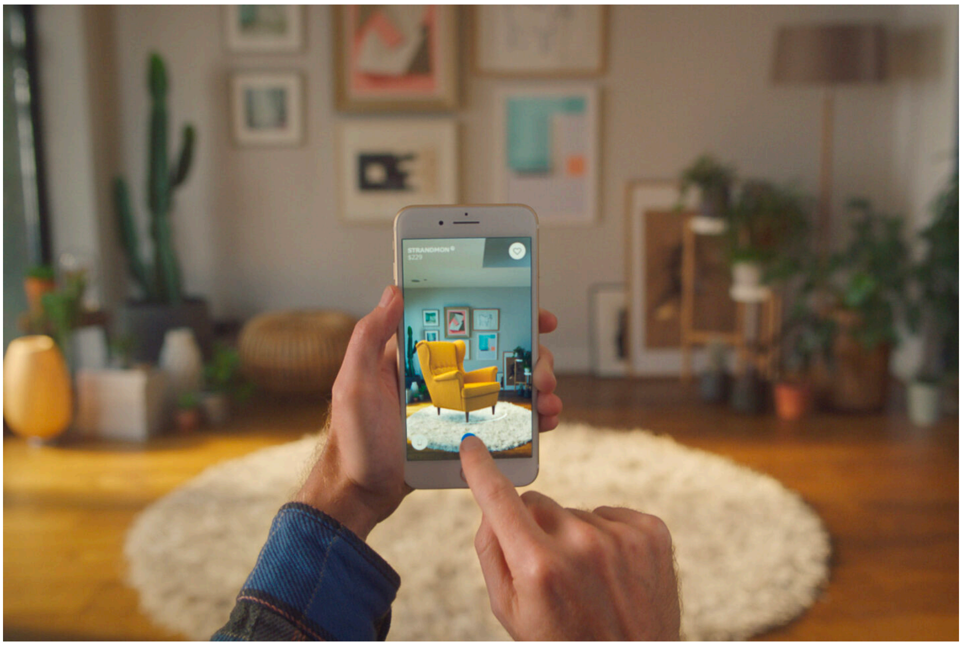

Figure 7. Example of markerless AR application (IKEA Place[18])

For making a chemical reaction balancing game both approaches (marker-based, markerless) could work. AR marker-based approach is chosen over markerless approach due to the possible added educational value. Without markers, there would be fewer interactive elements of the game and fewer interactions to the real-world environment. There are existing AR learning games that contain many non-AR mobile game features with only a few AR features (eg. *Arloon Chemistry*). *Arloon Chemistry*'s non-AR part could be implemented as markerless AR content. However, it might make the game control more complicated. **Directly porting games to AR does not guarantee a better player experience in many cases.** In their design, only a simple visualisation functionality has been implemented with markerless AR. In our case, we make full use of AR markers to provide a fun, interactive learning game.

[18] https://www.ikea.com/au/en/customer-service/mobile-apps/say-hej-to-ikea-place-pub1f8af050

3.3.2 2D Image Tracking

2D image tracking is being used intensively in this AR learning game. It consists of 2 steps, which are **creating reference image library**, and **image tracking**.

There are a few important notes about the creation of the reference image library. In AR marker application, the AR markers are being used as the reference images. A computer vision algorithm extracts features of them and stores them in a reference database. In many AR toolkits, only grayscale information is used (eg. Google ARCore, Apple ARKit, Wikitude). As a single-channel image, it makes the program require less computational power than a multi-channel image like an RGB image.

Since only grayscale information is used, a reference image being good or not is determined by the features. To reach a good image tracking performance, preparing good reference images which provide enough unique features is important. According to Vuforia[19], the feature is a sharp, spiked, chiselled detail in the image. It is not known what exact computer vision algorithms are used for image tracking in ARCore and ARKit. However, the goal of different feature detector algorithms like Harris Corner or FAST is the same, which is to extract corners as features (see Figure 8).

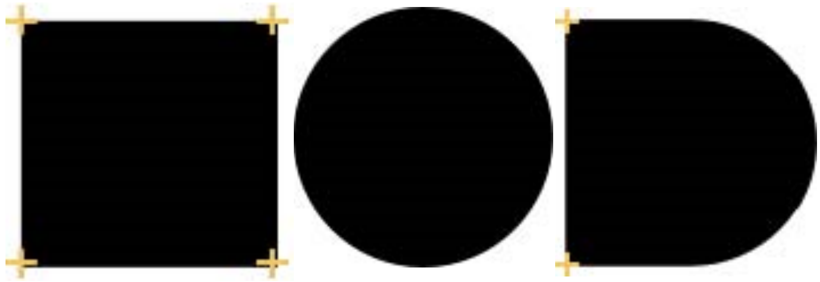

Figure 8. Features at the corners

After preparing the reference image library, the program will perform image tracking. Looking for the clusters of features, the AR toolkit could understand the real-world environments and update its knowledge. In 2D image tracking, the reference image is defined as trackable. Trackable is a collection of features at this point. If the system is confident in the tracked information, the trackable will be defined as tracked and the AR content will be triggered according to AR program input. AR Developers could define

[19] https://library.vuforia.com/features/images/image-targets/best-practices-for-designing-and-developing-image-based-targets.html

what to trigger when the trackable is being tracked, being moved or when losing the tracking.

3.3.3 Mobile AR

The mobile platform has become a popular platform for augmented reality applications due to its ease of accessibility and hardware evolution. According to the mobile augmented reality market research, mobile AR SDKs ARCore and ARKit have helped widespread adoption of mobile AR, and the gaming and lifestyle applications have become the most downloaded application sectors [2]. There are common challenges for mobile AR development. Firstly, the user has limited access with defined boundaries due to the small screen size of mobile phones. Secondly, there are social and privacy issues. To secure user identity and privacy is essential but there are no regulations regarding that in the AR environment.

3.4 AR Markers

AR markers are the most essential key in *Futuclass AR*. Their design is important since they have two main functions: the learning content and acting as the AR content trigger for image tracking. Both of these are important and to reach a workable design, several iterations for the markers were done during the development.

3.4.1 Design Guideline

To produce good AR markers with good tracking quality, it is good to follow some existing design guidelines. There are detailed guidelines about how to prepare a good reference image from Google ARCore and Wikitude. Table 2 shows the summary.

Table 2. Guidelines of targeted images for image tracking.

Image-based AR types	ARCore	Wikitude
Pixel	At least 300x300 pixels	Between 500 and 1000 pixels (width or height)
Compression	Avoid heavy compression	N/A
Color information	Not used	Not used
Physical information	Specify it to improve tracking especially for large image (>75cm)	N/A
Amount of features (empty space)	Avoid images having large amount of geometric features or very few features	Single-colored area not providing suitable graphic information for tracking. Crop the most prominent part of the image
Repeating Patterns	Avoid	Avoid
Quality Score Toolkit	Arcoreimg tool to get the image quality score (0 to 100) Recommend a score of at least 75	Star rating system. (0 to 3 stars) Test the image which gets at least 1 star

3.4.2 Marker Design

In *Reaction Balancing I*, players could pick molecules with their hands. The first idea to be proposed was to create some markers which could represent the molecules. They are called the *Chemical Cards*, which are the physical AR markers filling with learning materials like chemical symbols. To create workable AR markers more efficient, some good AR target image rating tools that are free to use have been found. In this thesis, the `arcoreimg` tool and the Vuforia development portal[20] have been used. Vuforia portal provides a GUI online rating system from 1-5 starts for the target image. It is for reference only since the rating system is for vuforia builds but not for ARKit or ARCore. Still, it provides a feature points visualisation for us to analyze our images. As for `arcoreimg`, it is a command-line tool provided by ARCore for rating the image from 0-100 score. Images with at least 75 scores are good images for ARCore image tracking.

[20] https://developer.vuforia.com/

Design version 1

The first design uses the chemical symbol as the main component of the AR marker. The marker is shaped like a poker card. It is a simple design but the symbol itself is not providing enough feature points for devices' image tracking. Thus, the molecules' pictures are placed randomly at the top and the bottom of the card. This initial design is rated 5 stars for "Augmentable" in the Vuforia portal (see Figure 9).

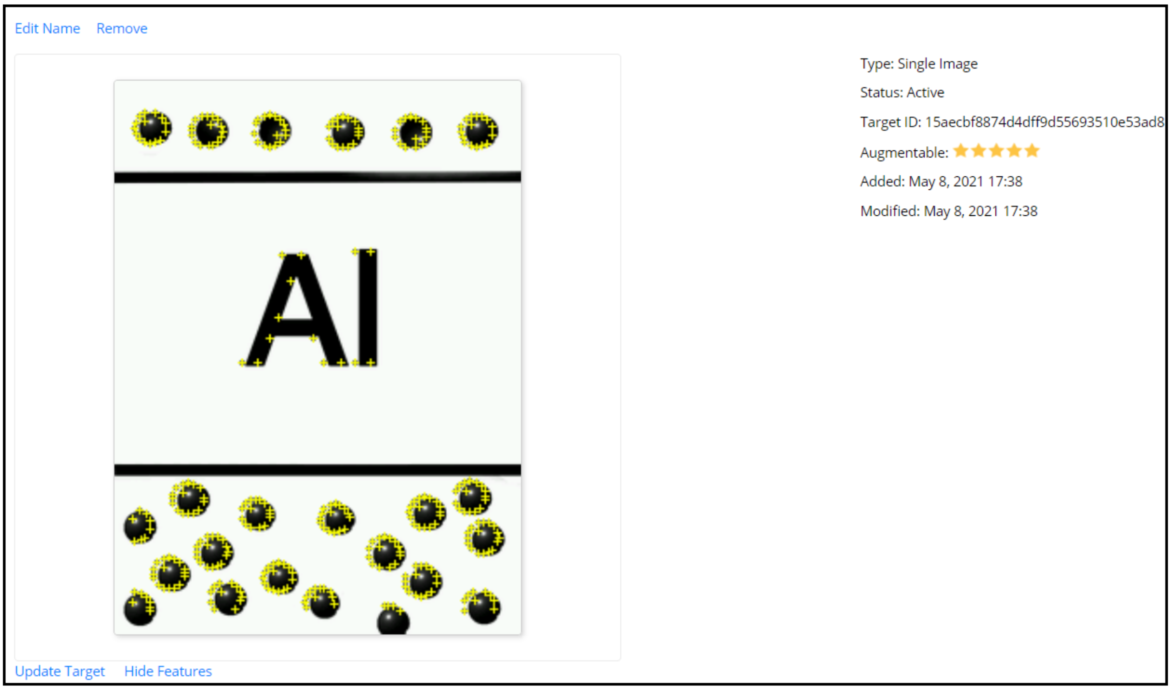

Figure 9. Screenshot of an AR marker with its rating and feature points on Vuforia

However, the initial design got a result of "`Failed to get enough keypoints from target image.`" from the `arcoreimg` tool. It means that the ARCore would not be able to track this image. From this point, it is suggested that the "Augmentable " rating information provided by the Vuforia portal is not a good reference for our design. Instead, the `arcoreimg` tool score rating is more important to our design.

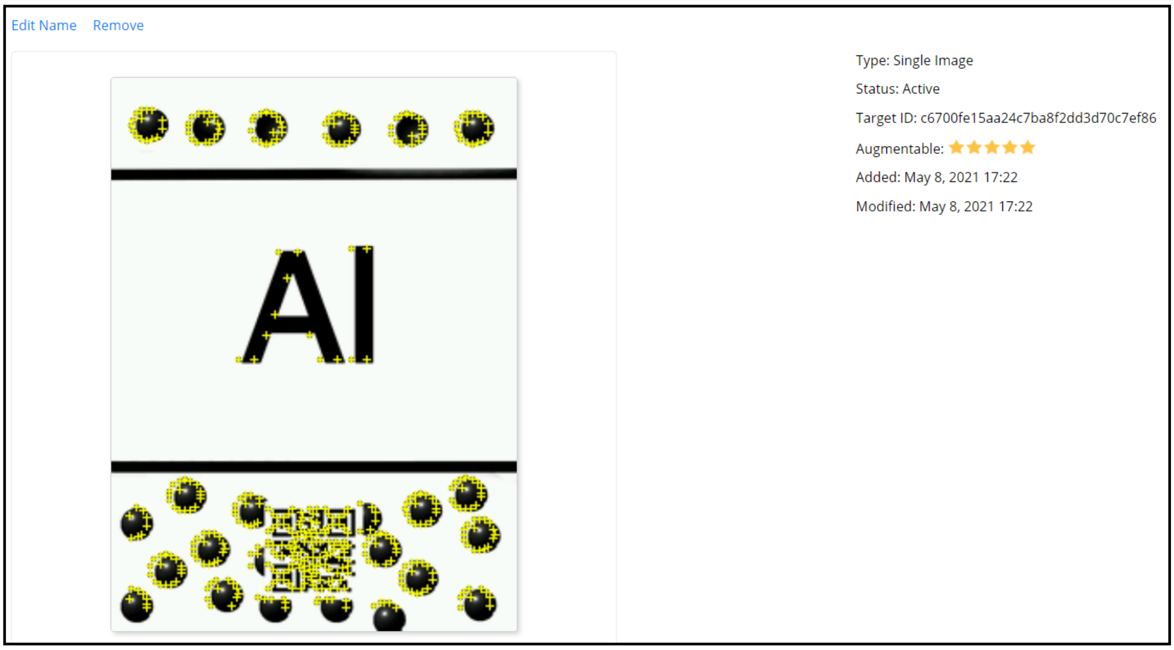

Figure 10. Screenshot of an AR marker with QR code on Vuforia

To make the AR markers work, more feature points have to be added. We decided to add QR codes to the markers since they could be generated fast. The markers with QR code has scored 100 out of 100 points in `arcoreimg` tool and with 5 stars rating for "Augmentable" in the Vuforia portal (see Figure 10). The updated markers have a lot of feature points showing around the QR code, which become the main sources of key feature points for ARCore. Thus, this design is adapted and QR codes have been added to all the markers. It seems that the feature points of the molecules of the previous do not provide useful information for ARCore to track.

Design version 2

In the second iteration, the molecule images have been removed from all the markers. QR code has been enlarged to fill the space of the marker. This design provides enough unique feature points. Thus, the markers could get high scores in `arcoreimg` tool (higher than 75). but it does not provide enough learning content as learning material.

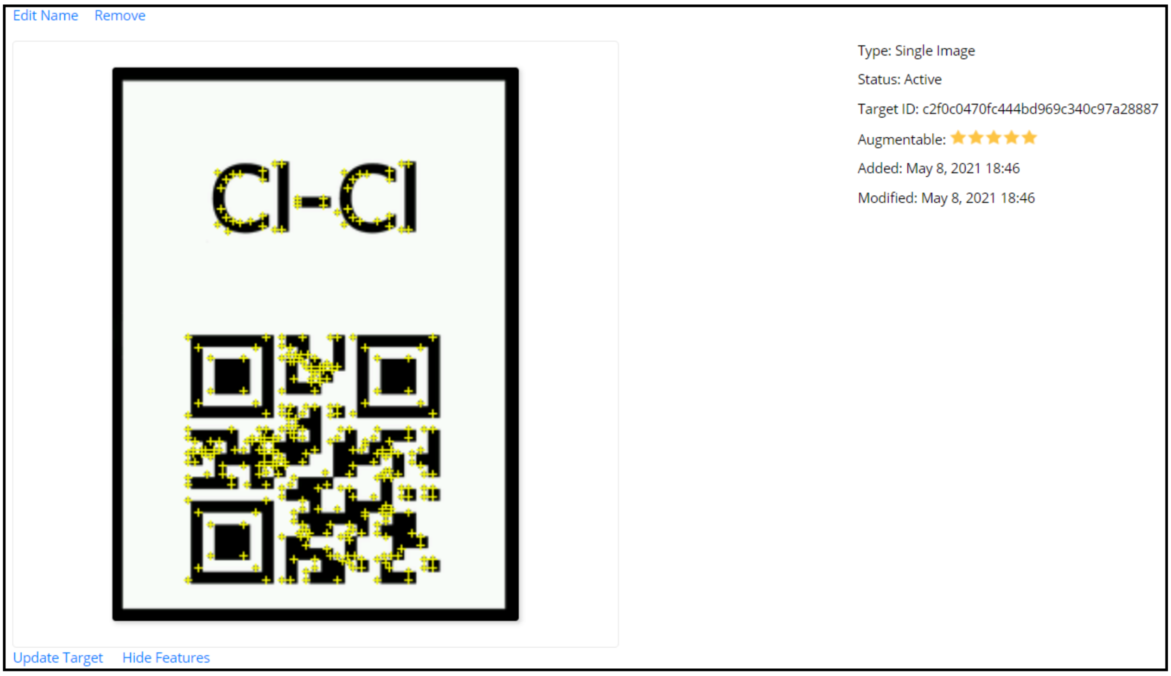

Figure 11. Screenshot of an AR marker without molecule pictures on Vuforia

A card border is added for making it more like a card. It also helps players to cut out the card after printing. The card border element is included in all marker designs starting from this version. This design has been used during the early stage of game development.

Design version 3

To make the card a better learning material, unnecessary visual content should be removed like the QR code. The third design was made with that purpose in mind. QR code has been removed. In order to create more unique feature points per image, adding more elements to the marker are necessary. The names of chemical elements and additional information have been added to the card for improving image tracking and as part of the learning material. During the creation of the markers, the markers are tuned based on the score of `arcoreimg` tool. The baseline is getting at least 75 score. To conclude, it is a better design in terms of educational values with sufficient key feature points for tracking. This was the design version that was used in the usability test (see Chapter 5.2). All the markers are in the marker file (see Appendix - Accompany Files).

Figure 12. Element card

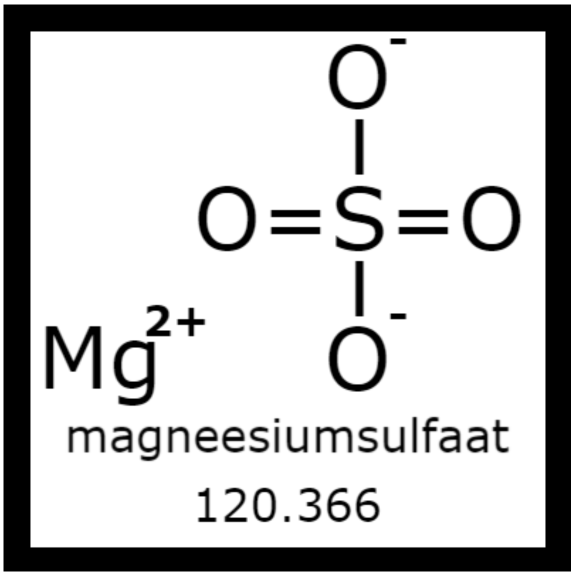

Figure 13. Molecule card

Two types of AR marker cards of that design are defined: the element card and the molecule card. In regards to the element card, the atomic number is at the top of the card and the name is at the bottom of the card (see Figure 12). As for the molecule card, the name is displayed below the chemical symbol and the molar mass is at the bottom of the card (see Figure 13). The card shape is also changed into a rectangle to reduce empty spaces.

Design version 3 (Find the Correct extended version)

It is a variation of version 3 for the modified Find the Correct game mode which was tested during the second usability test. The contents of the cards have been changed according to the task types of different levels of the game. The cards are now grouped into 3 collections: the formula card (see Figure 14), the name card (see Figure 15), and the characteristic card (see Figure 16).

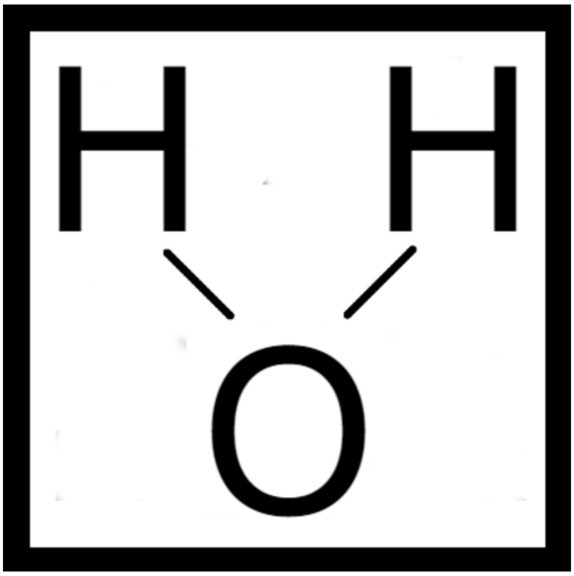

Figure 14. Formula Card

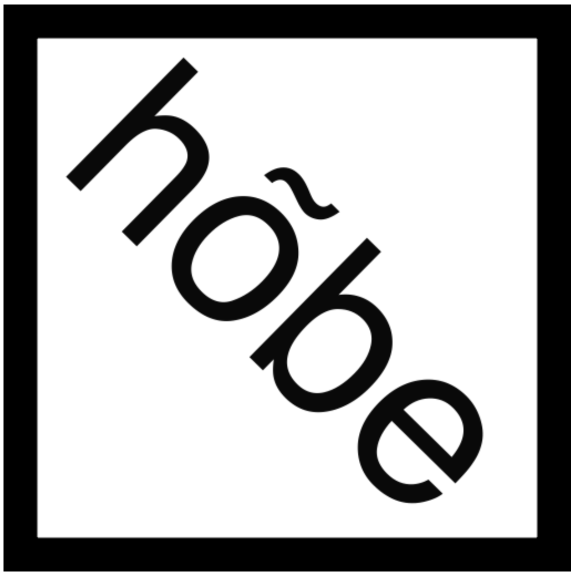

Figure 15. Name Card

Figure 16. Characteristic Card

They are created for different types of tasks in that game mode. The arrangements of the elements inside the cards have been adjusted according to the result of the `arcoreimg` tool to obtain scores higher than 75. In the next section, we describe what can be done with these cards.

3.5 Game Rules

Game is a kind of structured form of play [9]. There are many different types of games nowadays existing everywhere like board games, sports, card games and video games. Game design is an important field since it is the keystone to provide fun and entertainment to players. In our case, we also want to convey the educational value in addition to the fun/ entertainment.

According to Robert Zubek [10], game design involves three main elements: mechanics, gameplay and player experience. Mechanics include the rules, different assets in the game. Gameplay is the interaction between the player and the system. And the player experience, which is the most we take care of, means how the players feel when they play the game [10].

In this session, the game rules will be discussed. During the design phase, 2 more modules other than the reaction balancing have been proposed since they share similar technical requirements. Thus, there are 3 modules in this game, which are the ***Molecular Sandbox***, ***Find the Correct*** and the ***Reaction Balancing***. The following game design illustration figures were created using Figma[21] and Miro[22]. In the following figures, the white background of the device screen means that it is a non-AR scene. If it shows a green background, it is an AR scene with the real-time device's camera's view running as background. To improve the player experience continuously, iterative design[23] has been applied through the whole development process.

3.5.1 Molecular Sandbox

It is the simplest module. Users point the camera to a *Chemical Card*, and the corresponding molecule's 3D model is visualised in AR (see Figure 17). By tapping the molecule model on the screen with a finger, the user could read the name of that molecule on the screen.

[21] https://www.figma.com/

[22] https://miro.com/

[23] https://en.wikipedia.org/wiki/Iterative_design

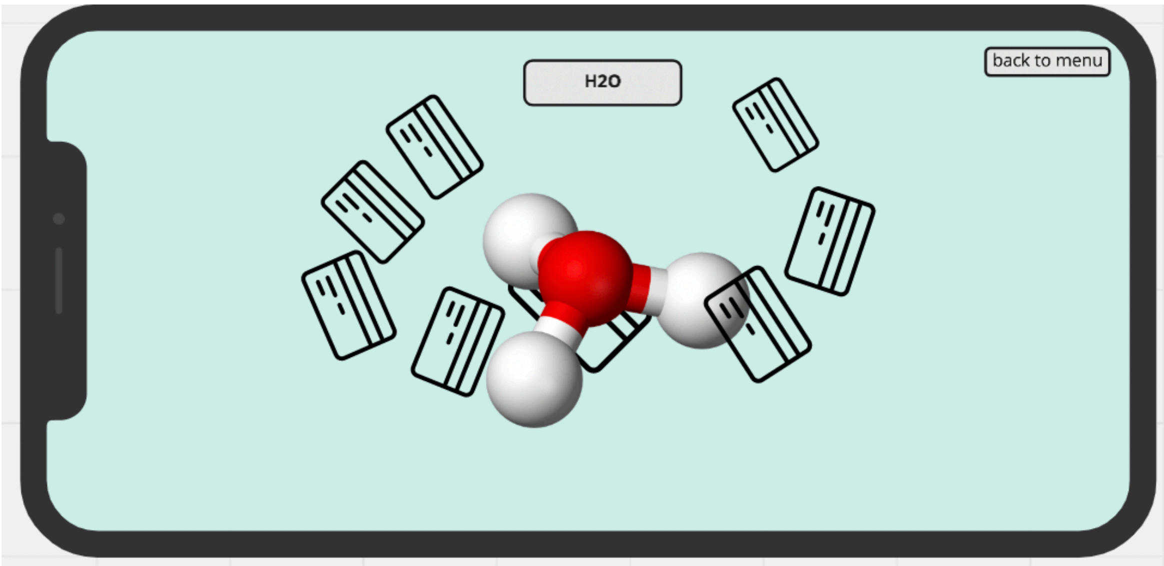

Figure 17. Molecular Sandbox

This is an example of an AR visualization but not an AR game, similar to the work of MoleQl (see Chapter 2.1.1). There are a lot of possibilities to make it more informative and interactive. For instance, more information can be displayed while tapping different molecules like text paragraphs and videos. In addition, buttons for resizing the molecules in AR can be added to let the user observe them from a different angle. There were considerations about how many molecule models should be visualised at once. If only one molecule is visualised at once, the user might be able to focus more on the visualised one without distraction. As for visualising multiple molecule models, the user could compare the models and observe the differences between the models. In this thesis, the multiple model visualisation has been chosen. In the future, this option could be implemented as a user setting and let the user choose between them.

3.5.2 Find the Correct

This is a task-based game mode. A task is being presented at the top of the screen, then the player needs to find the correct molecules. To search for it, they need to point the camera towards different cards. The detected cards visualise their virtual molecules in AR. The player chooses the one they think is the one they were asked to find by tapping on the virtual molecule. If the answer is incorrect, the player can retry. If the answer is correct, the fun fact about that chemical substance will be shown on the screen. Players can read it before heading to the next task.

As for the tasks, a simple task design is created initially. Given a chemical symbol, players need to find the corresponding *Chemical Card*. This makes the game easy to learn and easy to complete.

In early versions, a validate button was added to the screen for submitting the answer. Same as the sandbox mode, tapping to choose molecule functionality was introduced to this game mode to replace the pointing camera mechanism in version 0.0.4 (see Appendix - Version Control). The player can choose the right answer by just tapping the molecule or the card on the screen. In addition, a new feedback system was implemented. The feedback responses are as below:

- When the answer is correct, the task box will flash twice in green.
- When the answer is incorrect, the task box will flash twice in red.

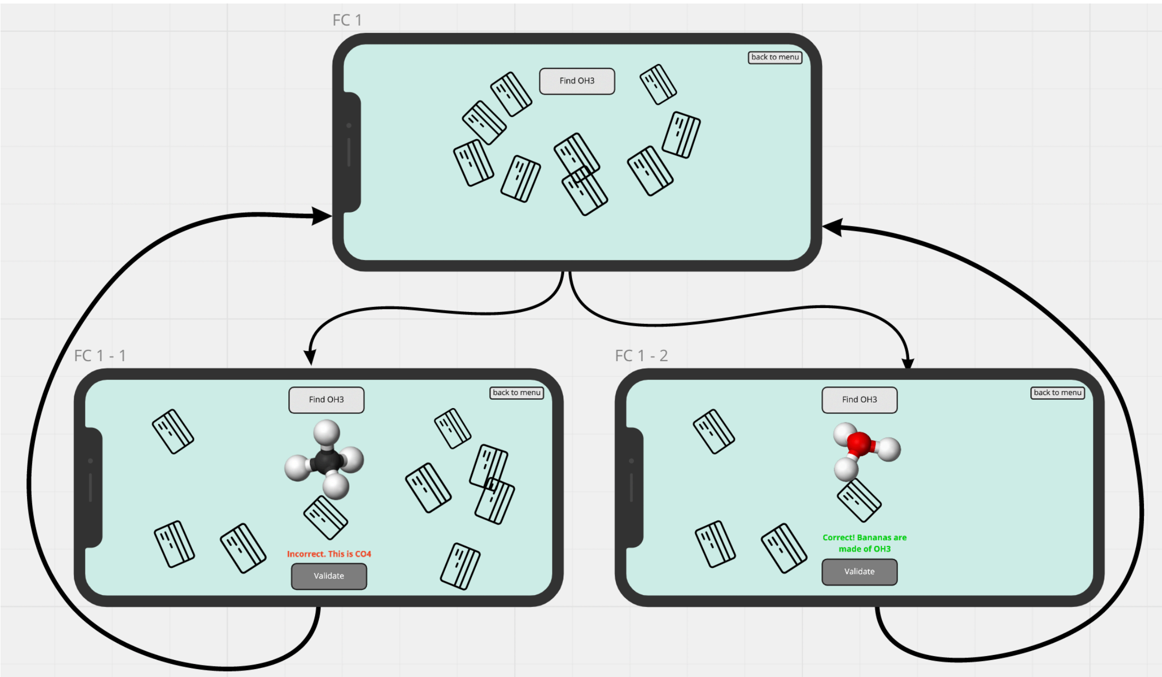

Figure 18. Find the Correct game flow

Figure 18 shows the basic flow of this game mode. The core mechanics is to create a task and let the player make use of the AR markers with the AR content to complete the task. This design of this game mode has been improved by adding more game elements like game levels and different groups of cards after usability testing review (see Chapter 5.2.3).

3.5.3 Reaction Balancing

In this mode, the player could choose what chemical reaction collection they want to balance through the game menu (see Figure 19). A reaction equation will be set as the task randomly. There are two buttons for the combination reaction collection in this prototype. It allows the player to balance the equations with two different designs. They are discussed in this chapter later.

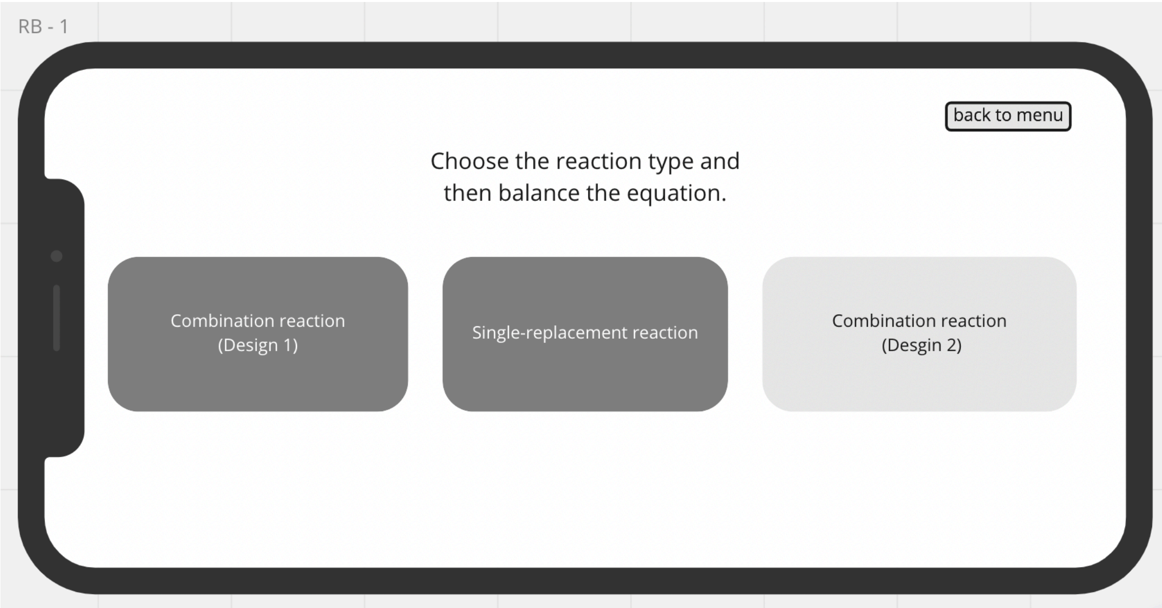

Figure 19. Reaction type menu

The app will then go into the AR mode, and it will guide the player to find the card of the reagents and the products of that equation. The player has to choose the card with the devices by tapping the shown AR chemical substance model or the card on the screen. In Figure 20, the screens RB 1-1 and RB 1-2 are the first step which is to find the reagent and the screen RB 2-1 and RB 2-2 are the second step for finding the product.

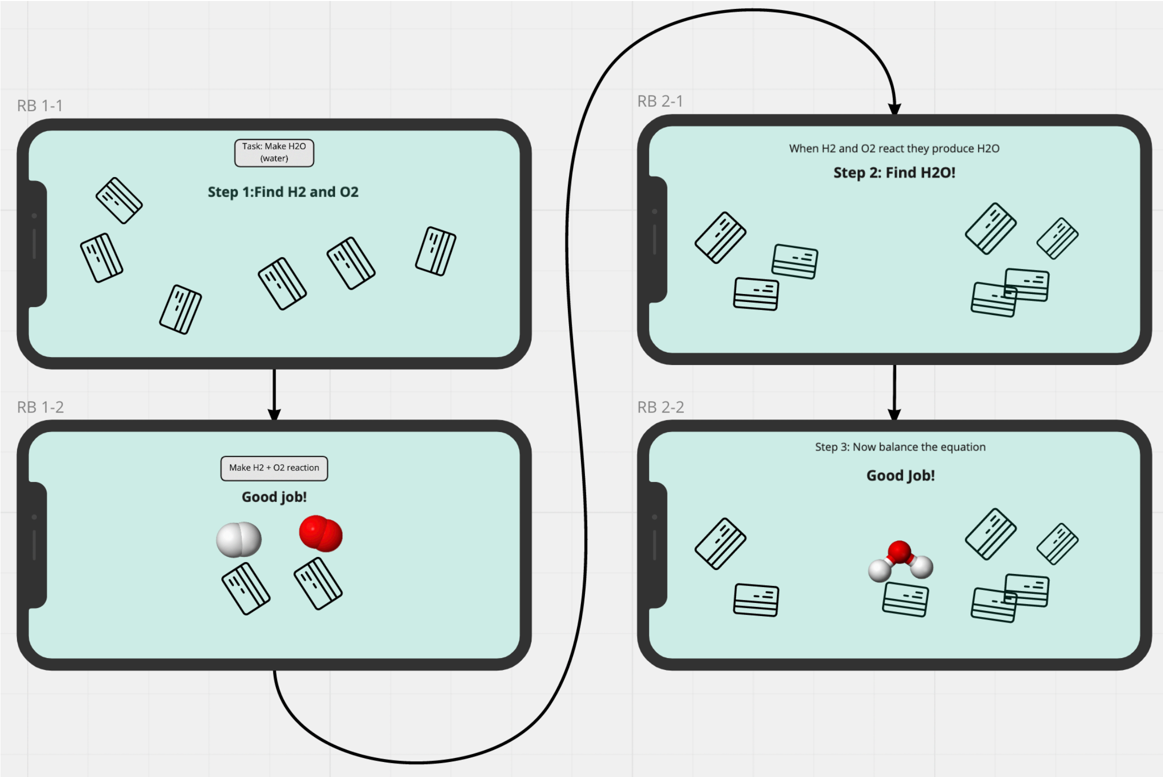

Figure 20. Flow of finding reagents and products

After finding the correct cards for reagents and products, there will be the reaction balancing task. In this game, 2 different versions have been designed which are the on-screen Reaction Balancing and AR Reaction Balancing. Depending on which mode the player had chosen in the menu (see Figure 19), the screen will go to either RB 3-A (Figure 21) or RB 3-B (Figure 22). A comparison testing is done in the second usability testing (see Chapter 5.3).

On-screen Reaction Balancing

After step 2, the app goes into non-AR mode. The unbalanced equation is displayed on the screen as shown in Figure 21. The reagents and products are having their 3D models displayed on the screen. Buttons for adding and removing the molecules are provided accordingly. When the player adds the molecule, the number of the molecule and the number of 3D models will increase and vice versa. The player needs to balance the equation by putting the right amount of reagents and products. The player can tap the validate button to check if the answer is correct or not. The maximum available coefficient is set to be 6 since it is not possible to visualise too many molecules on the mobile screen and most of the balanced equations do not need a higher coefficient than 6.

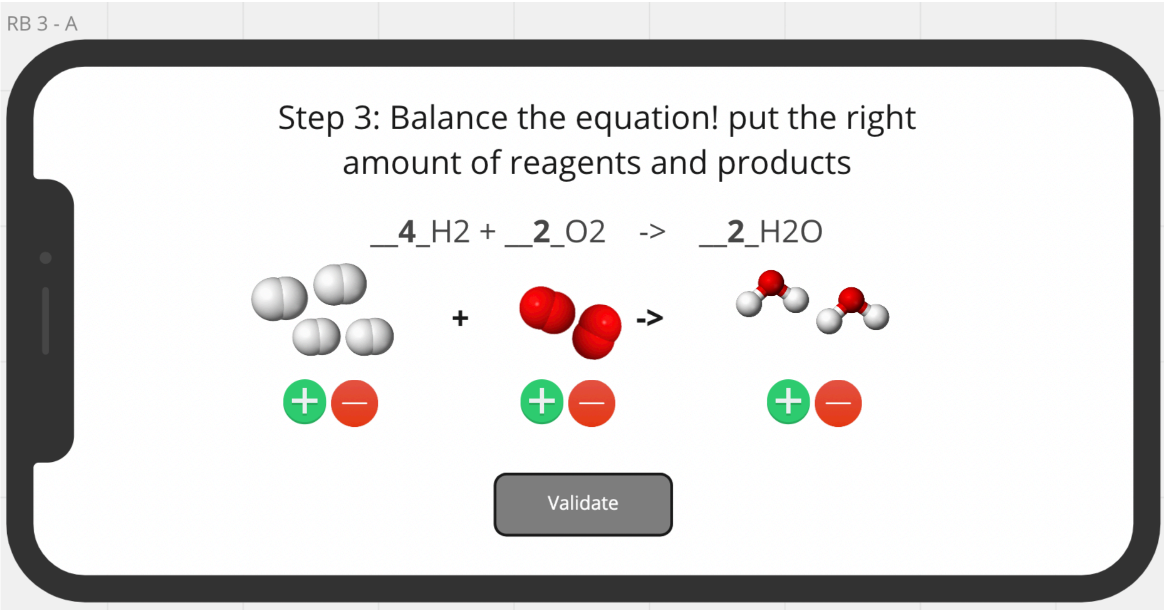

Figure 21. On-screen Reaction Balancing

It is an on-screen game after AR sessions asking the player to find reagents and products. This made the design of this game mode hybrid. It combines the non-AR sessions with AR sessions. This is similar to the design of *Arloon Chemistry*. Since the last step of this game is the hardest, the player has to spend more time solving it. Considering the stability of AR image tracking and battery usage, this design might be a good one to ease the above issues. The downside of this design is not providing a good AR experience compared with the previous steps.

AR Reaction Balancing

This is an AR scenario. After completing step 2, the correct *Chemical Cards* of reagents and products were found. In this step, the reagents and products 3D models are all shown on their cards correspondingly in AR. An unbalanced equation is displayed at the bottom of the screen (see Figure 22). A set of add and remove buttons is needed for changing the numbers of the molecules. The number of the 3D models of the molecules will change in AR when adding or removing the molecules. The validate button is for submitting the answer.

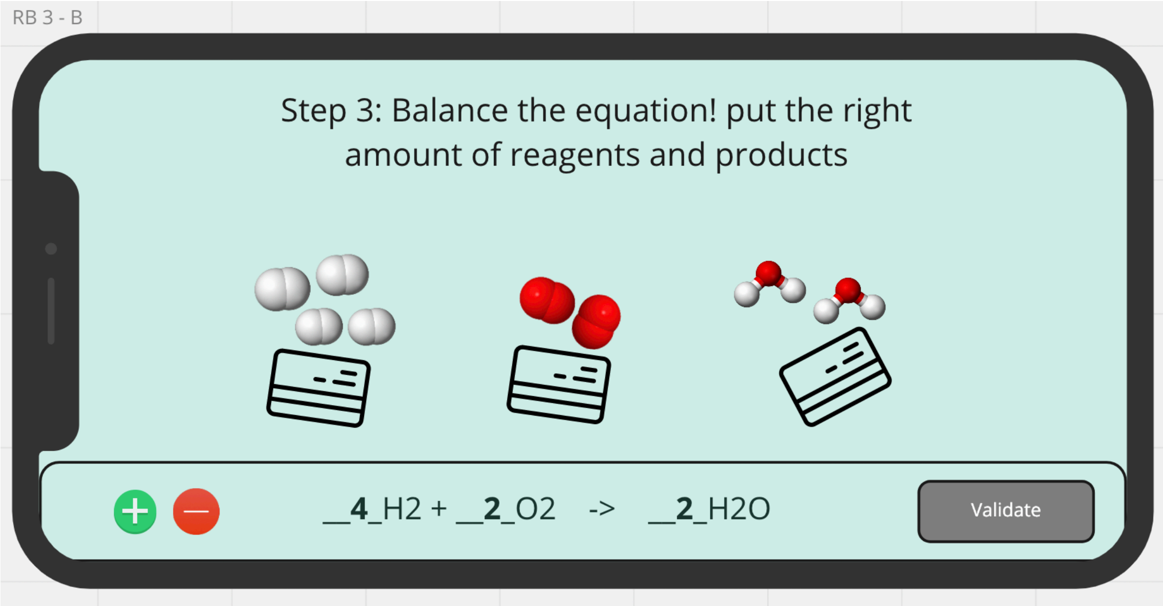

Figure 22. AR Reaction Balancing

Unlike the on-screen reaction balancing, this design makes use of the *Chemical Cards* found by the player during the previous steps. Only a set of on-screen buttons is created for adding and removing molecules in this design instead of one set per reagent/product in the on-screen design (see Figure 23). The on-screen validate button and the equation were moved to the bottom of the screen to provide a larger area for the AR visualisation. The equation symbols between the molecule models were removed in the AR design. No equation symbols are showing between the molecule models in AR since we do not restrict how the player arranges the *Chemical Cards* for this game. The molecule models would be the same but with different scaling and transformation in AR.

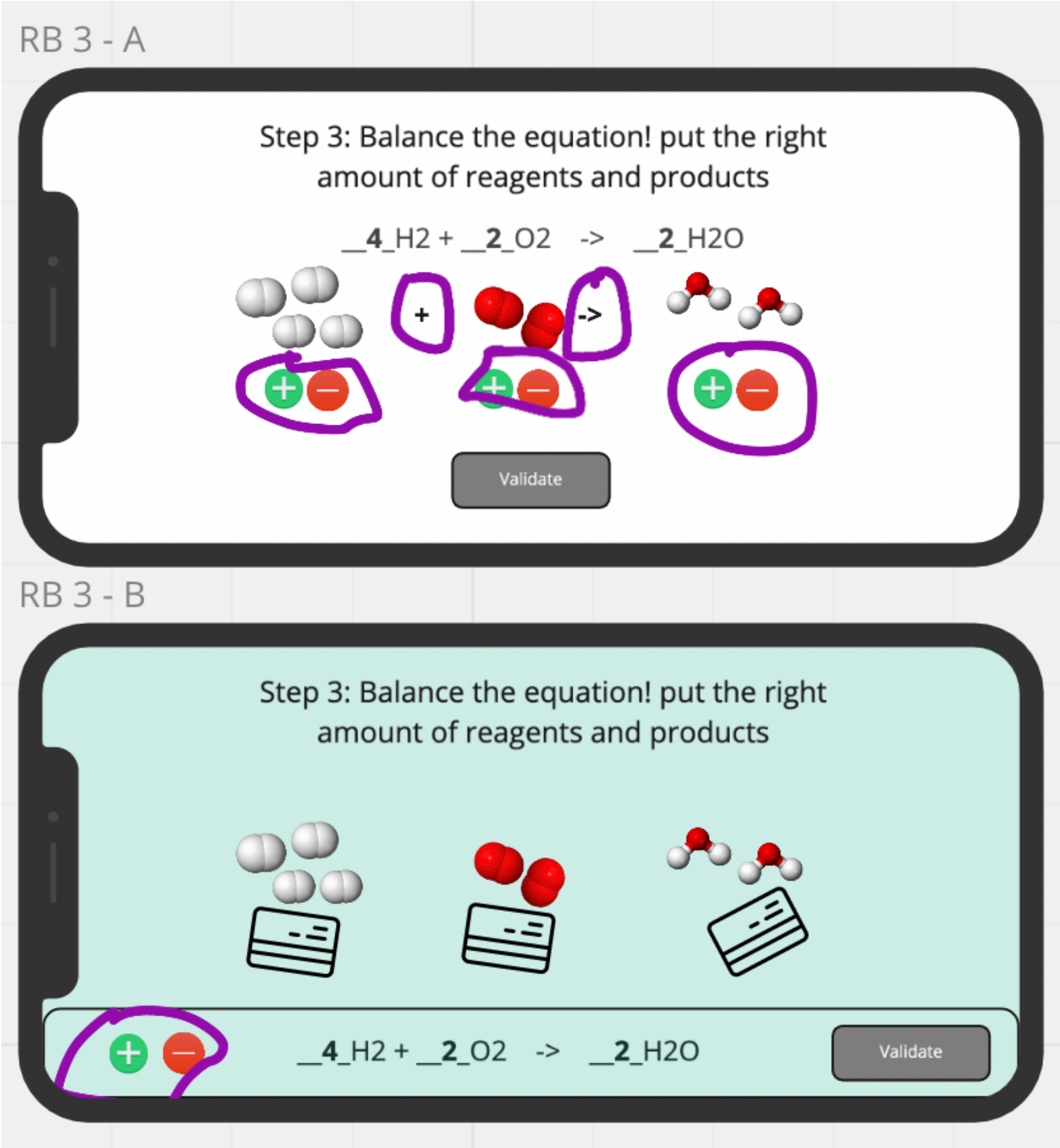

Figure 23. Differences between the on-screen design and AR design

This AR design makes all the steps of this game mode more connected. By changing the number of molecules, the same number of 3D models is shown in AR, it provides a better immersive experience to the player. The downside of this design is that the AR tracking performance of different smartphones might be different. The device might lose tracking when they move their devices. Though the player could still change the numbers on the screen and the molecules in AR, they would not be able to observe the changes until they reinitiate the image tacking. In short, this design requires a stable AR performance to support.

3.6 Differences Between the AR and VR Versions

Though *Futuclass AR* shares many similar concepts and assets with *Reaction Balancing I*, they are designed for different platforms so there are some key changes that have to be mentioned.

Emphasizing real-world interaction with printed cards (*Chemical Cards*). The AR version provides a different user experience by making use of real-world learning material together with 3D molecule models visualized in AR. The player experience might be not as immersive as what VR version has provided. However, by playing the games, the player would experience a learning process of visual memory[24] due to the frequent interactions with the cards. Additionally, the cards themselves exist in the real world even if the AR game is not being run. They are learning materials that might be able to help players' learning progress even after the game. Thus, the design is to make the cards play a more important role.

UI Design. As a mobile AR application, the UI area is limited by the device's screen and it would be different from a virtual world environment provided by VR devices. The AR scenes of the game would operate with a full-screen camera in real-time. Thus, the game information and the controls have to be placed properly such that they are enough clear but not obtrusive. The main AR content (molecule 3D model) has to be more easily seen at the scene. The GUI is designed to be with very few visible UI elements only, unlike *Reaction Balancing I*, which could provide a laboratory-like virtual environment filled with colourful 3D objects. UI elements like buttons and message boxes are created with the semi-transparent grey background. This makes the colourful molecules shown in AR or on-screen look more outstanding comparatively.

Different Playing environments result in different playing styles. Wearing a VR headset to play VR games requires space, and the players are suggested to be monitored by people surrounding for safety concern (see Figure 24). There are physical distances between players. With the immersion effect, players are focusing on the VR world. Direct players' communications are not encouraged in this situation. To share playing contents or play the game with multiple players, players all have to wear headsets and be separated

[24] https://www.sciencedirect.com/topics/medicine-and-dentistry/visual-memory

with safe physical distance. To conclude, it is designed to let the player immerse into the VR world and to isolate the player from the real world as far as possible.

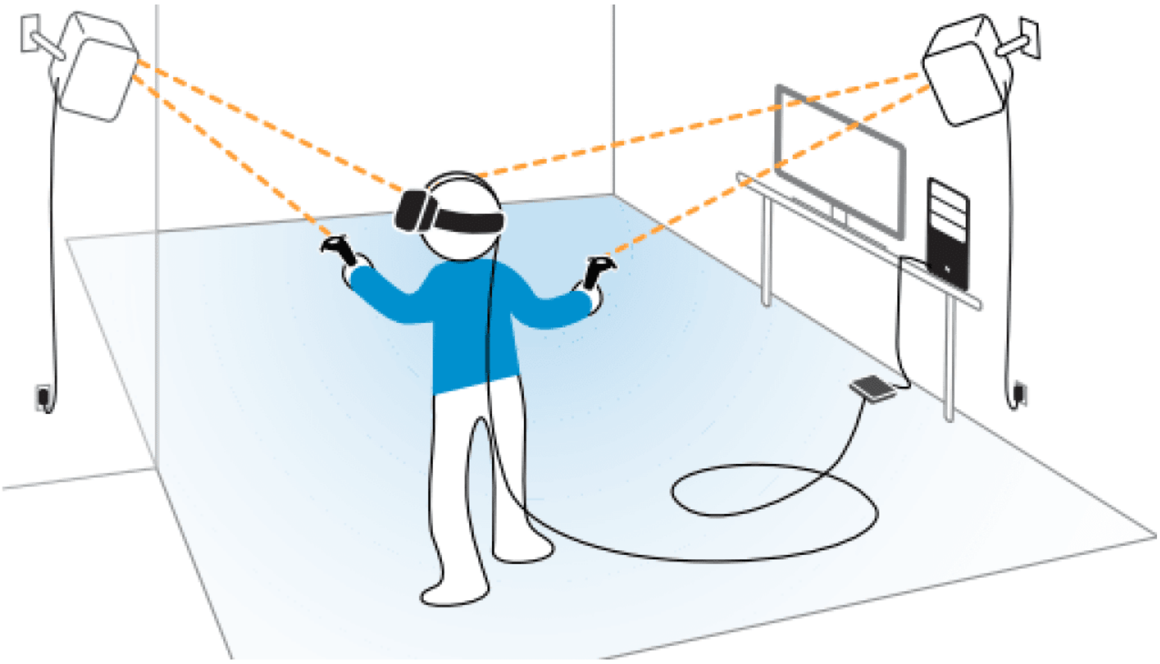

Figure 24. VR Space set-up[25]

Unlike the VR version, the AR version is designed to be playable with only printed cards and smart devices. The immersive experience of the AR version becomes not as important compared with the one in the VR version since it is not the strength of mobile AR. The strengths of the mobile AR platform are the real-world interaction and the flexible environmental requirement compared with VR. It allows students to play together with their friends and families at home or in the classroom. Close contacts, eye contacts and discussions are encouraged in such an environment. Players could communicate with surrounding people naturally and could complete the task in a cooperative way like finding the correct card from numerous cards. Thus, the playing experience would be more like playing a board game rather than playing a mobile game with this design (see Figure 25). The player could use the app with the cards on a table or a floor. The player could share their joy with his/her friends. Teachers could use this AR app as a tool to teach in the lesson and discuss the topics with the students.

[25] Image from the web: https://www.octopusrift.com/setup-your-room-for-vr/

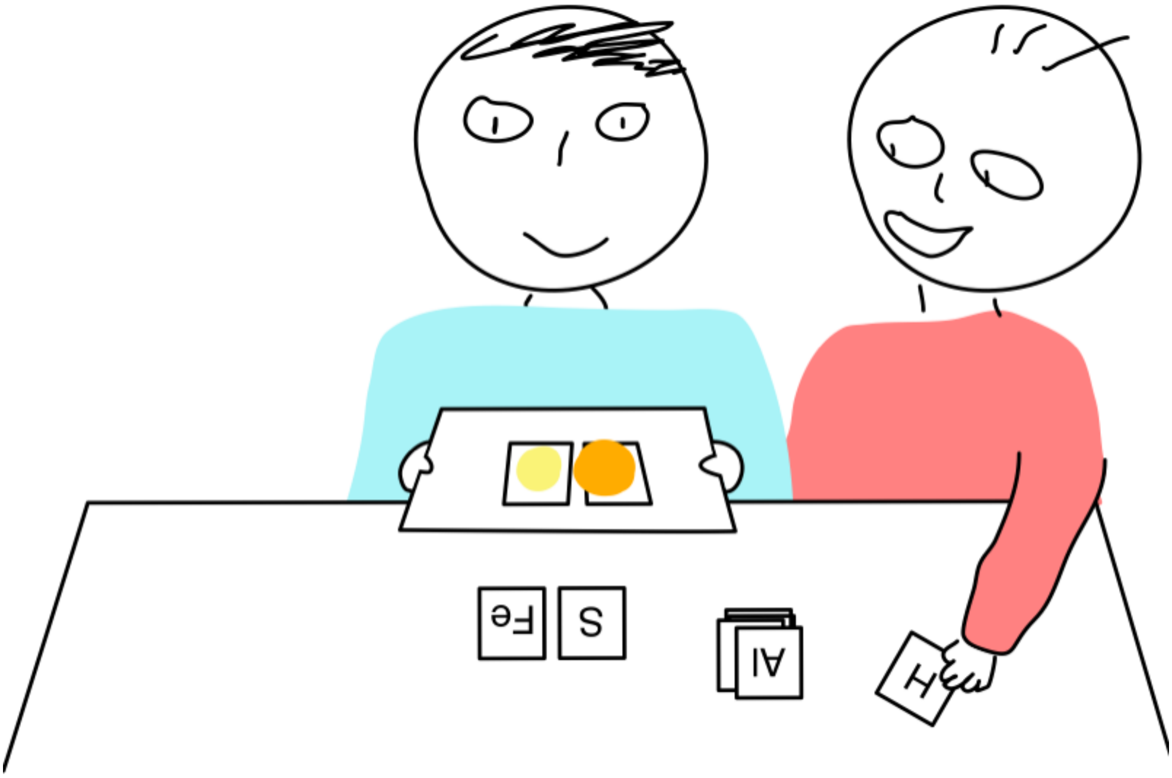

Figure 25. Illustration of playing *Futuclass AR* with a friend

In this chapter, the main concepts and the designs of the *Futuclass AR* were discussed. As mentioned before, the detail of the design has been updating during the whole implementation. To make a workable prototype, there were a lot of considerations and difficulties to overcome during the implementation. They are discussed in the following chapter.

4. The Implementation

This session is about the implementation detail. It is an AR game development. Thus, this part will focus more on the practical issues we faced during the AR development.

4.1 Tools

To fulfil the requirements, the game engine Unity is used for the implementation. In Unity, the game is developed in C# programming language and could be compiled to different platforms like Android and iOS easily. AR Foundation is chosen as the framework for AR development. In this session, we would discuss them and the reasons for choosing them.

4.1.1 Unity

Unity[26] is a cross-platform game engine. As a game engine, it provides many convenient functions and toolkits for developers. It provides a good GUI asset management system (see Figure 26) with a powerful scripting environment with its UnityEngine API. There are other good game engines like Unreal Engine[27] which also provide similar functionality and support AR development. However, Unity is the main tool used to build *Futuclass Hub* by Futuclass so it is better to use the same tech stack. Thus, Unity was chosen.

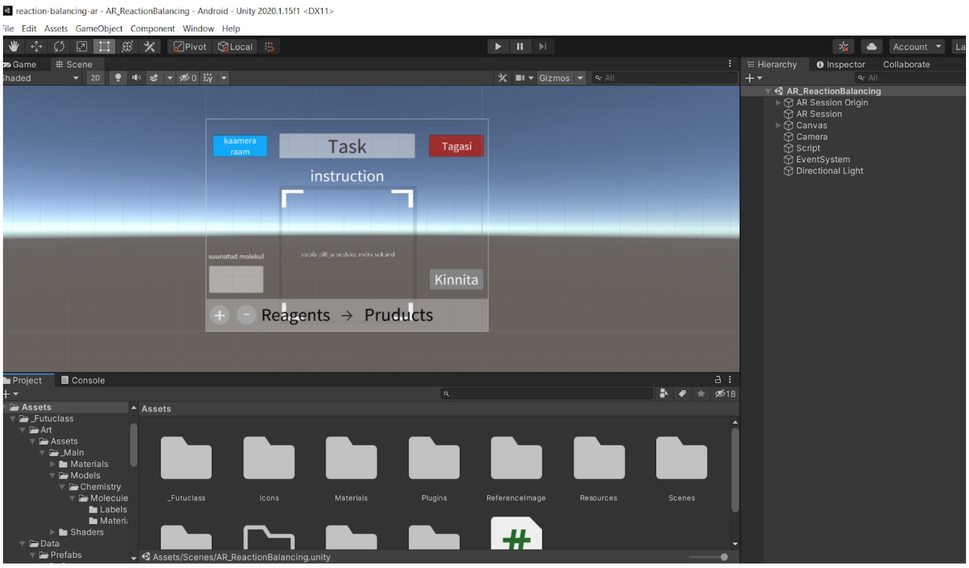

Figure 26. Unity Scene explorer

4.1.2 AR Foundation

AR Foundation the official framework for multi-platform AR development of Unity. It does not provide AR features itself but a high level interface for multi-platform scripting.

[26] https://unity.com/

[27] https://www.unrealengine.com

It supports different features for different platforms. 2D image tracking and raycasting are the main features being used for *Futuclass AR* learning game. By developing with the AR Foundation, a lot of time and efforts is saved hugely since the same development could be built to both target platforms (Android and iOS). Figure 27 shows the differences between AR Foundation development and the ARCore and ARKit developments.

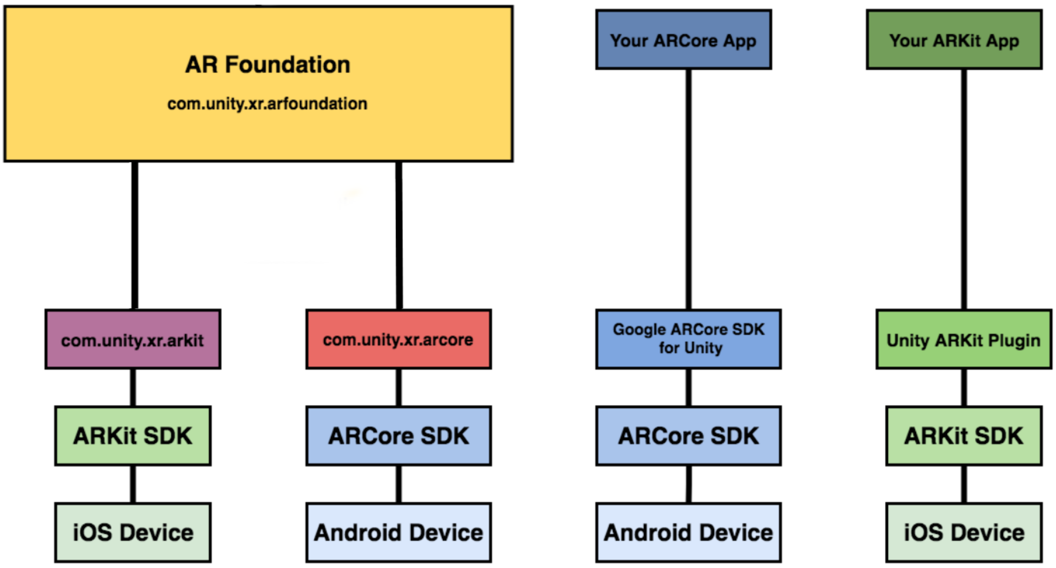

Figure 27. AR Foundation multi-platform development vs single-platform development

There are alternatives like Vuforia and Wikitude. They have powerful AR SDKs for AR development. They are easier to use than AR foundation since they provide good online tools with a good GUI which AR Foundation does not have. For instance, setting up multiple reference image libraries in AR Foundation has to be done with scripting but its Vuforia it could be done with a GUI tool. The only con the alternatives have is that they are third party tools so they have their costs and licenses.

To use AR Foundation because it is the only free option for app publishing and it provides enough AR functionalities. There is a possibility to switch to those third-party tools in the future if more powerful functionality is needed.

4.1.3 Vectormator

AR markers are one of the core elements of *Futuclass AR*. In this thesis, there are 33 markers for the most updated version (0.0.8) in the prototype (see Appendix - Accompany Files). During the implementation, the text sizes and the arrangements of the visual elements on the markers were modified frequently for AR tracking performance tuning. Vector design software is necessary for creating the markers. Thus, Vectormator [28]was used as the design tool.

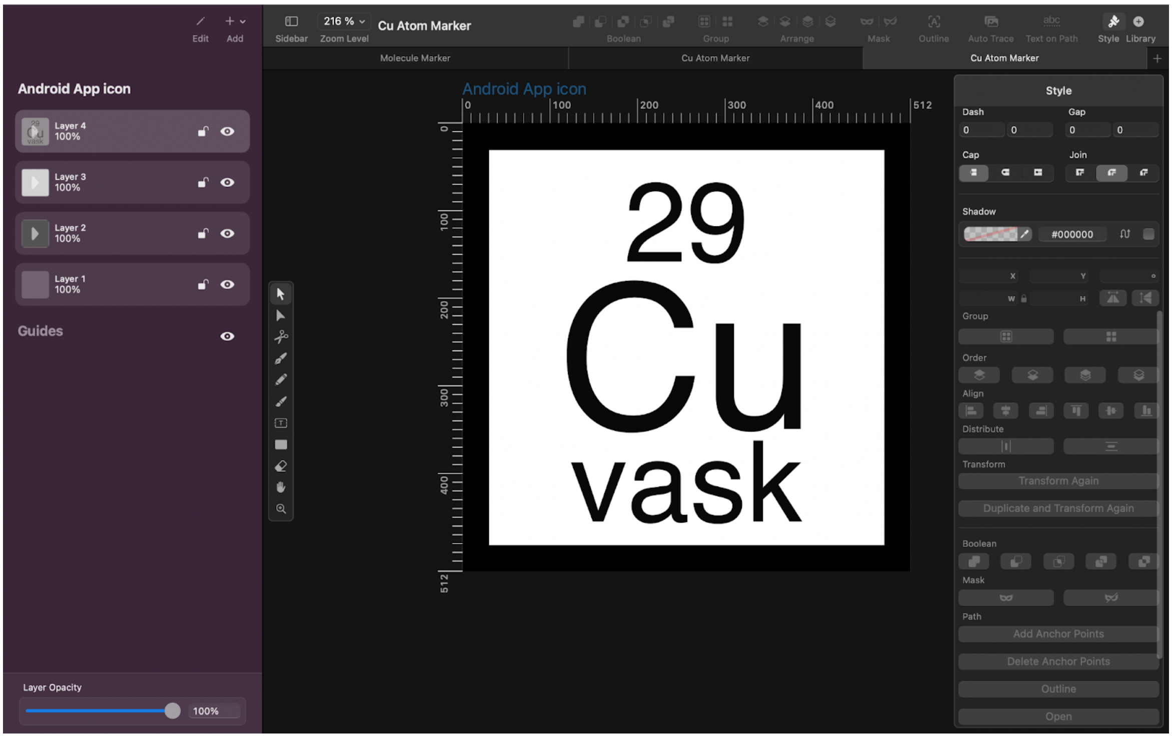

Figure 28. Screenshot of Vectormator[29]

It is completely free on iOS and macOS devices. It is a powerful vector graphic software so the marker designs are scalable and reusable. We could create different base designs for different types of *Chemical Cards* and save them as templates. Then, we could create a new marker with the template with identical styles like text size, text font and formatting (see Figure 28). The design files are saved individually so that they could be modified easily in the future. After completing the design, the markers are exported as separate PNG files in the software and will be imported into Unity.

Now that we have the required tools. In the following, it is going to introduce the actual implementation detail of the 3 game modes of the *Futuclass AR* in Unity.

[28] https://www.vectornator.io/

[29] Screenshot taken from the software Vectormator : https://apps.apple.com/gb/app/vectornator-vector-design/id1219074514

4.2 Game Assets

Game assets[30] are anything that goes into a game. The major assets of *Futuclass AR* include the chemical reactions assets, chemical elements 3D models, and AR reference images (*Chemical Cards*).

4.2.1 Chemical Reactions Assets

In chemical reaction balancing, there are different equations. Each equation consists of reagents and products. Reagents and products are either chemical elements or molecules. Molecules are formed by more than one chemical element. The equations have been grouped into different collections according to their characteristics. They are the *combination equations*, *single replacement equations*, *double replacement equations* and *decomposition equations* shown in Table 3.

Table 3. Equation Collections

Collection Type	Explanation	Example
Composition	- 2 reagents and 1 product in total. - The product is formed by the 2 reagents	$Fe + S \rightarrow FeS$
Single replacement	- 2 reagents and 2 products in total. - 1 reagent is an element and another reagent is a molecule.	$Zn + H_2SO_4 \rightarrow ZnSO_4 + H_2$
Double replacement	- 2 reagents and 2 products in total. - Both reagents are molecules.	$NaBr + CaF_2 \rightarrow NaF + CaBr_2$
Decomposition	- 1 reagent and 2 products in total. - The reagent decomposes into 2 products.	$H_2O_2 \rightarrow H_2 + O_2$

There are 6 equations of each collection. Some fun facts about different kinds of molecules are added to provide more learning content. Table 4 shows some examples of the fun facts. Equation collection, equations, molecules, elements and the related data like

[30] https://unity.com/how-to/beginner/game-development-terms

fun facts are stored as data objects in this application for better management and easy maintenance. The details will be described in Chapter 4.3.

Table 4. Examples of fun facts of chemicals

Chemicals	Explanation
Fe (Iron)	A common metal used in many constructions. Iron is the most abundant element on Earth in terms of mass.
Zn (Zinc)	Active metal used mainly to prevent corrosion or corrosion protection.

It is noted that in this prototype, only the *combination equations* and *single replacement equations* have been created as data assets in Unity. The remaining equations of other collections could be easily created in the near future.

4.2.2 3D models of Atoms and Molecules

The 3D models are those originally from Futuclass. The molecule 3D models are formed by atom 3D models. Atom models are presented as spheres. Different Atom models have different sizes. The 3D model assets are named by their chemical symbol and type like "O Atom" and "H2O Molecule". Figure 29 shows how the "Al2O3 Molecule" molecule model is formed. The elements are set with different materials. The elements' materials have different **colour** , **metallic** and **smoothness** values.

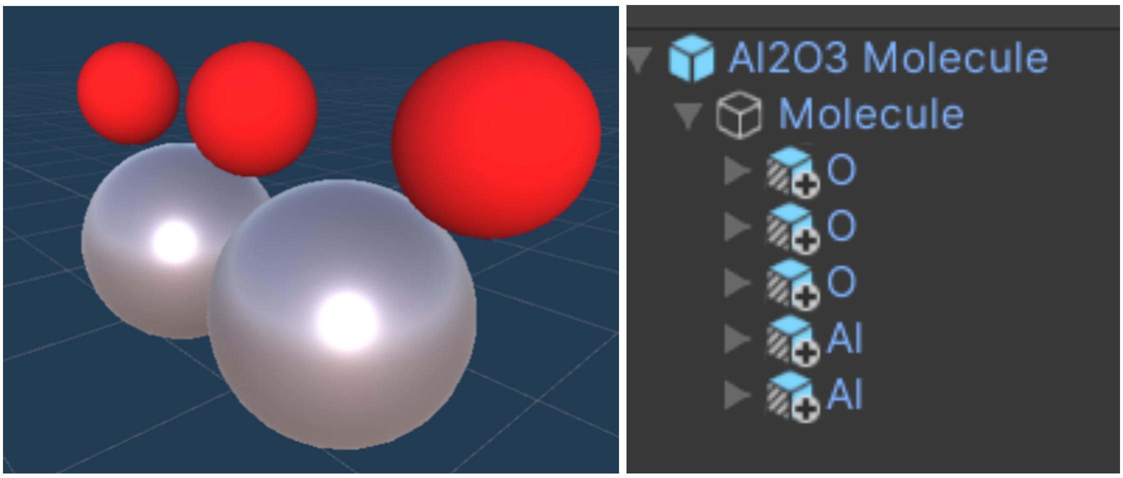

Figure 29. Al_2O_3 molecule model and its components

Small changes had to be made in *Futuclass AR* in the pivot points of some molecule models. This is because some of the pivot points are not at the centre of the molecule models. This made the transformation animation weird and not accurate. There is no automatic method to correct the pivot point in Unity. Resetting the position of some molecules manually according to the transformation offset value became unavoidable (see Figure 30).

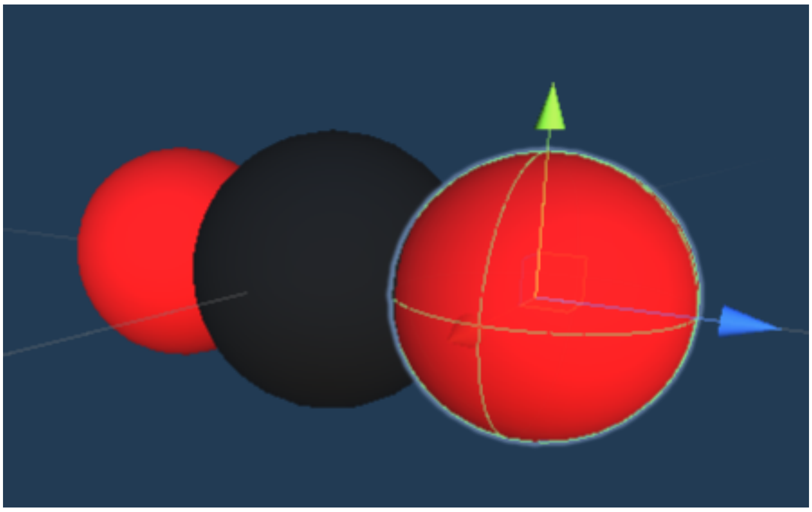

Figure 30. Pivot point altering

The transformation of them is all set to default. All the transformation will be done in runtime They will be described in Chapter 4.6.3.

4.2.3 AR Reference Images

They are the image assets of the AR Markers. Here, the design version 3 is described. The pixel size is 512×512. The physical size is 2.95×2.95 in. The physical size is input into the game engine reference image library for improving tracking quality (see Figure 31). The asset is in PNG format without compression. They are named according to the chemical symbol and the type like "Fe Atom Marker", "FeS Molecule Marker". The naming is important since the name of the asset will be used later by the image tracking program. The images are all being tested with the `arcoreimg` tool provided by Google ARCore. ARCore recommends using images with a score of at least 75. In this application, all the reference images have scores of at least 90 out of 100. There is no available image tracking quality scoring tool for iOS so only the `arcoreimg` tool is being used.

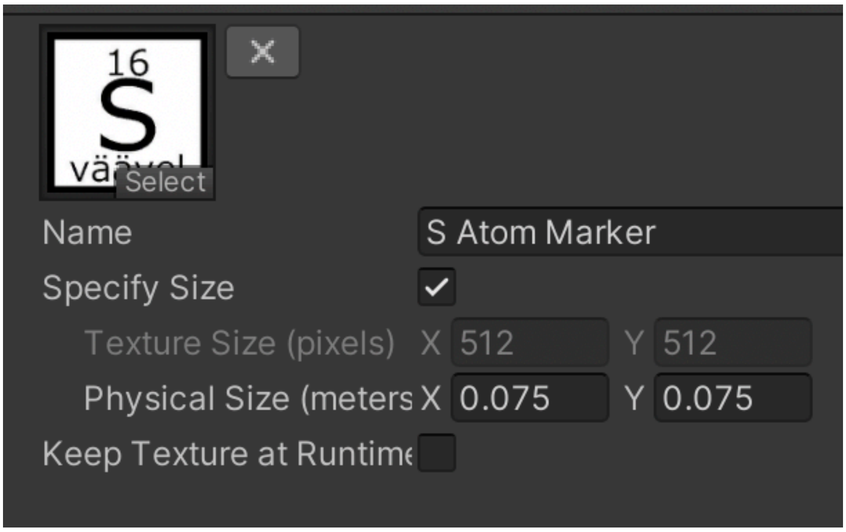

Figure 31. AR reference image properties

All the major assets of *Futuclass AR* have been introduced. For chemical reaction assets, they have to be created as data for the game. In the following, the implementation of these data assets in Unity is discussed.

4.3 Data Objects

Data objects represent a huge amount of game elements in this application. No traditional database or data file types have been used for this implementation. Instead, we make use of the data container of Unity called `ScriptableObject`[31]. Figure 32 is the class diagram showing the relationships between them.

[31] https://docs.unity3d.com/Manual/class-ScriptableObject.html

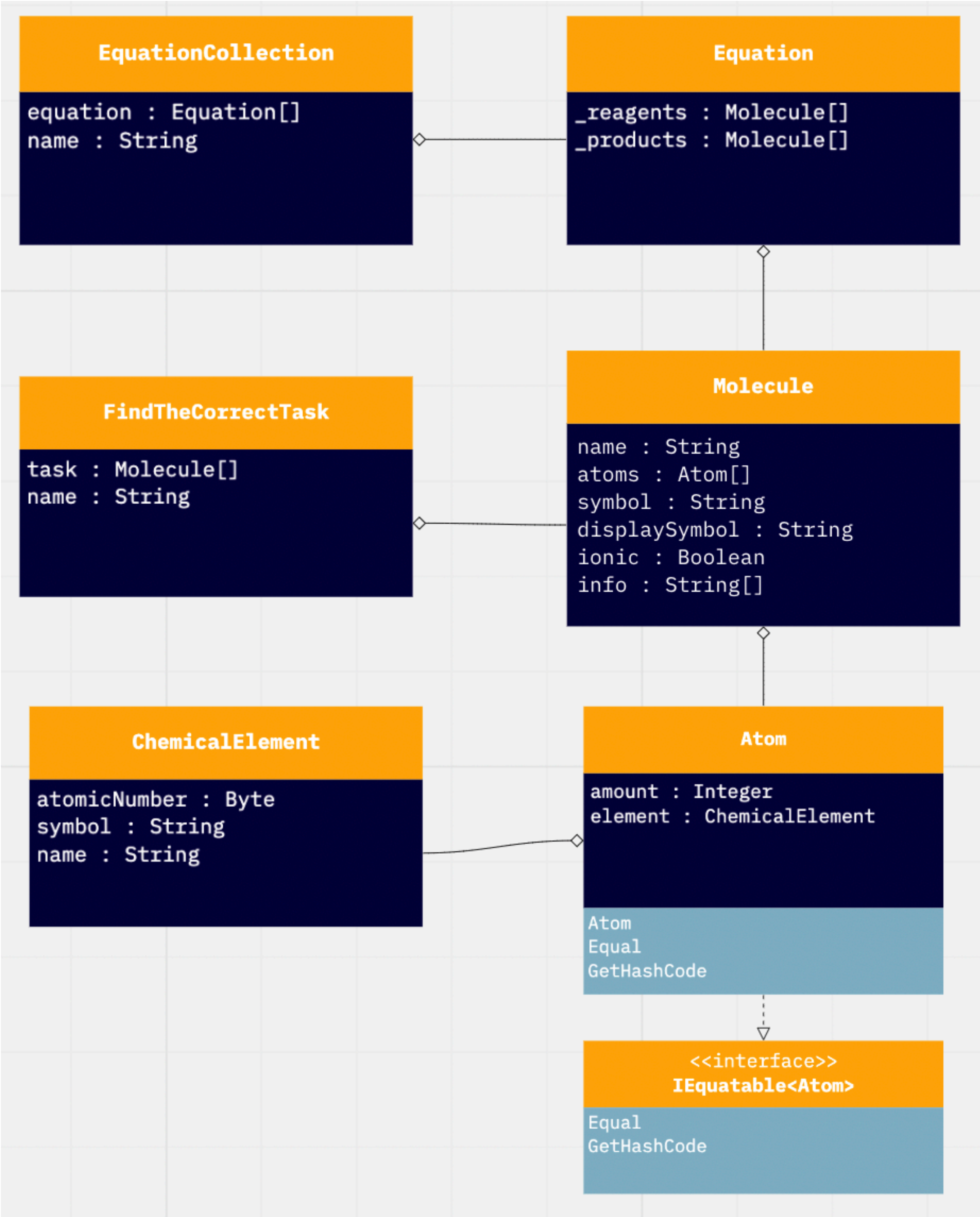

Figure 32. Scriptable objects of Futuclass AR

All of them are implemented as `ScriptableObject` except for the `Atom`. These `ScriptableObject` data containers could be managed easily via the Unity GUI Asset folder. Moreover, the `CreateAssetMenu()` attribute supported by `ScriptableObject` makes the creation of new data assets fast and easy via Unity GUI. It is a data-oriented design[32], which allows the non-tech game designer to add new content to the game via GUI without touching the code.

[32] https://en.wikipedia.org/wiki/Data-oriented_design

4.3.1 Tasks

The task information for the *Find the Correct* game mode includes what molecule to be found with a question. In Unity, the tasks are grouped into a scriptable object called `FindTheCorrectTask` object. The `FindTheCorrectTask` object is aggregated of `Molecule` objects (see Figure 33), which represent the answer to the questions. As for the questions, they are created during runtime making use of the different fields of those `Molecule` objects like the symbol and the name.

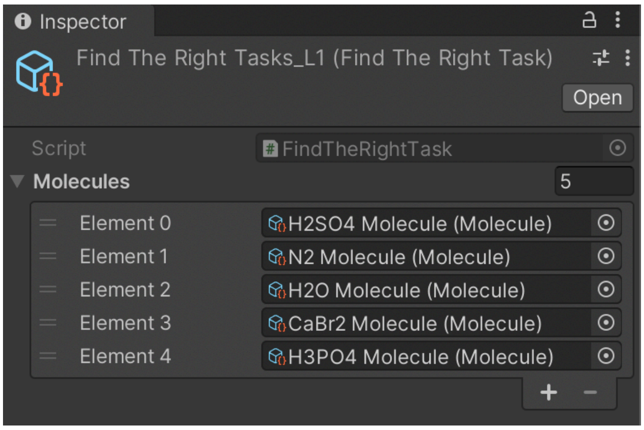

Figure 33. `FindTheCorrectTask`

The chemical equations for the *Reaction Balancing* game mode are grouped into 4 collections. They are represented by the `Equation Collection` objects and `Equation` objects. `Equation Collection` objects is aggregated of `Equation` objects that are serialisable.

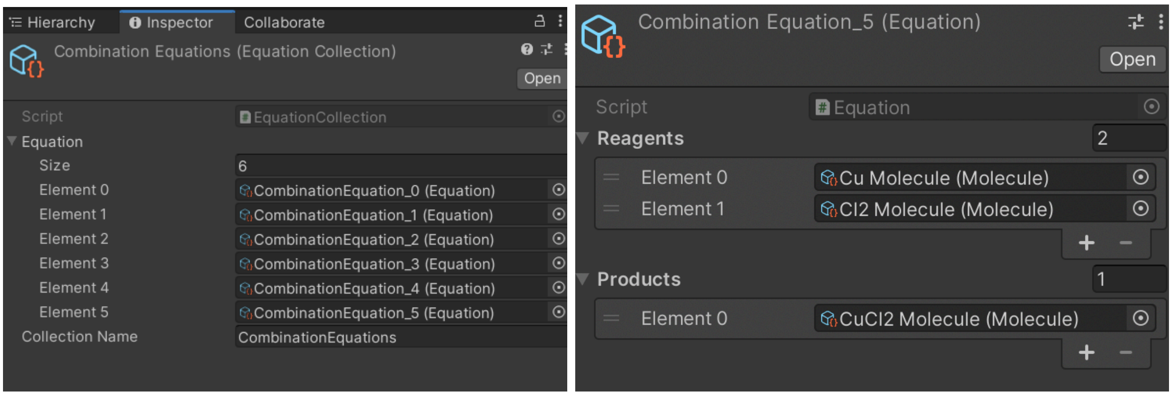

Figure 34. `Equation Collection` (left) and `Equation` (right)

For instance, a *combination equation collection* is a group of 5 chemical equations. We can create the equations individually via Unity GUI as `Equation` objects. Then, the `Equation` objects could be added to the `Equation Collection` object representing the *combination equation collection*. This makes the equation collections to be easily maintainable via the Unity GUI inspector (see Figure 34). The next subchapter is going to explain how these reagents and products are formed.

4.3.2 Molecule, Atom, Chemical Element

In the `Equation` object, the reagents and products are `Molecule` object arrays. `Molecule` objects are aggregated of `Atom` objects, which are implemented as an interface for the `Chemical Element` objects. `Molecule` object contains fields about the molecule like the name and symbol which are used in every game mode in *Futuclass AR*.

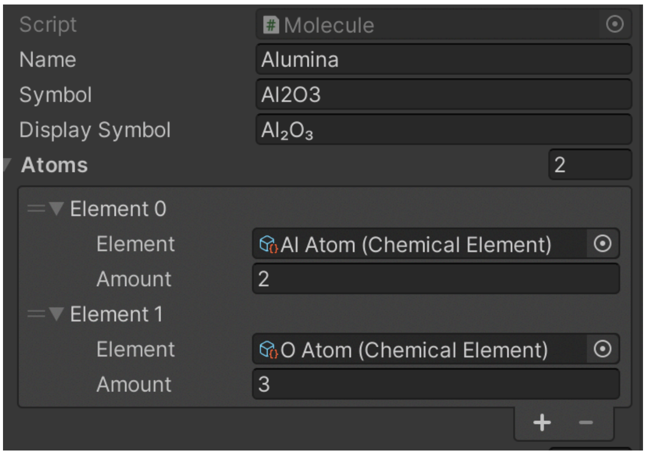

Figure 35. Setting up a molecule in GUI inspector

In Figure 35, the Molecule Al_2O_3 is formed by the Al Atom and the O atom. The amount can be set via the Unity GUI Inspector. In such a way, we do not need to create multiple chemical element instances to represent the atom. This is indeed an implementation of the flyweight pattern [11].

Flyweight pattern is an important data structure since it helps us to avoid data duplication. The data which could be shared would be defined once only as of the intrinsic state. For instance, an oxygen chemical element data asset will be created once only. All the molecules formed with the oxygen will refer to that particular one data asset. As for data unique to each instance, they are the extrinsic state. For instance, we need to create 2 different molecules objects for H_2O and CO_2 separately. Even though only one H_2O molecule object is needed to be set in the entire game since different equation objects could refer to that specific H_2O molecule.

The key concept of data objects have been discussed. In the following, it is to introduce how the player interacts with these data objects to play the game.

4.4 User Input

Futuclass AR is a software that implemented interactive computing[33]. It is software that accepts input from the user when it runs. As for a video game, the user input is events triggered inside the game loop, which runs continuously during gameplay [11]. In the implementation of this prototype, we tried to avoid putting any unnecessary functions to the update() method to reduce the number of executions. For instance, the answer checking logic will only run once right after valid user input is detected. To archive this, the Observer Pattern is being implemented for handling the user input.

Figure 36. Screenshot of getting correct answer in Find the Correct

A separate script is created called `ARObjectSelectionController`. It is of as a `MonoBehavior` class. The `Update()`[34] of the `MonoBehavior` is called one per frame. In the `Update()` function of the controller, it keeps listening to the mobile screen's touch event. If there is any touch, it will check whether the user can touch a molecule model or the card in AR with the result of the hit of raycasting. If the touch hits a valid molecule or its card, the program will broadcast a message in that scene by calling the function `SendMessage` in Unity API. The message will contain the symbol(s)' of the molecule(s).

[33] https://en.wikipedia.org/wiki/Interactive_computing

[34] https://docs.unity3d.com/ScriptReference/MonoBehaviour.Update.html

Multi-touch is supported by setting the above variables as arrays. This controller is used in all different AR scenes of this application since it does not care who receives the message.

4.4.1 Feedback System

After receiving valid user input, the answer checking logic will only run once after a valid user touch input is detected in *Find the Correct* game mode. As for AR reaction balancing mode, it serves as a selector of the molecules instead. After checking the answer, the game will give out feedback including colour flashing and UI message (see Figure 36).

The responsive feedback flashing (see Chapter 3.5.2) was implemented with a special type of function called `Coroutines`[35] in Unity API. This is a good function for creating a sequence of events over time like animation and changing colour. In `Coroutines`, it supports time delay execution with the function `WaitForSeconds`[36]. By calling the `WaitForSeconds` inside the `Coroutines`, the code for changing the colour of the feedback box could be delayed.

4.5 Reaction Balancing

The reaction balancing plays an important role in this game. There are 118 elements in the periodic table. And, there are much more equations that could be added to the game potentially. Thus, it is necessary to ensure the correctness of the algorithm and make the game scalable.

To balance a chemical equation, it is to change the scalar numbers of the reagents and products to make sure that each side of the chemical equation (reagents and products) must represent the same quantity of any particular element. Moreover, a balanced equation must have its coefficients with the smallest whole number only. If any fractional coefficient exists, it is needed to multiply them to make them whole and smallest.

For reaction balancing, we are only interested in the elements but not the molecules. Thus, the comparison is at the `Atom` object level but not at `Molecule` object level. When the reaction balancing game starts, the system will compute 2 lists of `Atom` objects based on the `Molecule` initialized in the reagents and the products. The 2 lists are named as

[35] https://docs.unity3d.com/Manual/Coroutines.html

[36] https://docs.unity3d.com/ScriptReference/WaitForSeconds.html

ReagentAtoms and ProductAtoms. For instance, after loading the Equation object of Equation 1, the ReagentAtoms and ProductAtoms lists would be created. There are 2 Atom objects in the ReagentAtoms list, which are H Atom object and O Atom object. The amount properties of both of them would be 2. As for the ProductAtoms list, there are also 2 Atom objects, which are H and O and the amounts properties both of them are also 2. Obviously, this equation is balanced at the beginning. For an unbalanced equation, when the numbers of the coefficients change, the program changes the amount properties of the Atom accordingly.

$$H_2O_2 \rightarrow H_2 + O_2 \quad (1)$$

For object comparison, Atom objects have been implemented as IEquatable. We compare them with a custom hashing function returned by each Atom object. The hashing function:

```
return ((element != null ? element.GetHashCode() : 0) * 397)
^ amount;
```

397 was generated by the override method. It is a sufficiently large prime number for the good distribution of hash codes. For example, the H Atom object with amount 2 in the ReagentAtoms list would return a hash code same as that returned from the H Atom object in the ProductAtoms list.

For checking if the equation is balanced, the below boolean function is used.

```
bool IsBalanced = !ReagentAtoms.Except(ProductAtoms).Any()
&& !IsDivisible;
```

The IsBalanced function checks the following conditions:

1. Total Numbers of atoms in both the reagent side and product side are equal
2. Kinds of the atoms in the reagent side and product side are the same
3. The coefficients are smallest-possible whole-number

For checking coefficients, it is enough to check if they are divisible by 2, 3 and 5 by a boolean function IsDivisible.

The above covers some key concepts about how the user input is handled and the answer checking in *Futuclass AR* in runtime. They are important since they represent the rules of the game. The necessary game data and the game logic have been discussed. The next chapter is going to introduce the implementation of the core visual contents, which are the AR contents of *Futuclass AR*.

4.6 AR Sessions

The AR sessions play the most important role. In *Futuclass AR,* all the 3 game modes contain `AR Scene`. `AR Scene` is an environment including all the things we need for the game at one time with AR Foundation features enabled. Though AR Foundation provides good documentation, the basic methods and templates provided are not enough for our design. Additional scripting and customised AR tracking logic are discussed.

4.6.1 Image tracking

In an `AR Scene`, there are 3 necessary components which are the `ARSession`, `ARSessionOrigin` and the `ARCamera`. The `ARSession` controls the AR lifecycle, including enabling or disabling the AR functionality. `ARSessionOrigin` manages all the transformations of the trackables and transforms them into world space in Unity. `ARCamera` manages the camera device properties.

As for 2D image tracking, the `ARTrackedImageManager` is used. It will create `GameObjects`[37] for each tracked image. It needs a `ReferenceImageLibrary`, which is a collection of images to be tracked in the real-world environment. The `trackedImagePrefab` is a field for attaching a component to the `TrackedImage` object. For instance, by adding a cube 3D model as the `trackedImagePrefab`, it can be spawned on the tracked image in AR. However, in this case, the cube will spawn on all the detected images in the library.

[37] https://docs.unity3d.com/560/Documentation/Manual/class-GameObject.html

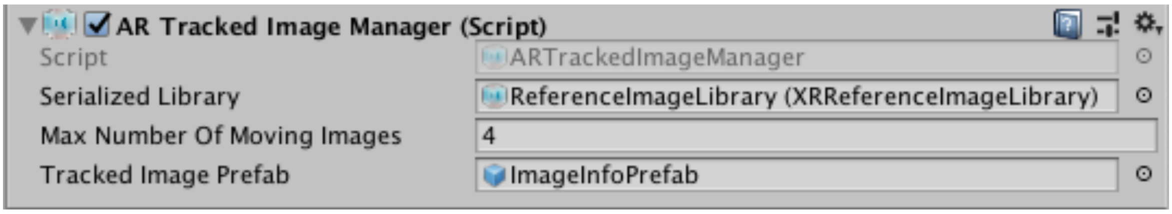

Figure 37. Default setup for AR Tracked Image Manager[38]

The main event of the `ARTrackedImageManager` is the `trackedImagesChanged` event. The event has 3 event arguments, which are `added`, `updated` and `removed`. The `added` event triggers when the tracked image is detected the first time within the current `AR Session`. The `updated` event triggers when tracking information of the image detected changed like position changed or losing track. For `removed`, it runs only if the `AR Session` is going to be disabled. By subscribing to these events, we can write functions to control the transformation, visibility and rotation of the AR Objects.

4.6.2 Multiple Prefabs at Runtime

The default script `ARTrackedImageManager` does not support multiple fields for `trackedImagePrefab` (see Figure 37). A custom script has been implemented. In Unity, If the tracked image prefab field is set to null, the `ARTrackedImageManager` will create a `GameObject` with the name of the tracked image detected. The tracked image name is used as an identifier in this implementation so the names of the images have to be descriptive to let the system know what AR content to show. For instance, the reference image of the H_2O molecule will be "H2O Molecule Marker".

By making the name of the images and corresponding prefabs to be aligned, we could create a dictionary at the `Start()` of the script, where the name of the image is the key and the prefab object is the value. Then, we can instantiate all the prefabs (Molecule 3D models) needed for the AR scene and assign them to the dictionary. All the instantiated prefabs are set to be inactive immediately after creation. This allows the AR scene to preload all the necessary model assets. It is a good practice for performance optimization. It is an implementation of the **object pool** pattern[11]. By instantiating all the necessary

[38] Image taken from the web: https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@4.1/manual/tracked-image-manager.html

prefabs at the beginning, it ensures the device will assign enough memory pool to the `AR scene`. The created objects will only be destroyed when the scene is being closed.

After instantiating the prefabs, we can write functions like setting the right prefab to be active, performing transformations and disabling the prefabs. In our case, the correct molecule model will spawn correctly on the corresponding *Chemical Cards*.

4.6.3 AR Objects Hierarchy

In this implementation, The 3D model objects have been prepared as prefabs. The three components are the parent object, the 3D molecule model prefab and the transparent overlay card prefab. The parent object is instantiated in runtime and the name of it is set to be the same as the corresponding tracked image.

As for 3D molecule model prefabs, they were implemented by Futuclass. The 3D model `Molecule` prefabs have a `Molecule` game object, which includes corresponding 3D molecule atom prefabs. For example, the `CuCl2 Molecule` structure is shown in Figure 38.

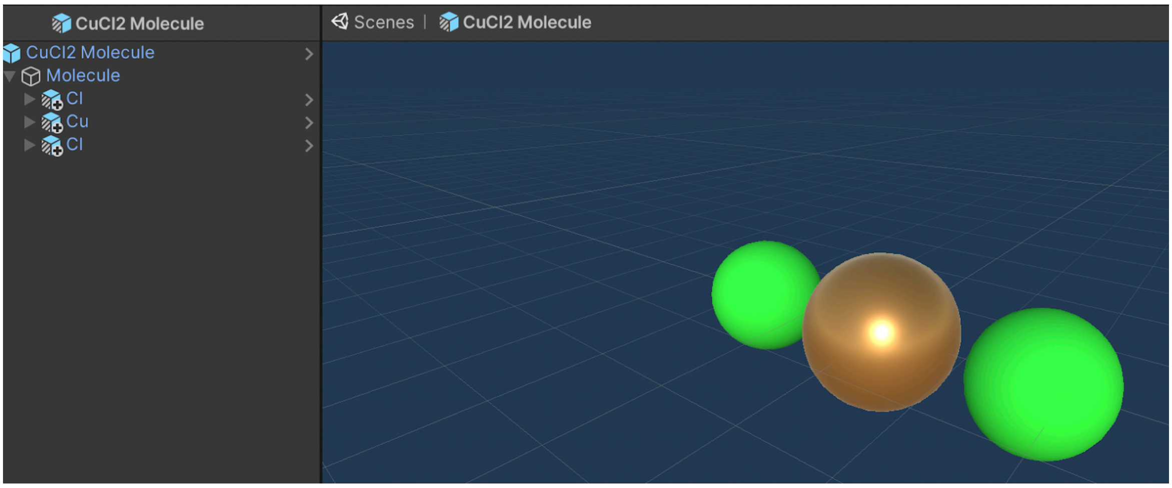

Figure 38. CuCl2 3D model prefab structure

The `Overlay` prefab is set with a transparent material. Its length and width are the same as those of the tracked image. The height is set to be very short. This transparent prefab is added as an additional raycasting object besides the molecule object for user input.

The transformation of the `Molecule Marker` object will be updated by the AR tracking script described earlier. Thus, the transformations of the 3D molecule model and the overlay card are set to parent the transformation of the marker object so they could share the same positional changes. The transformation stack is shown in Figure 39.

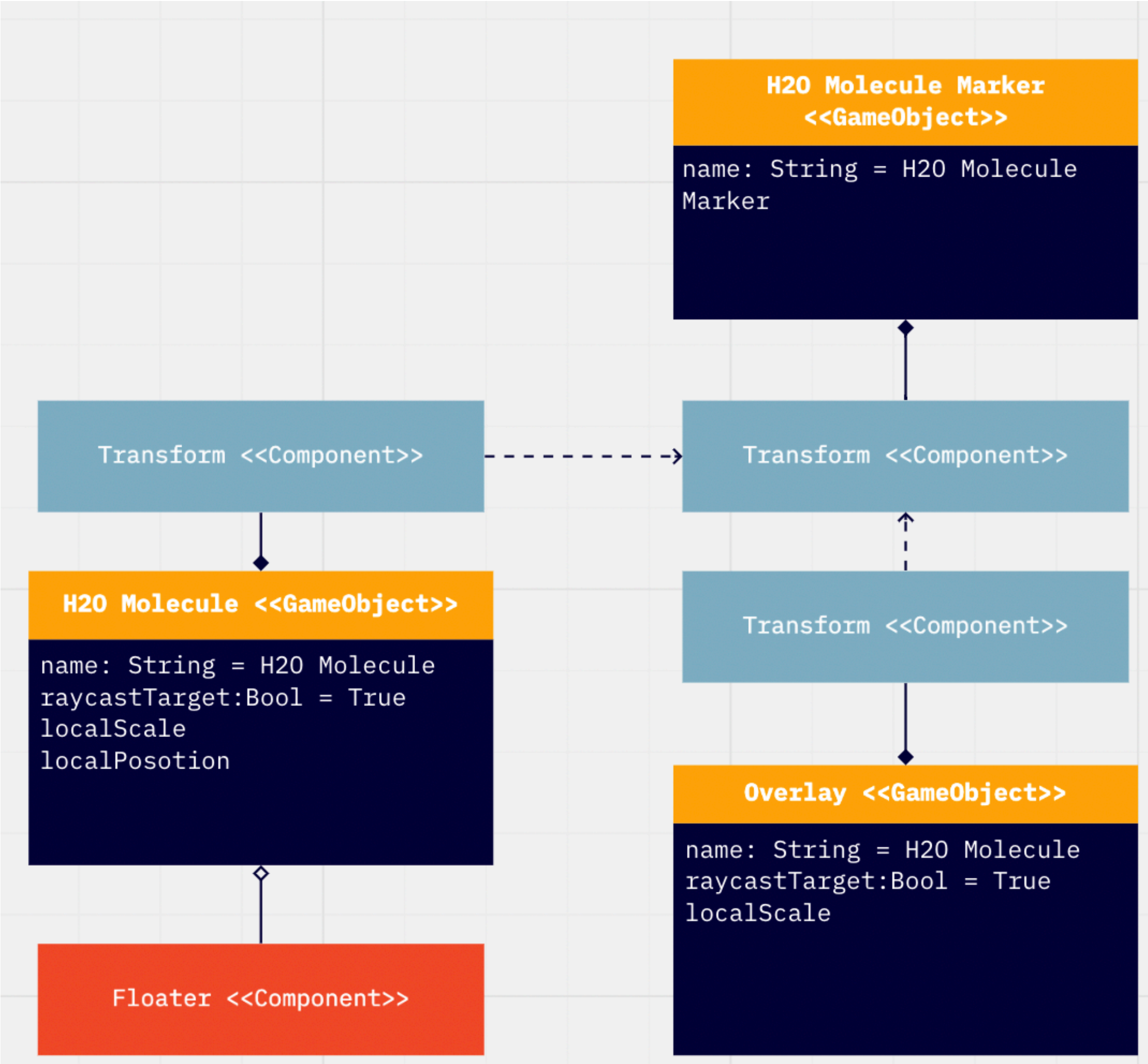

Figure 39. Transformation stack of the AR objects

We apply the scaling to the molecule. This enables us to adjust the universal scaling of the molecules easily. To make the floating animation more appealing, a floater script is created. The floater script includes logic to change the local transformation and rotation. The rotation is in self-space spinning 15 degrees per second. The floating is done with a sine function as below.

```
tempPos.y += Mathf.Sin(Time.deltaTime * Mathf.PI * frequency) * amplitude;
transform.localPosition = tempPos;
```

For design 2 of the reaction balancing, the numbers of the 3D molecule models have to be updated according to user input in step 3 (Chapter 3.5.3). To achieve this, an

`ARSpawnMultipleController` is implemented. Figure 40 shows the changes of 3D molecule models in AR after user input.

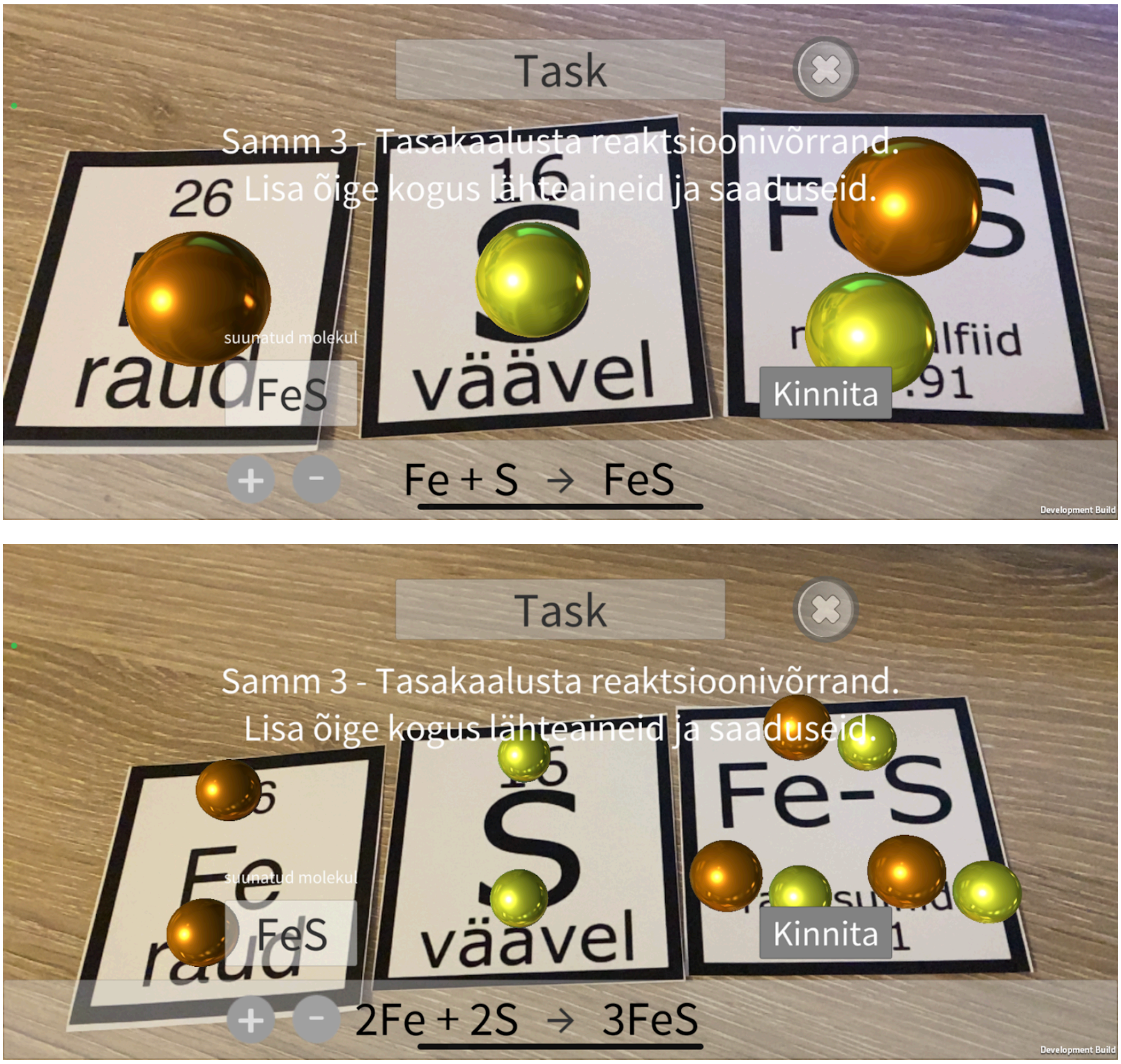

Figure 40. The molecule 3D models before adding molecules (upper) and after adding molecules (lower)

This subchapter has demonstrated how the AR contents were implemented in *Futuclass AR*. With AR Foundation, the tracking logic was compiled into ARCore codes for Android build and ARKit for iOS build accordingly. However, the performances between different platforms and different devices vary. Optimisation regarding the tracking issue is discussed in the following.

4.6.4 Improving Image Tracking on Android devices

During development, it was found that the performance on Android is not as good as iOS with an identical setup. Though the images are all having high scores in the `arcoreimg` tool, the time needed to initialise tracking on Android is generally slower than that on iOS. The reason behind this might be due to software or hardware. This issue affected the gameplay especially when the device needs to initialize tracking again after losing track. Thus, it was necessary to improve it by modifying the tracking script.

Table 5.The different tracking states

State	Description
Tracking	The image is being tracked.
Limited	The image is being tracked but not well or it was tracked but not it could not be tracked anymore.
None	The image has never been tracked before. (initial state)

In the script of `ARTrackedImageManager`, the `ARTrackedImage` provides additional information about the `TrackingState`[39] in Table 5. The original universal tracking script for both platforms (ARCore and ARKit) is to display the 3D model when the state is `Tracking`. When the state is `Limited` or `None`, the 3D model will not be displayed. However, the `Limited` state is usually triggered frequently in ARCore. This may be due to the comparatively poor tracking power of ARCore. For instance, when the player covers the tracked *Chemical Cards* with his hand and then moves away, the spawned molecule 3D model will disappear since the tracking state becomes `Limited`. It takes a couple of seconds for the device to detect the image again. To reduce the frequency of losing track, the AR object is kept visible at most 3 seconds after changing to `Limited` state. If the tracking state becomes Tracked again during the 3 seconds buffer, the AR object will remain visible instead. This could be done with the `Coroutines`. The ARCore will still try to track the targeting image with limited data and update the transformation of the AR object during the period.

Since the core component of *Futuclass AR* is the mobile AR development, there were a lot of testings being done during the thesis for improvement especially for the AR

[39] https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@4.1/manual/tracked-image-manager.html#tracking-state

performance. Indeed, software development is about trial and error[40]. In the next chapter, internal testing and usability testing are discussed.

[40] https://en.wikipedia.org/wiki/Trial_and_error

5. The Testing

Dynamic game contents should be accurate, clear and fun. The interaction methods between the player and the AR contents should be straightforward and easy to adapt. The game application itself should be stable and solid. During the thesis, internal testings were done iteratively. Besides, **usability testing** is essential to the development of *Futuclass AR*. Usability testing allows us to identify problems, uncover problems and learn about our target user [12]. This chapter discusses the internal testing and two usability testing sessions. The differences between different versions will be mentioned since different versions were tested during the development.

5.1 Internal testing

During development, there was iterative internal testing done by me, Kristine Tamm and Arnold Rein Tatunts of Futuclass. Kristine is the supervisor of this project and Arnold is the educational content specialist who helps creating learning content and gives comments regarding educational values. During development, the non-AR elements like game logic and UI rendering could be tested via the game view and the console in Unity (see Figure 26). However, it is not possible to test the code for AR development using AR Foundation namespace on a personal computer. ARCore and ARKit are not supported with the simulator of Android and iOS. Thus, the AR performance is tested with mobile devices.

During the development, tasks, issues, updates and comments are added into a hit list sharing with Kristen and Arnold. In the hit list, the tasks are categorised into 3 priorities from 1 to 3, where priority 1 means the greatest importance. The testings were very frequent since the development process was applying agile software development[41]. There were no hard deadlines for the tasks created during internal testing. However, a stable version without any priority 1 issues had to be ready before each usability testing session. The tasks and the priorities were created after discussions with Kristen and Arnold. The review meetings were held on-site in the office in Delta Centre[42] or via video call.

[41] https://en.wikipedia.org/wiki/Agile_software_development#Agile_software_development_values

[42] https://delta.ut.ee/en/

5.1.1 Image Similarity

Creating good reference images is only one of the factors for good AR tracking performance. In creating marker-based AR applications, not only the markers have to have enough features but the distributions of the features of those marker images have to have enough level of uniqueness. For example, the *Chemical Cards* of $ZnSO_4$ and $MgSO_4$ have got 5-star ratings of `Augmentable` by the Vuforia development portal as shown in Figure 41.

	Target Name	Type	Rating ⓘ	Status	Date Modified
☐	MgSO4	Single Image	★★★★★	Active	Apr 18, 2021 11:41
☐	ZnSO4	Single Image	★★★★★	Active	Apr 18, 2021 11:34

Figure 41. Images rating by Vuforia

In the `arcoreimg` tool, both of them are having scores higher than 75. It is confident that each of them is a good reference image for 2D image tracking.

However, when we put both images into the reference image library. Tracking issues occurred where the molecule models of both of them spawned incorrectly. For instance, sometimes the $ZnSO_4$ was spawned on the $MgSO_4$ card, sometimes they both spawned on either one of the cards as shown in Figure 42.

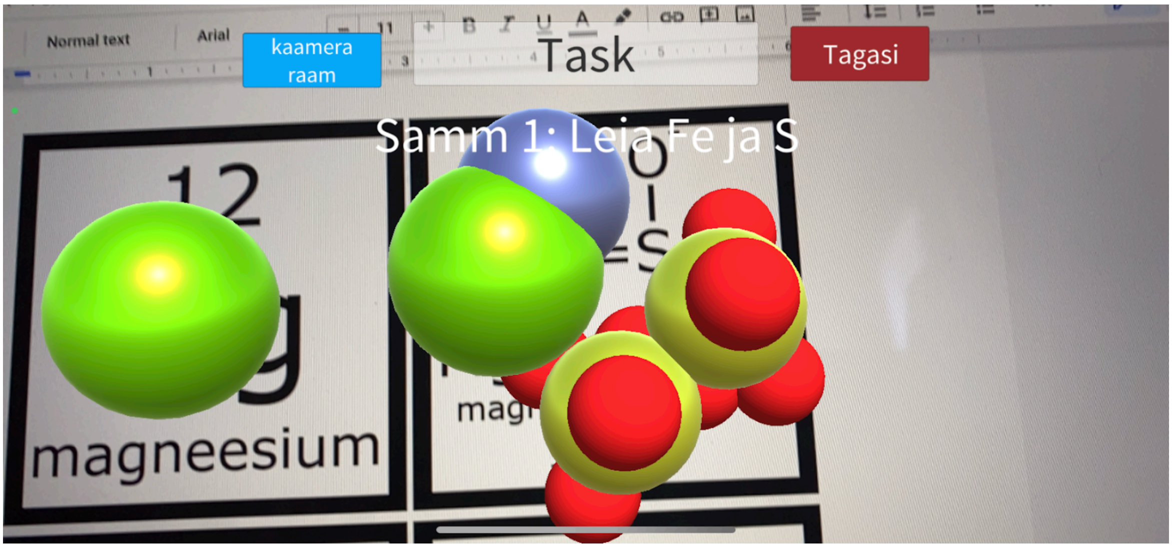

Figure 42. Multiple molecule 3D models on one marker wrongly

This is the curse of image similarity. Given the 2 images, the features of them are having similar distributions to each other as shown in Figure 43. It is a huge challenge for this implementation since the educational contents of the reference images are mainly designed with either name of molecules, chemical formulae or atomic numbers. Elements with no educational value are not a good design fit into *Futuclass AR*.

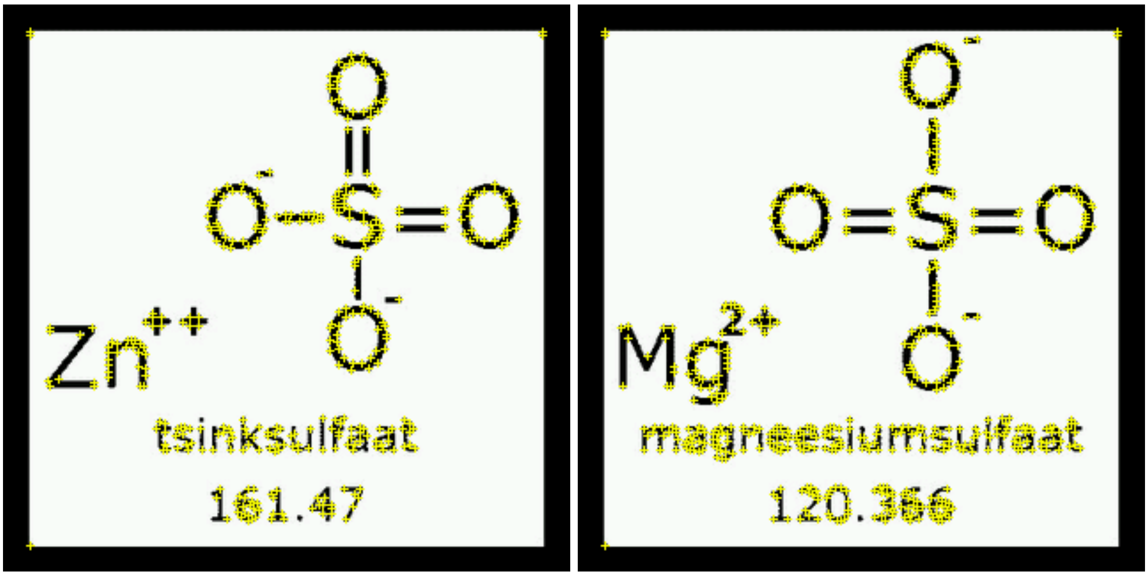

Figure 43. Similar feature distributions on the both images

To update the reference images of both cards, it is to make the distributions of their features to become more different to each other. It could be done by changing the sizes of the elements inside the image and changing the arrangements of them. The fixed reference images are shown in Figure 44.

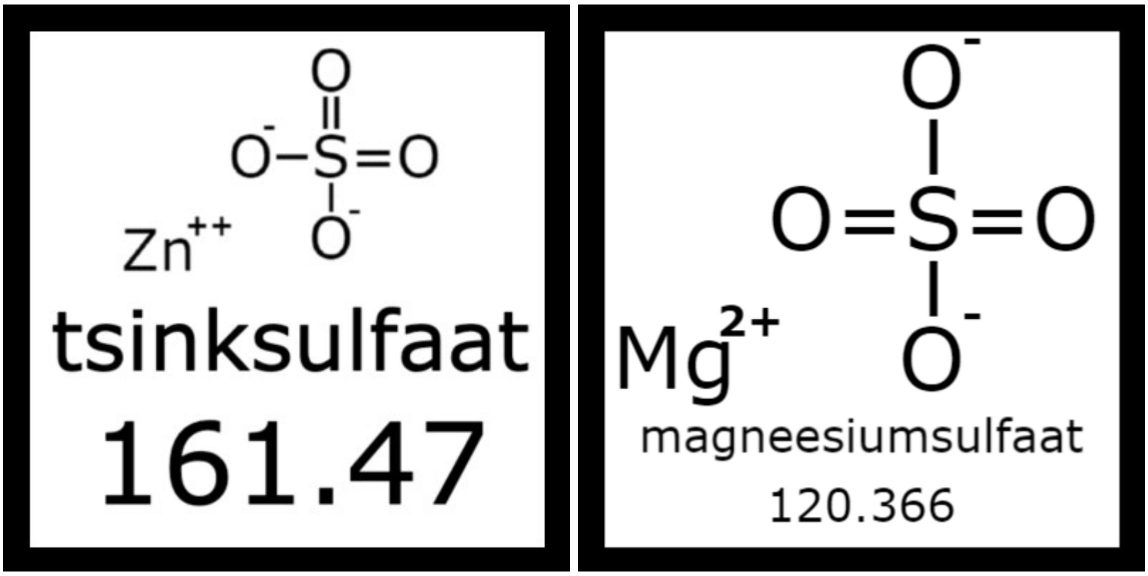

Figure 44. Reference images with different sizes of elements

It is noted that there are no tools or documentation for image features comparison provided from ARKit, ARCore, AR Foundation or Vuforia. Currently, this image similarity issue could be avoided only by observation and practical testing. The good practice is not to add similar images into the same reference image library. Thus, we have learnt that it is better to not include too many tracking image assets for one game mode. For different game modes, different reference image libraries for each of the game modes can be created and the library switched in runtime dynamically.

5.1.2 Version Control

The app version number increases when there are major changes. The repository is in a private repository due to business reason (see Appendix - Source Code). The major versions and the update descriptions are in Appendix - Version Control.

5.2 Usability Testing #1

The main goal of the 1st usability testing was to test the *Find the Correct* game mode and the image tracking performance on different devices. For internal testing, we have 2 testing machines for iOS and 2 for Android. We have learnt that the image tracking performance in ARKit and ARCore are quite different. The ARKit usually provides a better tracking performance than ARCore with our own devices. Thus, testing the app on a wider range of devices especially for Android devices is necessary. In this testing, 7 testers with Android devices and 4 testers with iOS devices had participated. The game *Find the Correct* was chosen for this testing since it has simpler mechanics than the *Reaction Balancing* game. A blog of this testing is in Accompany Files, which is summarized by Futuclass.

5.2.1 Virtual Class Testing

The usability testing 1 was held on the 1st of April. It was a virtual class session with 10 pupils of 7th grade and their teacher via Zoom[43] (see Figure 45). Testing in person was not possible due to the Covid-19 pandemic. It was the first usability testing with a wide range of mobile devices. Thus, there were a lot of difficulties and issues before and during the testing.

[43] https://zoom.us/

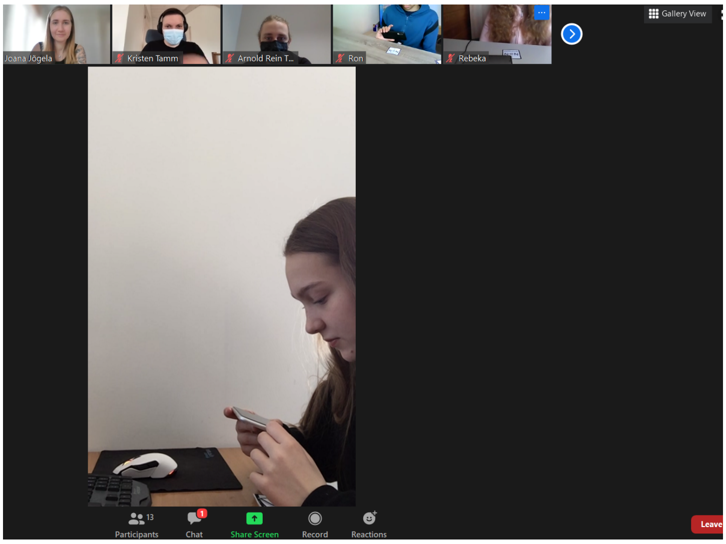

Figure 45. virtual class testing session via Zoom

To deliver the app to the testers, the OSs of the devices of the testers were collected. For Android users, their google accounts were collected. Their accounts were added to the internal testing group in the Play Console[44] (see Figure 46). After that, the download link was sent to them via mail for them to download the testing app on Google Play.

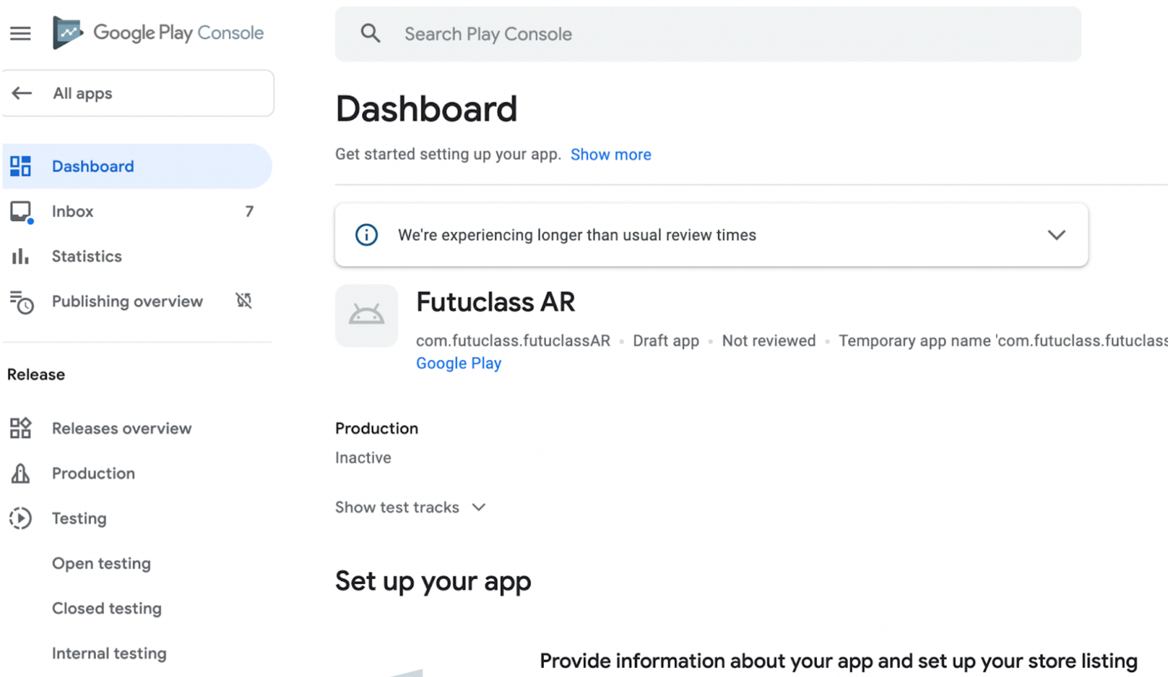

Figure 46. Google Play Console

44 https://play.google.com/console/about/

As for iOS users, their Apple IDs were collected, then their accounts were added into the internal testing group in TestFlight[45] (see Figure 47). After that, The TestFlight will automatically send the testing invitation to them.

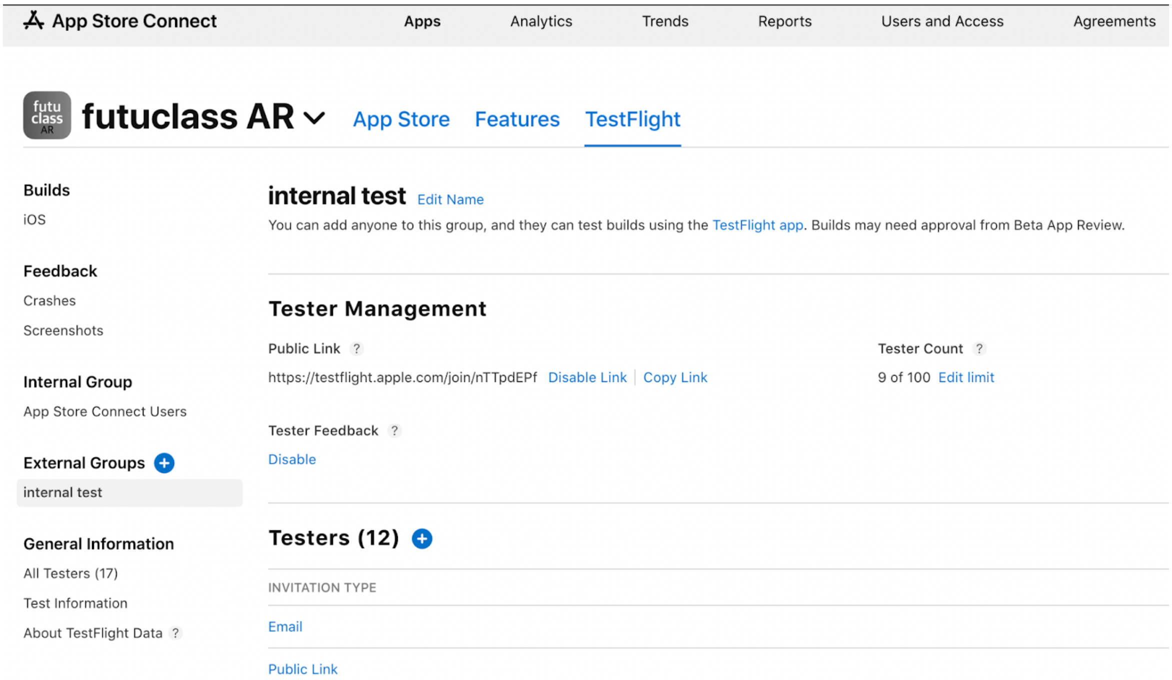

Figure 47. TestFlight on Apple App Store Connect

The app and the *Chemical Cards* files were delivered to the teacher 1 week before the lesson. Then, the teacher tried the app herself and did point out a few chemistry-related mistakes on our AR markers. After fixing those, the teacher had prepared the class herself to make use of the app and the markers. The app and the *Chemical Cards* were finally sent to the pupils 1 day before the lesson. The testing version 0.0.7 was the app for this testing (see Appendix - Version Control).

The lesson was held by the teacher, and the pupils were instructed to play the Find the Correct game during the lesson. Arnold and Kristen and I were the observers of this lesson. After pupils completed some tasks in the app, the teacher asked some questions about the fun facts of the molecules they learnt when playing the game. At the end of the lesson, we asked some questions about the technical issues the pupils faced like not visualising the molecules with *Chemical Cards*.

45 https://developer.apple.com/testflight/

5.2.2 Results

The testing data were collected with 2 types. **qualitative data**[46]. For *Find the Correct* game mode, it was about the observation and the verbal feedback from the pupils during the class session. As for the AR image tracking performance issue, we asked the testers about the difficulties they faced during the game and how the issue/bug happened.

iOS vs Android

The AR image tracking was the main concern area for us since we already knew the performance differences between ARKit and ARCore. Optimisation in the tracking script for ARCore had been done before this testing version. Still, there were 2 out of 7 pupils who could not play the game because of serious tracking issues. The AR contents were messed up on their devices. For instance, the device could only track 1 *Chemical Card*, even the others were inside the camera's view. It was thought that these might be hardware compatibility issues. The device catalogue[47] was checked and the models of the pupils' devices were in the supporting catalogue. The builds were delivered via Google Play so the application package was built and verified by Google in the cloud. Their cases were still in the study. The issue could not be reproduced by my Android device and Kristen's Android device. Besides, 2 pupils could not install the app with their own Android devices due to old Android versions. It was important to define the available OS versions for testing for future testing. As for iOS, there were no issues regarding tracking.

Find the Correct

In the feedback session, pupils said they liked this game mode during the testing session but the game content was not enough. The tasks pool was too small, which had 7 tasks in total for this version. They wanted more tasks and different tasks. We identify this as the main target for improvement.

There were some AR tracking issues on Android devices. Nonetheless, the teacher said she would like to use the app as a tool for chemistry lessons. And, the pupils like the app for a common reason, which is the molecule visualisation in AR. There were some pupils

[46] https://www.nngroup.com/articles/quant-vs-qual/

[47] https://play.google.com/console/about/devicecatalog/

quotes on the blog[48]. For instance, a pupil said it was much cooler than a textbook and TV. These quotes were all about AR visualisation, which was identified as the strength of this app.

5.2.3 Improvements

From the above, focusing on improving the game play of *Find the Correct* game mode is important. A new level design has been added to this mode including different collections of tasks and a game point level system. Tasks are grouped into 3 different levels from easiest to hardest in Table 6.

Table 6. Find the Correct Task levels

Level	Task Hint on Screen	Target Card Group
1	chemical name (eg. Iron)	*Formula Card*
2	chemical formula (eg. O-O)	*Name Card*
3	chemical name	*Characteristic Card*

Regarding the game point level system, the initial point is 0. Each correct answer gives one point. Each incorrect answer will deduct all the current points back to zero. To level up, 5 points is required within the level. 5 tasks are grouped as one level because of the following reasons. Firstly, the concern of battery draining issue on mobile devices. During internal testing, the mobile devices for both platforms became hot after opening the AR scenes in *Futuclass AR.* It was observed that the battery power dropped significantly after several minutes. Thus, the design should avoid user to open the AR scene for a long period. Secondly, if the number of tasks required for level-up is too small like 2-3 tasks, the game level might become too easy for the player. The player might not enjoy the game since they could not experience the sense of achievement. After level-up, the point will reset to 0 at the new level.

As for tracking issues on some Android device models, several rounds of internal testings had been made after the testing. However, the build had no issues on our own testing devices. There was the possibility to change the AR Markers' format but then the game flow would have to be adjusted accordingly. As for further testing, it would be necessary

[48] https://sites.google.com/view/futuclassi-koosloome/arendusblogi

to look deep into ARCore tracking code in Kotlin programming language generated by Unity. The issue had been discussed with Futuclass and Kristen agreed to postpone for solving this issue.

5.3 Usability Testing #2

With the improvements made in *Find the Correct* game mode after the 1st usability testing, here it was to test the improved *Find the Correct* game mode. Another goal of this testing was to test the two different designs of *Reaction Balancing* game mode, which was not tested in the 1st usability testing. Due to the ARCore performance issue on Android devices, this testing was only targeting the iOS platform. Thus, we could have reliable game design testing with solid AR performance within the limited time scope. Like the previous testing, the blog of this testing is available in the Appendix - Accompany Files, written by the author of the thesis.

5.3.1 Physical Class Testing

Since the situation of the Covid-19 became better, the school opened again for the 9th and 12th graders. The testing session was not in virtual format this time but a physical class session with 9th graders on 5th May in Tartu Private School (TERA). There were 16 pupils. They were 9th-grade so they were the highest graders within our target group. It was believed that the testing would be valuable since physical class session allows pupils and teacher interacting with each other in person. This was beneficial to our testing especially since the strength of AR application is the interactions between virtual content, real-world environment and different players.

The original idea was to deliver the application via TestFlight to pupils' iPads. However, the pupils' iPads were managed by the Apple School Manager of the school so pupils could not download our app with TestFlight directly. After several testings with the IT department of the school, it was known that the school managed iPads could not install applications through TestFlight due to permission issues. The only way to let pupils have the app on their iPads was to publish the app as a custom app in AppStore but we did not have time to wait for the review approval from Apple. Thus, alternatively, Futuclass has helped to gather enough iOS devices.

Figure 48. Pupils playing *Futuclass AR* in the classroom

There were 16 9th-grade pupils. The class session was a Chemistry lesson in Estonian. The class's teacher Meelis Brikker was the facilitator of the testing. And, the education content specialist Arnold was the observer and representative of Futuclass. Pupils were grouped into small groups of 2-3 people for small-group testing, as shown in Figure 48.

5.3.2 Tools

In the previous testing, the feedback was collected through observations and verbal questions only during the testing session. It was found that we did not have enough time for the pupils to share their opinions in detail. Thus, it was suggested to create a feedback form to gather feedback on top of the observation. The feedback form was created using Google Forms. The form was created in English and translated into Estonian by Kristen. The forms in English and Estonian could be found in the Appendix - Accompanying Files. The form has two parts, which are the *Find the Correct* and *Reaction Balancing*.

The *Find the Correct* part includes different kinds of questions. This game was tested in the earlier testing so this was to test the updated version. The question “How much do you like this game mode?” is asked to learn if pupils find the game fun.

The *Reaction Balancing* part is in a form of **comparative usability testing**. In the Design chapter, the 2 different designs of reaction balancing game were introduced. We have deferred the design decision until now and test the 2 available versions in this testing. In comparative usability testing, it is suggested that the testers can provide much more information and better feedback when they can experience multiple designs during the testing [13]. There are different types of comparative usability testing including preference testing, performance testing and A/B testing [14]. We had considered randomising the testers into groups for detail performance testings like record the task completion time of *Reaction Balancing* tasks of different design versions. However, after discussion, we preferred to organise the testing session in a format of a whole class due to the time limit of the class. The pupils would follow Arnold to try the *Futuclass AR* step by step in a class format. Thus, it was not a valid randomised controlled testing like A/B testing but a simple preference testing.

Preference testing is a way of qualitative research. The common way is to ask questions. Questions like “Which way of reaction balancing was easier to understand for you?” and “Which version did you prefer?” have been asked to let pupils make choices between them. The open-end question “Why did you prefer this version?” has been asked to understand the reasons behind their choices.

At the end of the form, two open-end questions “Describe what was broken or didn't work well.” and “Have you got any recommendation on how to improve the app?” were asked to help us understand different undiscovered bugs and suggestions for improvements.

5.3.3 Results

This was the second usability testing. In the previous usability testing, there were poor image tracking issues on few Android devices. Testing with only iOS devices allowed us to test our game with a more solid image tracking performance. Detail testing data regarding the game design was collected without affecting by the tracking issues.

Find the Correct

In general, most of the testers could clear all the levels of the game mode *Find the Correct* and with positive feedback. In Figure 49, more than 90% of the testers could pass all the levels.

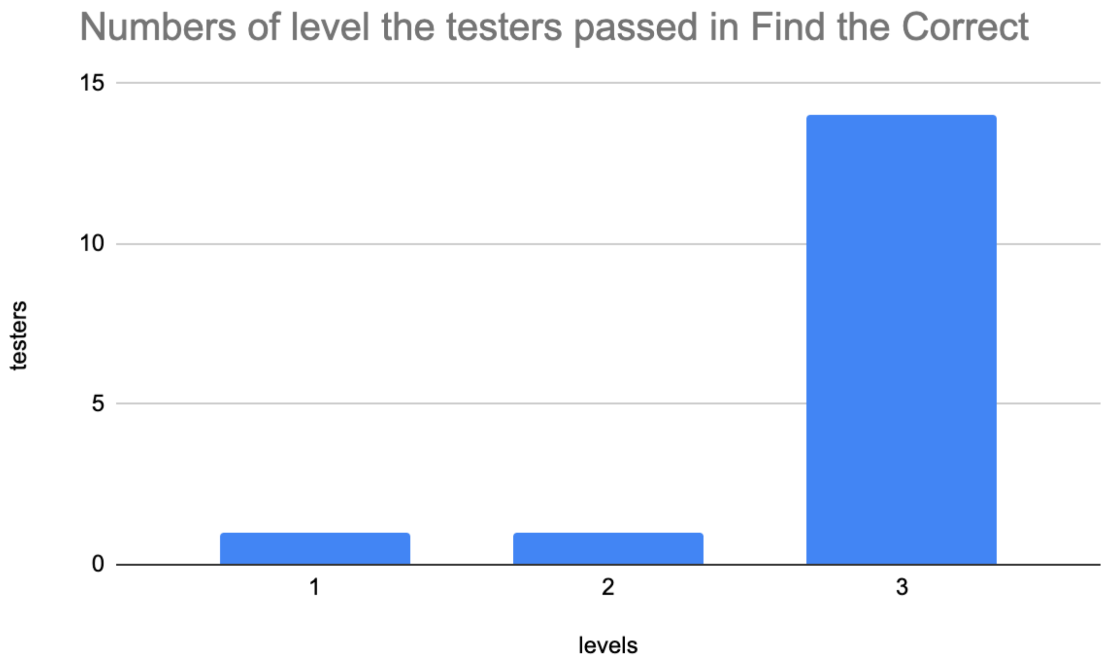

Figure 49. Levels passed in *Find the Correct*

The form result shows that testers think the game is easy or fairly easy in the KDE plot (Figure 50). This is expected since they are 9th-grade pupils. This result is supported by the fact that nearly most of them could pass all the levels of the game in Figure 49. During the class, some pupils said that it was a good format for rehearsing study topics.

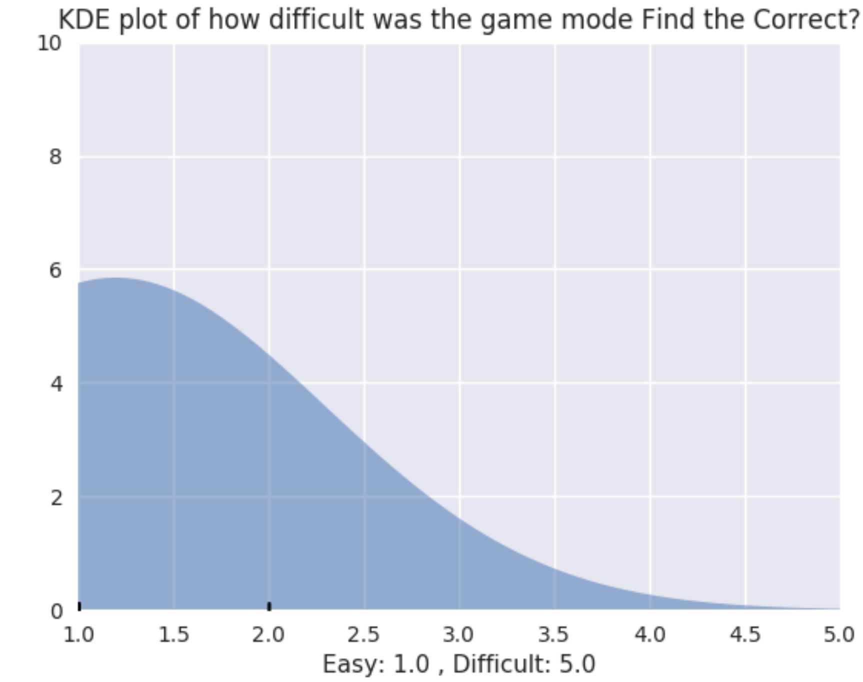

Figure 50. Difficulty of *Find the Correct*

The KDE plot below shows that the majority of the testers like the *Find the Correct* game mode as shown in Figure 51. New content and game levels have been added to this game mode after the first usability testing helped to produce this positive result.

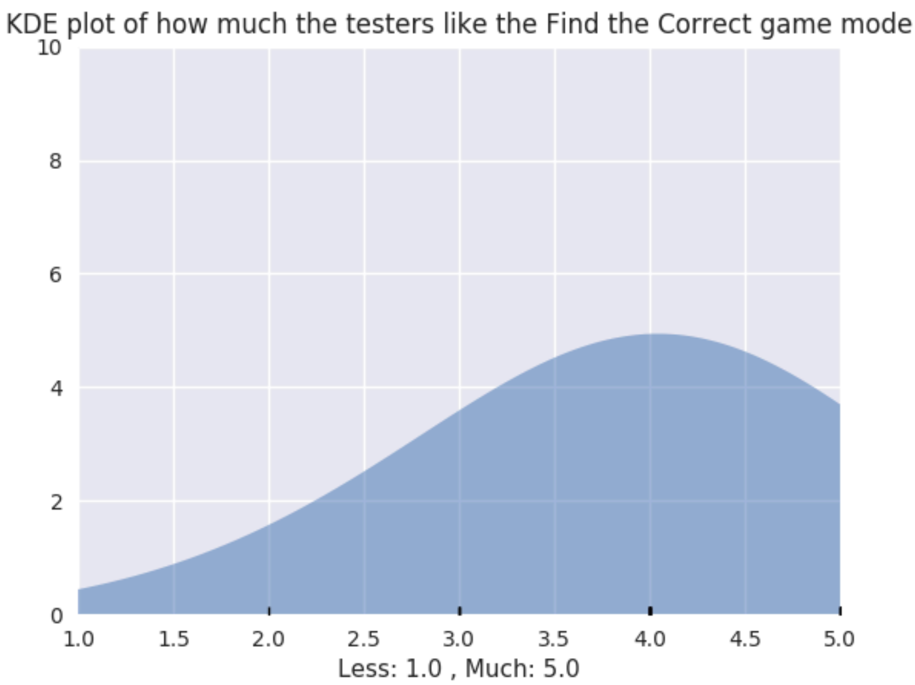

Figure 51. How much the testers like the game

Reaction Balancing (On-screen vs AR)

According to the form result, more students prefer on-screen design (10/16) than the AR design (see Figure 52). Most of the students who prefer on-screen design think it is easier and simpler to use. Even though some students prefer AR design and they think it has better visualisation and it is more interesting.

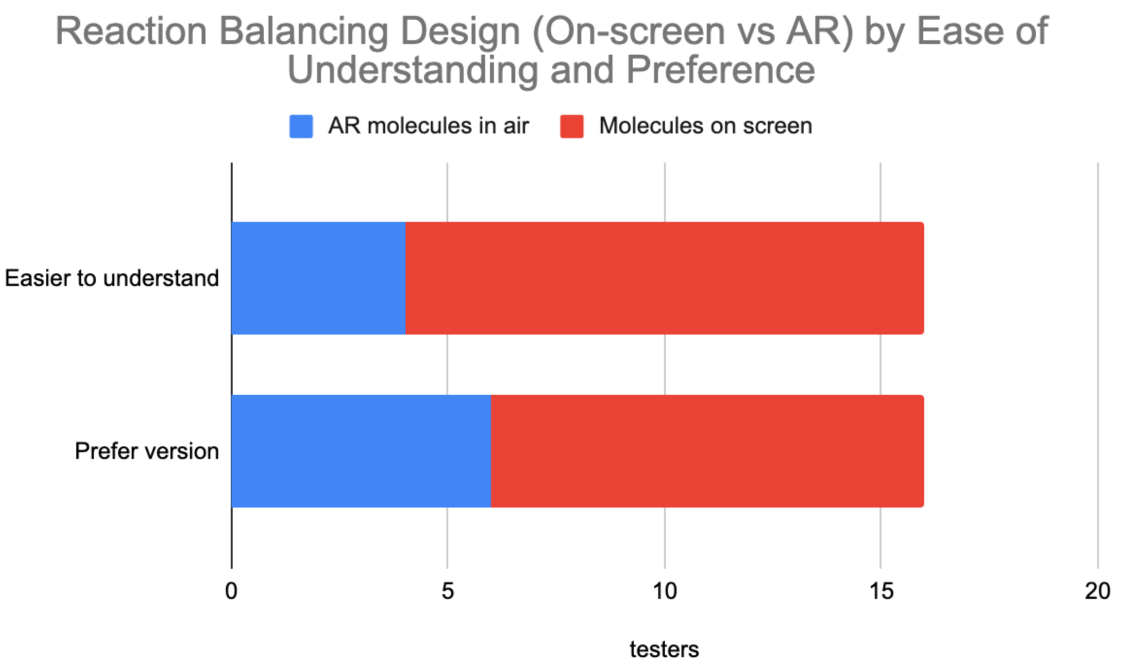

Figure 52. Chart of the research results of the 2 design versions

It is noted that this preference testing is not optimised since the testing orders were not randomised. The pupils tried the AR design first, and then tried the on-screen design. With a small sample of the usability test, this result is not a direct answer to our design problem but it could be a good reference for us [14]. Since we have learnt the reasons behind the preferences, there are now more supportive factors for us to decide on the design of the *Reaction Balancing*. This is discussed in Chapter 5.3.4.

Observations

The observation was done by Arnold physically in the classroom. There are some interesting facts to be mentioned. Firstly, pupils could remember the fun facts they learnt from the app. Phosphoric acid was the popular one when he asked what fun facts about the molecules they found interesting. This suggested that the pupils did not omit the fun fact on-screen after finding the molecule. There was concern about the *Find the Correct* not having a good didactic design in the earlier usability testing but this issue seems to be solved after the modification.

Another interesting fact was that even Arnold asked pupils to try to crash the game, some pupils still focus on balancing the equations instead. The equation balancing tasks seem to provide fun to pupils.

One more fact was that when the students know how to balance reaction equations, they do not pay a lot of attention to the number of models on the screen. They want to do the balancing as quickly as possible using the formula. It is good to see the pupils could learn how to balance the equations through the reaction balancing game using the formula. However, this also implies that the learning process became much like an exercise, but not a game. The pupils might not enjoy this process after several iterations. This is an important case to be considered in future improvement.

5.3.4 Improvement Suggestions

There are some suggestions that have been made for the *Molecular Sandbox* mode, the UX design and the *Reaction Balancing* game mode. They were being implemented or under review with Futuclass.

Reaction Balancing

During this thesis, the review of the design between the 2 versions (on-screen, AR) was not complete. There were 3 options:

1. Keep On-screen Reaction Balancing only

2. Keep AR Reaction Balancing only

3. Keep them both and allow the player to choose

The on-screen version is considered to be more preferred from the result of Usability Testing #2. However, it is not suggested to consider it as a determinative factor for this design problem. The purpose of comparative usability testing is not to allow users to make every design decision but it is more about gathering information so we can evaluate and choose the best aspects of two designs [13]. Following this principle, the improvement suggestions for each of the design versions are proposed.

For the on-screen version, it is identified that the visualisation provided by the AR version is better than that of the on-screen version. We could improve the on-screen version by making better 3D visualisation on-screen and add animation to the molecule models on-screen. Additionally, buttons on the screen could be replaced by dragging and dropping molecules. It is to draw the player's attention from the equation back to the 3D molecule models to make the game process more lively.

For the AR version, we have to simplify the mechanics of the controls to make it easier to play. Removing the on-screen add remove buttons and assign them to the molecule models in AR could be a solution. Besides, the tracking performance issues on certain Android devices have to be considered. The research about tracking performance on Android devices continues.

Keeping them both could be an option but it requires much more development cost. It is better to set priority between the 2 versions and focus on either one.

Molecular Sandbox

Currently, the M*olecular Sandbox* is not in priority. There was feedback saying that this visualisation mode could be turned into a learning mode featuring the periodic table. By now, the M*olecular Sandbox* shares the same *Chemical Cards* with the *Reaction Balancing* game mode. There are 118 elements in the periodic table. There are 2 design approaches we could start with.

The first design is to prepare *Chemical Cards*. Adding 118 markers based on our current design is not practical due to the curse of image similarity. However, if more unique elements could be added to the cards (eg. real-life pictures of the chemical elements), it could be possible. Alternatively, the *Chemical Cards* might not represent one single element but a group of elements based on the chemical properties. Then, the numbers of the cards could be greatly reduced.

Another design is to turn it into markerless AR. Like *Arloon Chemistry*, it could share a similar design but provide some more features in AR like forming molecules freely by dragging and dropping the molecules in AR together.

Besides, to enrich the learning process, more content about the molecules could be added into this mode for pupils to read freely. They could be implemented as some learning content on a webpage. Then, these online learning materials could be shared by both *Futuclass AR* and *Futuclass Hub*.

Polishing

There was feedback about the colour tone of the game being grey making the game. This was usually the last step in game development. This prototype had not come to this step yet. Even though, here are some suggestions. Firstly, it is to design a theme for this game. It could be a "laboratory-like" theme like the *Futuclass Hub*. Then, tune the colour combination to match the theme. Secondly, the text instructions shown at the beginning of each of the game modes could be replaced by infographic or video animation. Thirdly, sound effects have to be added to the game. Without any sound effects, the player focusing on an unbalanced equation is like solving a question in a written exam. Music is powerful when it comes to evoking emotion from players [15].

The two usability testing sessions have provided much useful information for improvements. Problems were identified and some improvements had been made while some issues were still under investigations. Shortly, I and Futuclass would like to launch a mass-scale impact testing. It will focus on measuring the educational values of *Futuclass AR*.

6. Conclusion

In this thesis, the AR learning game *Futuclass AR* has been developed. This game is an AR version of the DLC *Reaction Balancing I* of the VR game *Futuclass Hub* developed in Estonia by Futuclass OÜ. It was a side project to them targeting at the mobile AR market.

The development of *Futuclass AR* was carried out in a team of three members. These members were the author of this thesis, Kristen Tamm, who, being a supervisor of this project and customer representative, and Arnold Rein Tatunts, responsible for designing educational contents in the project.

Futuclass AR is an AR learning game that let the player learn and rehearse chemistry topics through the interactions between the player and the *Chemical Cards* - AR Markers designed in this thesis. All of the markers are designed as learning materials.

The *Reaction Balancing* in *Futuclass AR* was designed into 2 different versions, which are the on-screen design and AR design. Two new game modes - *Molecular Sandbox* and *Find the Correct* were designed. The former provides a free-to-explore environment for molecule visualization in AR. The latter is a task-based game mode, which provides the player with a hint to find the correct card. The performance tuning of an AR application was not easy since there were a lot of factors behind like marker design, tracking logic or hardware issues. Besides, it is time-consuming since the *Chemical Cards* have to be tested on mobile devices individually. Some target image analysing tools had been used to help study some markers' issues.

The first round of usability testing was done in a virtual class format on the 1st of April 2021 due to the Covid-19 pandemic. The testing session was a chemistry lesson with 7th-grade pupils and their teacher. The game mode *Find the Correct* was the main target to test. A poor image tracking issue was found for few Android devices. It was found that the game mechanics of *Find the Correct* has potential but lack game content. To enrich the content, game levels were been created. New types of tasks and *Chemical Cards* were designed.

After the situation of the pandemic became better, a physical class testing session was finally done as the second round of usability testing. It was on the 5th of May 2021. 16 9th-grade pupils and their teacher participated in the testing. The result showed that pupils enjoy playing the updated *Find the Correct* game mode. Besides, the result indicated that the pupils prefer the on-screen design of R*eaction Balancing* more because they think it is simpler and easier to play. The design decision is still in review with Futuclass. Nonetheless, some suggestions had been made based on the result evaluation.

Futuclass AR has potential. The testers like it and there is still room for improvement. In the future, the development of the *Futuclass AR* will continue.

I would first like to thank my supervisor, Raimond-Hendrik Tunnel, whose expertise was invaluable in guiding the writing process and formulating the research methodology. Your insightful feedback brought the work *Futuclass AR* to a higher level. Thanks to Arnold Rein Tatunts, who provided professional chemistry educational content and helped with usability testing. Thanks to all the testers for trying out the game. Thank Gleb Skibitsky for providing the 3D model assets of the molecules and sharing professional knowledge of Unity with me. Special thanks to Kristen Tamm, as my supervisor, who guided me and provided the opportunity for me to work on an AR game development project with Futuclass.

1. References

[1] Koster, R. (2013). *A theory of fun for game design.*

[2] *Mobile Augmented Reality Market by Device Type, Offering, Application | COVID-19 Impact Analysis | MarketsandMarkets™*. Marketsandmarkets.com. (2021). Retrieved 12 May 2021, from https://www.marketsandmarkets.com/Market-Reports/mobile-augmented-reality-market-174800140.html.

[3] Li J., van der Spek E.D., Feijs L., Wang F., Hu J. (2017) Augmented Reality Games for Learning: A Literature Review. In: Streitz N., Markopoulos P. (eds) *Distributed, Ambient and Pervasive Interactions*. DAPI 2017. Lecture Notes in Computer Science, vol 10291. Springer, Cham. https://doi.org/10.1007/978-3-319-58697-7_46

[4] Wouters, P., van Nimwegen, C., van Oostendorp, H., & van der Spek, E. (2013). A meta-analysis of the cognitive and motivational effects of serious games. *Journal Of Educational Psychology*, 105(2), 249-265. https://doi.org/10.1037/a0031311

[5] Laine, T., & Lindberg, R. (2020). Designing Engaging Games for Education: A Systematic Literature Review on Game Motivators and Design Principles. *IEEE Transactions On Learning Technologies*, 13(4), 804-821. https://doi.org/10.1109/tlt.2020.3018503

[6] Kapp, K. (2012). The gamification of learning and instruction. San Francisco: Pfeiffer.

[7] Johnson, D. (2021). *What is augmented reality? Here's what you need to know about the 3D technology*. Business Insider. Retrieved 12 May 2021, from https://www.businessinsider.com/what-is-augmented-reality.

[8] Radu, I. (2014). Augmented reality in education: a meta-review and cross-media analysis. *Personal And Ubiquitous Computing*, 18(6), 1533-1543. https://doi.org/10.1007/s00779-013-0747-y

[9] Tekinbaş, K. S., & Zimmerman, E. (2003). *Rules of play: Game design fundamentals*. Cambridge, Mass: MIT Press.

[10] Zubek, R. (2020). *Elements of game design*. MIT Press.

[11] Nystrom, R. (2014). *Game Programming Patterns*. Genever Benning.

[12] Moran, K. (2019). *Usability Testing 101*. Nielsen Norman Group. Retrieved 12 May 2021, from https://www.nngroup.com/articles/usability-testing-101/.

[13] Ross, J. (2021). *Conducting Qualitative, Comparative Usability Testing :: UXmatters*. Uxmatters.com. Retrieved 12 May 2021, from https://www.uxmatters.com/mt/archives/2017/03/conducting-qualitative-comparative-usability-testing.php.

[14] Anderson, N. (2021). *Comparative Usability Testing: A Key Method for Actionable Design…*. dscout.com. Retrieved 13 May 2021, from https://dscout.com/people-nerds/comparative-usability-testing.

[15] K, D. (2021). *Video Game Sound Design | Beginner's Guide*. The Ultimate Resource for Video Game Design. Retrieved 13 May 2021, from https://www.gamedesigning.org/learn/video-game-sound/.

Appendix

I. Glossary

AR-Marker - They are visual cues which trigger virtual contents in AR.

Augmented reality (AR) - An immersive experience of a real-world environment in which computer-generated perceptual knowledge is used to improve the objects in the real world, often through several sensory modalities like visual and auditory.

Chemical Cards - AR markers designed in this thesis.

Find the Correct - A task-based game mode. To complete the task is to find the card asked by the hint.

Flyweight - A software design pattern that allows you to fit more objects into the available memory by sharing common parts of them across multiple objects rather than keeping all of the data in each object.

Game Assets - Anything that goes into the game like visuals, animations and game logic

Game-Based Learning - GBL is using a game as part of the learning process.

Gamification - it is to turn the whole learning process into a game.

Game Objects - They are the fundamental containers for the game assets in Unity.

Learning Process - The process of acquiring new understanding, knowledge, behaviours , skills, values, attitudes, and preferences.

Molecular Sandbox - a free-to-explore mode for the user to observe different molecule models in AR with the *Chemical Cards*

Object Pooling - A software design pattern in which a collection of objects is initialised before they are required. These items will not be removed and will be held until they are needed. The objects are returned to the "pool" when they are no longer needed.

On-screen - On a mobile device screen

Reaction Balancing - The process of balancing an unbalanced chemical equation

Transform - It is the component of any Game Object for storing the position, rotation, scaling and parenting information about of the Game Object in Unity.

II. Accompany Files

- *FutuclassAR.mp4* - the video of the gameplay of *Futuclass AR*
- *Ver 0.0.7 AR Markers.pdf* - the AR Markers file used in Usability Testing #1
- *Ver 0.0.8 Reaction Balancing Markers.pdf* - the AR Markers file for the *Reaction Balancing* game mode and *Molecular Sandbox*, used in Usability Testing #2
- *Ver 0.0.8 Find the Correct Markers.pdf* - the AR Markers file for the *Find the Correct* game mode, used in Usability Testing #2
- *Futuclassi koosloome - Arendusblogi.pdf* - the development blog of *Futuclass AR*. The 1st usability testing summary was written by Kristen Tamm. The 2nd usability testing summary was written by the author of this thesis.
- *Futuclass AR usability testing feedback form(05.05) Google Forms.pdf* - the feedback form used in the 2nd usability testing in English
- *Futuclassi liitreaalsuse õpirakenduse testtund (05.05) Google Forms.pdf* - the feedback form used in the 2nd usability testing translated into Estonian by Futuclass
- *Futuclassi liitreaalsuse õpirakenduse testtund (05.05) EST Form responses.csv* - the feedback form responses
- *Futuclassi liitreaalsuse õpirakenduse testtund (05.05) ENG Form responses.csv* - the feedback form responses translated into English by Futuclass

Build files:

futuclass_0.07.apk - The Android build of version 0.0.7. It is work with the AR Markers file *Ver 0.0.7 AR Markers.pdf.*

Version 0.0.8 is available on iOS only at the moment and it is not included here due to iOS App Security reason. The build could be tested via TestFlight on AppStore. Please send an email to Kristen Tamm to request testing (kristen@futuclass.com).

III. Source Code

Available via request on https://gitlab.com/futuclass/reaction-balancing-ar. Please send an email to Kristen Tamm to request access (kristen@futuclass.com).

IV. Game Screenshots

Figure 53. Screenshot of the welcome screen

Figure 54. Screenshot of the main menu

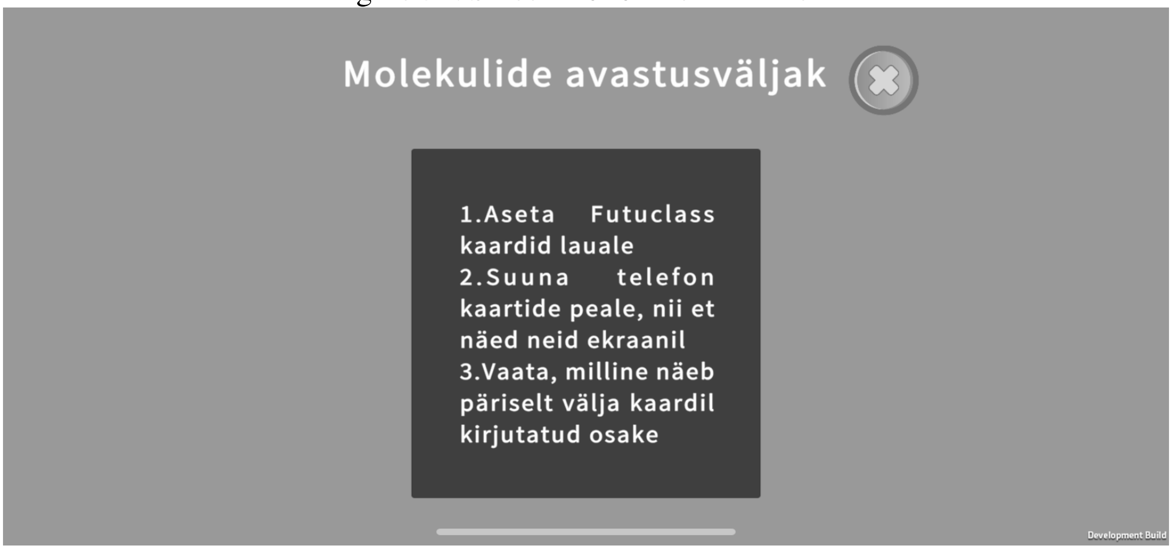

Figure 55. Screenshot of the instruction of *Molecular Sandbox*

Figure 56. Screenshot of *Molecular Sandbox*

Figure 57. Screenshot of Fun Fact of the molecules in *Find the Correct*

Figure 58. Screenshot of *Reaction Balancing* Menu

Figure 59. Screenshot of Step 1 - Finding reagents in *Reaction Balancing*

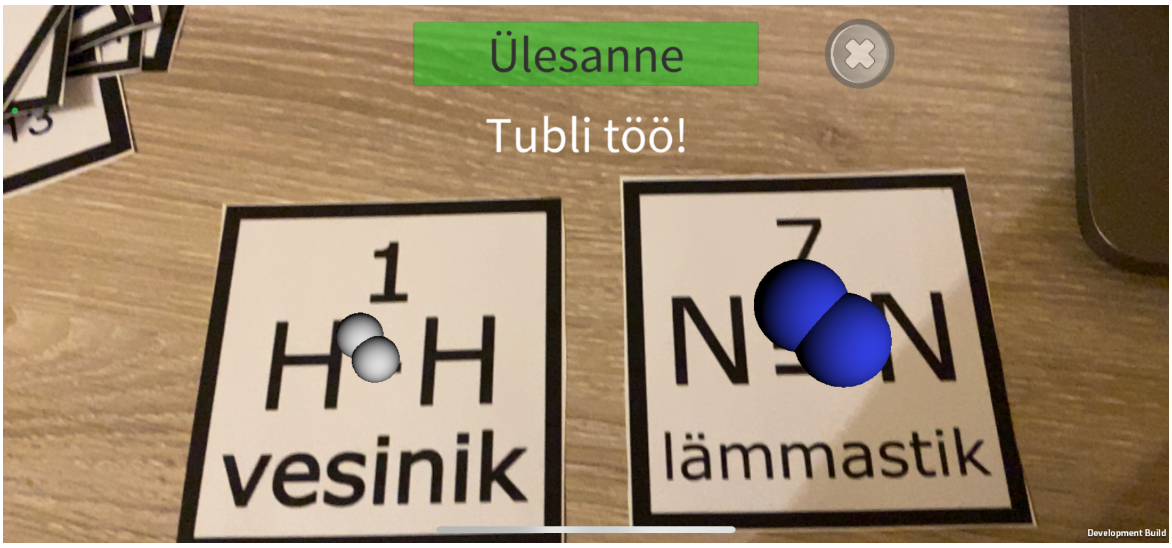

Figure 60. Screenshot of successfully completing the step 1 in *Reaction Balancing*

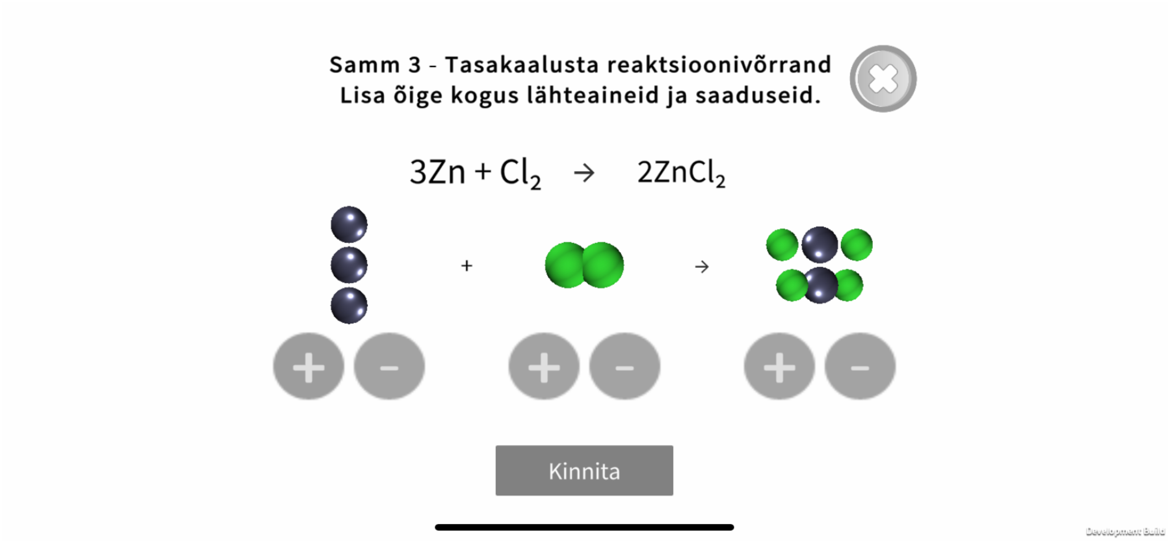

Figure 61. Screenshot of the on-screen design of Step 3 - *Reaction Balancing*

Figure 62. Screenshot of the AR design of Step 3 - *Reaction Balancing*

V. Version Control

Version	Description
0.0.1	Game menu and the *Molecular Sandbox* were created. 1st design version of AR markers were used.
0.0.2	*Find the Correct* game was created. The answer is determined by pointing the device's camera to the desired card.
0.0.3	On-screen *Reaction Balancing* was created. 2nd design version of AR markers were used.
0.0.4	Tapping molecule function was created to replace the pointing function for molecule selection. 3rd design version of AR markers were used.
0.0.5	*AR Reaction Balancing* was created.
0.0.6	Image tracking performance tuning was done for Android ARCore. The game had been translated into Estonian.
0.0.7	Fixes about some content of the markers. Tested in usability testing #1 on Android and iOS.
0.0.8	Updated *Find the Correct* game. Added multi-reference image library switching logic. Tested in usability testing #2 on iOS.

VI. License

Non-exclusive licence to reproduce thesis and make thesis public

I, Hei Chun Shum,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to

reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

Futuclass AR,

supervised by Raimond-Hendrik Tunnel and Kristen Tamm.

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Hei Chun Shum

13/05/2021