

# Deutsch Brawl

## Praktikum: Serious Games in Extended Reality

Supervised by Mengdi Wang  
mengdi.wang@tum.de

Ankur Deria  
i.am.ankur.deria@tum.de

Maximilian Mitterrutzner  
maxi.mitterrutzner@tum.de

Yiyang Xie  
xieyi@cit.tum.de

**Abstract**—In recent years, as the world’s game industry continues to grow, seven of the world’s top ten technology companies with the highest market value have shown interest in setting up game departments. The trend of accelerating the creation of games is even more obvious. [1] Game engines are pieces of software designed to aid in the creation of games. Different game engines have their own advantages and disadvantages, with a very prominent example being the Unity game engine. It enables developers to quickly create and optimize content because of its high degree of flexibility. [2]

In this project we used the Unity engine for the development of a serious game. Our goal was to create a fun and engaging AR card game, where learning the German language happened almost automatically. Challenges to overcome included object recognition, plane detection, prompt engineering, declension table gathering and more. The programming language that we use throughout the development of the game is C#. In addition to the Unity-internal tools, we used several external resources to create the serious game.

**Index Terms**—unity3d; game development; c#; serious games; object recognition; plane detection; prompt engineering

## 1. Background and Related Work

Today anyone can leverage powerful open source frameworks or game engine in order to develop any game imaginable, whether they present as a company or as an independent game developer. Currently more than 60% of the world’s games are developed in Unity. [3] Various highly successful games like Temple Run 1&2, Hearthstone Legends, Pokemon Go, League of Legends (Mobile), Genshin Impact, and others have been developed with the game engine Unity. [4]

Unity currently supports most major platforms, including iOS and Android and even providing VR and AR capabilities. Because of Unity’s strong cross-platform compatibility, it allows developers to do one-time development, multi-platform deployment. In addition, nowadays the trend of GAAS (Game as a Service) is becoming more and more significant. [5] Therefore, Unity engine is not only used in the game industry, but also in the medical, industrial, commercial, media and education fields. Unity is not only a game creation tool, but also a real-time 3D

interactive platform. This makes Unity the most likely foundation for building virtual worlds in the future. [3]

### 1.1. AR & AI

Augmented Reality (AR) and Artificial Intelligence (AI) are two separate concepts. However, in recent years, the relationship and interaction between them have become increasingly close.

AR is a technology that combines the virtual world with the real world. AR enables users to interact with virtual objects in a realistic environment.

AI is a technology that enables computers to think, learn and make decisions as intelligently as humans.

With the development of AR and AI technologies, the integration between them has become a popular topic of research and application. AI is good at many operations that benefit AR, such as tracking objects, creating models of the 3D world, recognizing object, detecting plane and making judgments about them. Deep learning models in AI can help AR systems interpret complex environments. In the end it can produce more realistic models and help people with scenario-based interactions, leading to optimized immersive experiences in AR.

### 1.2. Object Recognition

The object recognition is a generic term for a fundamental study in the field of computer vision. Its task is to recognize what object is in the image and report the position and orientation of this object in the scene represented by the image. Current object recognition methods can be grouped into two categories: model-based or context-based recognition methods, 2D object recognition or 3D object recognition methods. Additionally, there are 4 criteria recognized by most researchers: robustness, correctness, efficiency and scope.

Object detection has witnessed remarkable advancements with the emergence of deep learning models like YOLO (You Only Look Once) and others. These models leverage convolutional neural networks (CNNs) to efficiently detect and localize objects within images. However, while YOLO and similar models excel at detecting objects, they often lack the ability to understand contextual relationships or generate textual descriptions of the detected objects.

In contrast, GPT-4 Vision represents a significant evolution in the realm of object detection by seamlessly integrating natural language understanding with visual comprehension. While YOLO focuses primarily on identifying objects and their bounding boxes within images, GPT-4 Vision extends this capability by comprehending the context surrounding the detected objects and generating human-like text to describe them.

### 1.3. Plane Detection

Plane detection using AR Foundation in Unity is a fundamental aspect of creating augmented reality (AR) experiences where virtual content interacts with real-world surfaces. AR Foundation is a Unity framework that enables developers to build cross-platform AR applications compatible with both Android and iOS devices.

Plane detection refers to the ability of AR systems to recognize flat surfaces in the physical environment, such as floors, tables, and walls, and generate virtual planes that align with these surfaces. This capability allows developers to anchor virtual objects onto real-world surfaces, creating realistic and immersive AR experiences.

### 1.4. Networking

Networking in games serves as a cornerstone for facilitating multiplayer interactions, social engagement, and real-time gameplay experiences across distributed environments. It operates through the exchange of data between multiple devices over networks, whether it's the internet or local area networks (LANs), enabling players to interact with each other and the virtual game world seamlessly. Most multiplayer games employ a client-server architecture where a central server manages the game state and facilitates communication between multiple clients. This ensures consistency and fairness by enforcing game rules and resolving conflicts between players' actions. Additionally, some games may utilize peer-to-peer networking models, where each player's device acts as both a client and a server, communicating directly with other players' devices. However, this approach may introduce synchronization challenges and security concerns.

To mitigate latency issues, developers implement lag compensation techniques such as client-side prediction and server reconciliation. Networking APIs and libraries, like Unity's Multiplayer Networking or Unreal Engine's Online Subsystem, streamline the implementation of networking functionality by providing pre-built solutions for common tasks such as connection management and data serialization. In Unity, several networking frameworks empower developers to implement multiplayer functionality seamlessly. Among these frameworks, Photon Unity Networking (PUN), Mirror, and Unity Networking Transport stand out as popular choices. Photon Unity Networking (PUN) simplifies the process of adding multiplayer features to games, offering a cloud-based network infrastructure and comprehensive API.

### 1.5. Purpose and Significance of the Project

The AR serious game project developed using Unity for mobile devices holds significant importance and pur-

pose across various domains. Firstly, it serves as an innovative educational tool, leveraging gamification techniques and augmented reality (AR) technology to deliver immersive learning experiences. By allowing users to interact with virtual objects overlaid onto their physical environment, the game promotes active participation and deeper understanding of complex subjects. Secondly, the project aims to drive positive behavioral change by simulating real-life scenarios and providing feedback on user actions. Whether it's promoting physical activity, environmental awareness, or health and safety practices, the AR serious game fosters engagement and motivation for users to adopt desired behaviors. Additionally, the accessibility and inclusivity offered by mobile AR technology ensure that the game can reach a diverse audience, regardless of their location or technological proficiency. By making immersive experiences accessible via smartphones, the project enhances accessibility and promotes inclusivity in educational and entertainment contexts. Finally, the development of an AR serious game represents an opportunity for innovation and technological advancement, pushing the boundaries of interactive entertainment and showcasing the potential of AR technology in diverse applications. Through continuous iteration and refinement, the project aims to deliver compelling AR experiences that inspire learning, foster positive behaviors, and entertain users of all ages.

### 1.6. Definition of Serious Games

In addition, we can't ignore a new concept, Serious Games. Serious Game, a type of video game. Originally defined as "games for application purposes". [6] Specifically, it refers to games that teach knowledge and skills, provide professional training and simulation as their main content. Serious games originated in the 1980s and have been widely used in military, medical, industrial, urban planning, education, training, scientific exploration and many other fields. [7] In addition to entertainment, their main feature is to have a clear functional or applied purpose. The purpose is to emphasize fun and competition, and to bring educational value. [8]

**The definitions for serious games [9]:**

- 1) Interactive technological applications that go beyond traditional games for purposes. Including helping with personnel training, policy discussions, analysis, visualization, simulation, education, health, medicine, etc.
- 2) Can solve other problems, such as training military personnel to adapt to foreign cultures, enabling people to work as a team at work, and teaching children to understand scientific principles.

From the perspective of game developers, serious games have the advantages and characteristics of low application cost, high application benefit and high popularity.

From the player's point of view, serious games have usability, fun and interactivity. If there is no fun, there will be no difference between the so-called serious games and textbooks. Therefore, serious games also have the essential

characteristics of traditional video games at the same time. [10]

## 2. Development Tools and Environment

The programming language used for the system was C# and the game engine used was Unity 2022.3.13f1 (64-bit). Visual Studio 2022 and JetBrains Rider were used as the IDEs for writing scripts. In addition, GPT-4 was used for object recognition and GitLab for versioning the project. Several Android-based smartphones were used for testing the AR game.

### 2.1. C#

C# is released by Microsoft. C# is an object-oriented programming language. Developed by Anders Hejlsberg and his team. It is approved and recognized through Ecma and ISO. C# is built very close to the traditional high-level languages C and C++ and is an object-oriented programming language. On the other hand, it is very similar to Java and has many powerful programming features, which makes it popular among programmers. [11]

### 2.2. Unity & Unity3D

The game engine Unity3D is a software that integrates game development, interactive development, virtual reality, augmented reality and other interactive development. Nowadays, Unity has launched various subscription services such as Unity Cloud and Unity Polyspatia. In the future, Unity will also release the public version for Unity Muse, which can help developers to iterate and improve their content faster. There's also one Unity Sentis, which brings AI models to Unity and uses AI to create new features in games. Unity Sentis is currently in beta phase and will launch with the Unity 6 engine in 2024. [12]

### 2.3. Visual Studio & JetBrains Rider

Microsoft Visual Studio(VS) and JetBrains Rider are both powerful IDE tools used by programmers today. For the first one, the latest release of Visual Studio 2022 is a 64-bit IDE tool. [13] Both of these IDEs can develop applications for all major operating systems and clouds. They also have features that allow for efficient programming, debugging, diagnostics, custom extensions, synchronized collaboration, etc. [14]

### 2.4. GPT-4

GPT-4 is a large multi modal model, which accepts image and text as the inputs, then emits text as the outputs. Currently, it is not as capable as a human in many real-world scenarios, but exhibits human-level performance on a variety of professional and academic benchmarks. The generative pre-trained models (GPTs) by OpenAI keep being developed and updated. The latest public version, GPT-4, is more reliable, more creative and capable of handling more nuanced instructions than last version, GPT-3.5. [15]

## 2.5. AR Foundation

AR Foundation, a pivotal component within Unity, revolutionizes the development landscape for augmented reality (AR) experiences by offering a unified framework compatible with various platforms, notably iOS and Android. Launched by Unity Technologies, AR Foundation serves as a versatile interface that harmonizes the disparities between ARCore for Android and ARKit for iOS, streamlining the development process.

With AR Foundation, developers can craft AR applications without the need for platform-specific code, allowing for efficient deployment across multiple devices. The framework encompasses a rich set of features essential for building immersive AR experiences. This includes robust object tracking capabilities, enabling virtual content to anchor seamlessly to real-world objects, and plane detection functionality, which identifies surfaces within the environment for precise placement of virtual assets. Furthermore, AR Foundation empowers developers with tools for light estimation, raycasting, hit testing, and more, facilitating interactions between virtual and real-world elements. Its cross-platform compatibility ensures broad accessibility and ease of deployment, making AR development more accessible to creators across diverse platforms and devices.

With AR Foundation, Unity continues to democratize AR development, fostering innovation and creativity in the rapidly evolving AR landscape.

### 2.6. Photon Unity Networking 2

Photon Unity Networking 2 (PUN 2) is a powerful networking solution for Unity developers seeking to implement multiplayer functionality into their games with ease and efficiency.

Developed by Exit Games, PUN 2 builds upon the success of its predecessor, offering enhanced features, improved performance, and seamless integration with Unity's ecosystem. With PUN 2, developers can create multiplayer experiences across various platforms, including PC, mobile, and consoles, without the need for complex networking infrastructure. The framework provides a robust client-server architecture, facilitating communication between game clients and a central server, which manages the game state and ensures synchronization between players.

PUN 2 offers a range of features, including room management, player matchmaking, and synchronization of game objects across networked instances. Additionally, PUN 2 includes support for authoritative server architecture, enabling developers to implement server-side logic to prevent cheating and ensure fair gameplay. With its user-friendly API and comprehensive documentation, PUN 2 empowers developers to build engaging multiplayer experiences quickly and efficiently, making it a popular choice among Unity developers for implementing networking functionality in their games.

### 2.7. AR Foundation Remote 2

AR Foundation Remote 2 is a significant advancement in the realm of augmented reality (AR) development, specifically tailored for Unity developers.

Developed by Kyrylo Kuzyk [16], AR Foundation Remote 2 streamlines the AR development workflow by enabling developers to iterate and test AR experiences directly within the Unity Editor. This powerful tool eliminates the need for constant deployment to physical devices, significantly accelerating the development cycle and enhancing productivity.

AR Foundation Remote 2 works by establishing a connection between the Unity Editor and a mobile device, allowing developers to preview and interact with AR content in real time on their mobile devices while making adjustments in the Editor. This seamless integration facilitates rapid prototyping, debugging, and fine-tuning of AR applications, ensuring a smoother development process. With features such as live reloading of AR scenes and support for device-specific configurations, AR Foundation Remote 2 empowers developers to create high-quality AR experiences with greater efficiency and precision. Overall, AR Foundation Remote 2 represents a valuable tool for Unity developers, enabling them to unleash their creativity and bring immersive AR experiences to life more effectively.

### 3. Project Development Analysis

At the beginning of the project/practicum, we have tried to analyze the requirements of this serious game in the following ways:

#### 3.1. Requirements Analysis

First of all, it is important to consider what platform the game will be played on. Because there are many different devices for playing games, different devices can play different games and bring different experiences to the players.

- 1) PC relies on mouse and keyboard to operate games.
- 2) Game console, such as XBox, PS5, relies on game controllers to operate games.
- 3) Mobile phone rely on the touchscreen to play games.

Secondly, there are various types of games, such as RPG (Role Playing Game), FPS (First Person Shooting Game), TPS (Third Person Shooting Game), TAB (Table Game), etc.

In addition, the length of time spent playing the game is also important. In the fast-paced life, many players prefer games with high interactivity and appropriate duration. Surveying the length of a single game for some games yielded the following results: League of Legends: about 30 minutes; Hearthstone Legends: about 15 minutes; csgo is fixed at 2 minutes a round. Through the survey, it can be seen that shorter game times are likely to have greater advantages. Therefore, how to control the duration of the game and ensure its interactivity is another important point that needs to be taken care of in the design process of this game. [17]

After analyzing the above requirements, we decided to create a serious game of tabletop cards called Deutsch

Brawl. It is a 3D game with simple content, particularly easy to play and user-friendly. [18] Players can scan real objects with the camera. Through object recognition and plane detection, the physical object is transformed into a German word card. Then, the players can choose randomly generated grammar cards, such as different articles, grammatical tense or subject-verb-object. With these grammar cards the players can combine them with the object card that fits the grammar to add different buffs. These buffs can increase their attack power, life value or have different effects. Ultimately, a turn-based tabletop card game of strategic attack and defense is formed with a grammar and vocabulary learning aspect.

#### 3.2. Technical Analysis

The game engine Unity is used to create the whole game, and the programming language is C#. In the development process, the main technology is mainly through the C# programming. We write scripts and develop the functions in the game engine, Unity, to carry out the relevant operations. [19] Firstly, the various types of components needed to be added, such as: Audio Source components, Rigidbody components, Animator components, etc. Secondly, we need the introduction of a variety of extensions to open source packages, such as: the use of different AI tools to recognise the object, to detect the plane and German grammar or syntax, the use of YOLO (deep learning models), ChatGPT (GPT-4) [15], or other perception tools to realize other functions like Prompt engineering.

Most of the functions used in this project has attached into the game engine Unity in the many years of its development. Most of the knowledge used in this project can be learned in the university courses or online videos. Overall it is achievable for students of related majors. Therefore, it is completely feasible technically.

#### 3.3. Educational Purpose Analysis

The purpose of our developed serious game is learning German. If people want to learn a language quickly and efficiently, vocabulary and grammar are the most important points for the beginners.

Vocabulary, the process of memorizing foreign words is the most painful. This process is uninteresting, so beginners can easily give up. We will make the process of learning words freer by allowing players to use object detection, where the camera takes pictures for objects that they find interesting. These objects are then transformed into word cards. Afterwards, players can use these cards to play games and memorize words at the same time.

German grammar is renowned for its complexity, particularly evident in the categorization of articles into masculine, neuter, and feminine genders. Moreover, the German language encompasses the Nominativ, Akkusativ, Dativ, and Genitiv cases, necessitating alterations to noun suffixes as speakers articulate their thoughts. This intricate system underscores the importance of meticulous attention to detail in language usage, as the correct application of gender and case is paramount for effective communication in German. Mastery of these grammatical nuances not only facilitates clarity in expression but also demonstrates

a profound respect for the linguistic heritage and cultural richness encapsulated within the German language.

All of the above mentioned grammar is often hard to understand for beginners, and overwhelms new speakers regularly. In our serious game, we have focused on increasing the player's German vocabulary and memorization of grammar. At the same time, we have taken care to make the game as fun as possible, so that the player can learn German without it feeling like a chore. Therefore our game is suitable to assist beginners in learning German.

### 3.4. Operational Analysis

This game was developed with educational and recreational purposes in mind, so it is particularly easy to use. [20] The operation is also particularly simple and can be learned by a new-hand in less than a few minutes. Additionally, we have added an assisted mode for the game. During the game, the players can turn on the assisted mode and just get the best answer at any time.

We have kept the game to a reasonable 10-15 minutes to ensure that the users will not feel boring during the learning process and keep concentrated. [21] Therefore it is operationally feasible.

### 3.5. Other Analyses

Our developed serious game is free and only used for teaching purposes. All the contained material models, videos and audio whose copyrights have been purchased or certified. They are not intended for use in commercial games or applications. The project will not involve private information or access to permissions. The entire game is designed to comply with the law. [17]

At the same time, the game is fully compliant with ethical standards, does not contain socially negative content, such as horror or pornography. It is suitable for players of all ages. Therefore it is legally feasible.

The entire game system, including the scenes and game objects etc., is set up in Unity. The scripts required for the entire design are created in Visual Studio and JetBrains Rider. ChatGPT (GPT-4) can be downloaded and used free of charge through the corresponding official website with the Education certificate provided by the university, TUM. The key for ChatGPT (GPT-4) was provided by the tutor, Mengdi Wang.

In the standards of open source sharing and university education, the game does not have any paid features, so users can play and modify it as much as they want. Therefore it is economically viable.

### 3.6. Numerical Planning

Numerical planning affects the logic, rationality, immersion, and duration of the game. A game's entertainment depends on its logic and rules. The quantitative representation of the rules is the numerical values. In addition, standardizing game numerical value can not only reduce the cost of research and development, but also reduce the cost of operation. How to use numbers to drive all kinds of systems to achieve the preset purpose is the biggest role of numerical values.

The planning method of numerical value contains many kinds, most of the numerical values are calculated by applying some formulas, so as to ensure the fairness of the game, i.e. game balance. In order to ensure its balance, we gradually set the numerical values in the process of developing the game, as well as constantly debugging and changing them over and over again.

After systematic design, we standardize the names of the values of all the modules included in the game, and determine the size of their values.

### 3.7. Functional Testing

Since the whole development process is complex and has many functions, it is possible that some problems will occur. We can check whether the interface of each functional module of the system meets the design requirements through functional testing. If there are problems, we can solve them before deadline, which not only ensures the robustness and stability of the game, but also ensures the experience and immersion of the players.

After the initial completion of each function of the game, we start to run the game in parallel, testing whether each module is running normally, to ensure that the game experience meets the requirements of the requirements analysis. We must provide the guarantee that different plays can play the game smoothly, including real-time interaction between real scenes and game scenes, cascade connection of different devices, identification of real objects, placement of game cards, generation and stacking of Buff, detection of foreign language syntax, adding and subtracting of life values and scores, etc.

### 3.8. Expected Results

Through the project, the following results are expected to be obtained:

- 1) To master the basic knowledge and usage of the game engine Unity3D, and be familiar with the operation of game development tools, scene design, object control, animation production, and UI interface design.
- 2) To design and realize a serious game with a certain degree of usability and interest, which can show the players' creativity and ability.
- 3) To improve our programming skill level, and at the same time, we can get some fun and satisfaction.

### 3.9. Key Difficulties

Several challenges arose during the completion of the project. The most impactful are listed here.

**3.9.1. Object Recognition.** In the beginning, this was the riskiest aspect of the project, as it was unclear whether existing recognition models will satisfy our needs. We identified three possibilities:

- 1) Proposal A: Use the "GPT-4 with Vision" API by OpenAI. [22]

- 2) Proposal B: Use YOLO (state of the art deep learning models) using an implementation in Unity. [23], [24]
- 3) Proposal C: Use the perception tools provided by the Unity game engine. [25], [26]

We first tested out proposal A, because while an OpenAI API-key is required, it is in theory easy to use and integrate. In practice, we encountered issues with making the model deterministic and having it output the required information reliably (see 3.9.4), but due to the overall success of using this method, we stuck with it.

**3.9.2. Plane Detection.** Plane detection refers to the ability of AR systems to recognize flat surfaces in the physical environment, such as floors, tables, and walls, and generate virtual planes that align with these surfaces. This capability allows developers to anchor virtual objects onto real-world surfaces, creating realistic and immersive AR experiences. We decided to use Unity's AR Foundation tool-set for this purpose, as it is a reliable and established method for plane detection.

**3.9.3. Grammar data.** In order to generate grammar cards that have the correct grammar of the object cards on them (and for validating played cards), it is necessary to know all declensions of all object cards in the game. Our first approach to achieve this involved simply asking the same GPT-4 vision endpoint used for the object detection, but it turned out to be unreliable and, to our surprise, often incorrect.

In most cases, for any object detected by the GPT-4 vision endpoint, the DE Wiktionary - a free and open german dictionary [27] - will contain a definition of the word together with explanations, sample usage and - most important for us - a complete declension table. We decided to leverage the MediaWiki API [28] in order to access those pages and extract the relevant content. This approach proved to be very robust, and in the case of a missing entry we are still able to rely on GPT-4 to provide a mostly correct answer.

**3.9.4. Prompt Engineering.** One of the most significant challenges we encountered was prompt engineering; creating a prompt for the GPT-4 vision endpoint so it would not only return the information we want, but also do so in the exact format we specified. Because contrary to most usages of generative AI, we directly process the output programmatically. We intended to get four pieces of information from the AI:

- **The (german) name of the object** Getting the name of the object was not too hard, but GPT-4 tended towards using rather outdated or complicated names, which needed to be addressed.
- **The attack stat of the object** We told the AI to base this on the size of the object, compared to an average item found commonly on a desk. The value is to be constrained between 1 and 5 (both inclusive).
- **The defense stat of the object** This stat is supposed to be based on the perceived toughness/robustness of the object and constrained between 10 and 20 (both inclusive).

- **The corresponding declension table (initially)** As already mentioned, this did not provide reliable results, so we used the approach detailed in 3.9.4.

We tested a multitude of prompts, and in the end settled on a English-German hybrid, as it provided the best results in both labelling the object with an accurate German word and choosing sensible stat values. Getting the AI to use the full range of the possible stat values with an appropriate variance proved to be much more challenging than expected; the AI does not possess information about previous requests, so it keeps assigning the same values. We were not able to completely mitigate this behaviour and only managed to suppress it slightly.

## 4. Overall Functional Design

DEUTSCH BRAWL is an AR mobile card game designed to help beginners learn German grammar and vocabulary. The basic principle is that 2 players (each with their own phone) can duel each other and gain advantages through the correct use of their knowledge of German vocabulary and grammar. Each game is split into two phases.

The main game mechanic is based around assigning given grammar changes in certain grammatical cases to the suitable objects. This is rewarded by granting the object card one of several special effects. The players will have to use their knowledge of German grammar concepts like articles, suffixes, genders, plurals, etc. and as a result, the players will learn proper grammar through their pursuit of victory.

### 4.1. Phase 1 - Card Collection

In this phase, the players first connect to the same virtual room. To do this, one player will create a room by selecting an arbitrary room name and using the "Join as host" button. The other player must enter the same room name and select "Join as client".

After the players connect to the same room, they then define a playing area (using plane detection provided by AR Foundation) on which the battlefield will be placed. This battlefield will serve as the main playing area for phase 2.

After this is done, the players can look around the room they are in and take snapshots of objects in the real world. Once a picture is taken, the OpenAI vision endpoint [22] based on GPT-4 is used to detect the main object in the picture. The picture is transformed into an object card.

The name of the main object in German is the card title. The card is also assigned both a health and an attack stat, depending on certain characteristics of the object in the real world.

The card also has a table with entries for the singular and the plural of every grammatical case the object can be used in. In the beginning, this table is empty, but during the second phase the entries can be filled, which each filled entry providing some kind of buff or special ability.

The player can decide whether to keep the object card, and if they do, it is available to be played in the second phase. Alternatively, the object card can be discarded, if the player is not satisfied with the result.

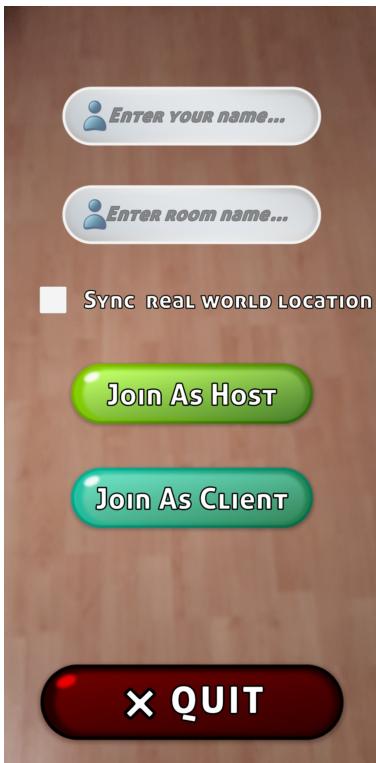


Figure 1: Start screen seen upon opening the game



Figure 3: An example object card

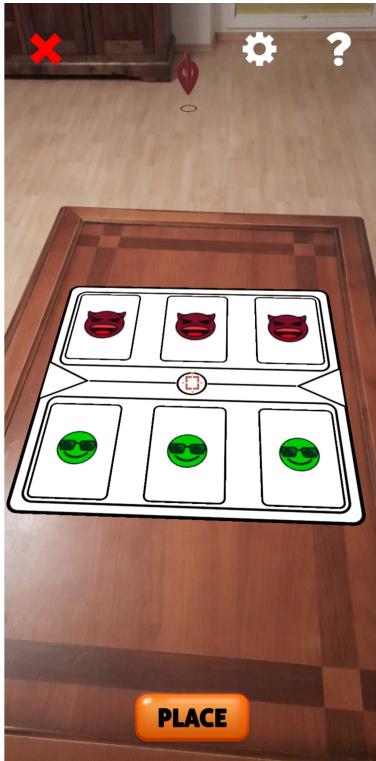


Figure 2: Placing the game board. Markers like in the background denote possible placements

This phase ends once the player has collected three cards, and they will be shown a waiting screen where they can review their cards. Once the second player also picked enough cards, the second phase starts.



Figure 4: The waiting screen shown if one player is done before the other

## 4.2. Phase 2 - Game Loop

This phase represents the core game loop. Here, players can play the previously collected cards. The cards will be placed on a grid-like battlefield where they fight against each other until they are all destroyed or one player runs out of health (that is, bottles). But first the player is introduced to the game, and given an explanation of the basic concepts of the game.

**4.2.1. The battlefield.** The gameplay will take place on the previously positioned battlefield. It has two rows of slots - one for each player - where (only) object cards can be placed. To the left of the battlefield several bottles are placed, with the goal of the game being to steal all of the bottles from your opponent. On the right, a traffic light displays whether you are currently able to play. This is determined by the current state of the game.



Figure 5: The tutorial screen the player encounters first in the second phase

**4.2.2. Rounds.** The game is divided into rounds, with three parts to each round:

- 1) Player turn
- 2) Opponent turn
- 3) Attack phase

The order of the player/opponent turns is changed every round, so no player has an unfair advantage.

In their respective turn, the players get two actions, each of which they can spend to either draw a card, or play a card. It is only possible to draw grammar cards as the players do not get any further object cards beyond the initial ones. The players must exercise haste, as their turn duration is limited. The aforementioned traffic light tells

the players when they are able to play: if the light is green, they can perform their actions, when it blinks yellow their turn is about to end and if it is red, they cannot perform any more actions. The number of actions left can also be seen on the center of the battlefield as shown in 6.

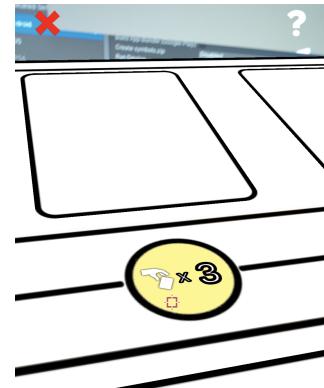


Figure 6: Number of actions left

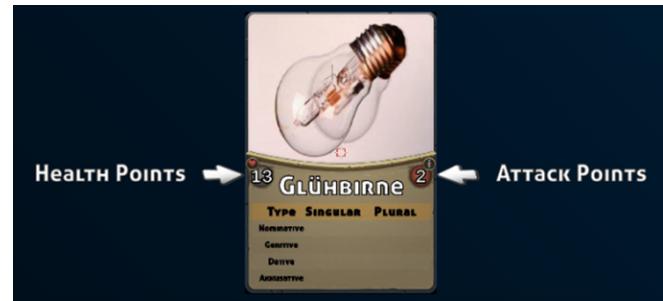


Figure 7: An object card depicting a light bulb

**4.2.3. Selecting and placing cards.** During your turn, you have the opportunity to play a card by tapping on it once. Then, while aiming at the desired slot on the battlefield with your crosshair, swipe the card towards the battlefield. It's important to note that object cards can only be placed on empty slots, while grammar cards can only be applied to already-played object cards.



Figure 8: Placing an object card on the battlefield

Once a card is placed on the battlefield, it remains in its position and cannot be moved unless affected by the "swap" effect. This rule ensures that the strategic placement of cards is carefully considered and that players must think ahead to optimize their gameplay. By adhering

to these guidelines, players can engage in a structured and strategic card-playing experience that rewards thoughtful decision-making and planning.

**4.2.4. The attack phase.** Once both players have completed their turns, the attack phase commences. In this phase, all object cards positioned on the battlefield automatically initiate attacks against the card directly in front of them, unless obstructed by a specific effect.

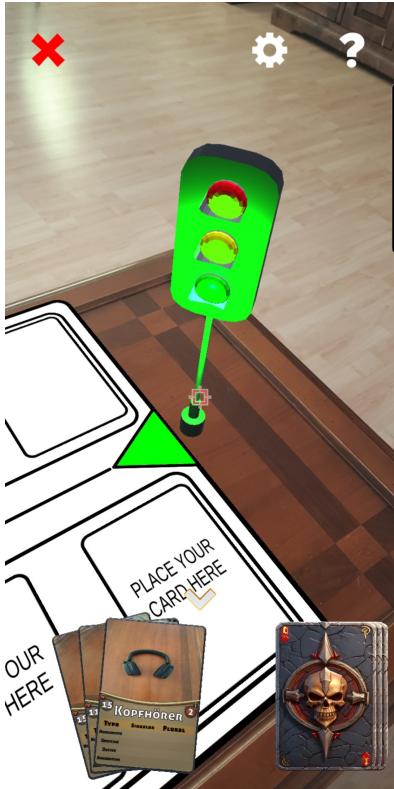


Figure 9: The traffic light indicating whose turn it is

Each card deals damage based on its attack stat and sustains damage equal to the attack stat of its opponent. It's crucial to note that cards do not regain health during this phase. If a card's health reaches zero, it is promptly destroyed, and any excess damage is transferred to the respective player. In instances where a card lacks an opposing object card, it directly inflicts damage upon the player instead, thereby stealing a quantity of bottles equal to its attack stat.

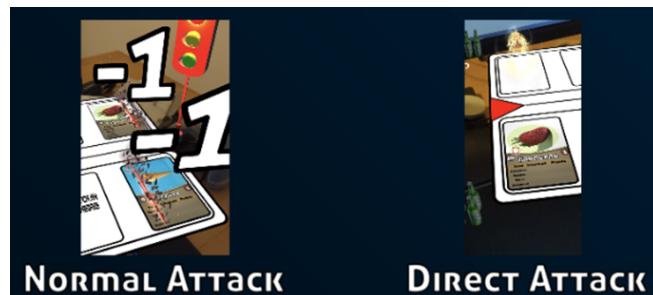


Figure 10: The two possible situations when attacking

**4.2.5. Grammar cards.** The cards drawn from the deck are exclusively grammar cards. These behave entirely

differently to object cards. Every grammar card consists of two things: the grammar, and the effect.



Figure 11: Structure of a grammar card

The grammar is what the players need to match to the object card (by playing the grammar card while highlighting the targeted object card). To make this non-trivial, only the changes that grammatical case necessitates are displayed, so a grammar card displaying "die -e" could be matched to the object card "der Tisch", as the nominative plural of "der Tisch" is "die Tische". To further restrict the potential applications of a specific grammar card, the intended grammatical case is displayed and required to match. So if the grammar card from before also displayed "genitive plural", it could not be matched to "der Tisch" as the genitive plural of "der Tisch" is "der Tische", not "die Tische".

Players will only draw grammar cards that can be applied to at least one object card on the battlefield that is theirs, so as to not cause frustration. The special effect of the grammar card is chosen randomly and not tied to the grammar itself.



Figure 12: A grammar card with a genitive singular grammar and an attack buff effect

If the player manages to match the grammar card correctly to an object card, the aforementioned effect comes into play. This effect is randomly chosen from a pool of six different effects ("the card" designates the card the effect is applied to):

- **Attack buff** Increases the attack stat of the card.
- **Health buff** Increases the defense stat of the card.
- **Heal buff** Heals the card back up to its maximum health (the defense stat).
- **Freeze turn** At the start of the next attack phase, the card will apply a "Frozen" effect to its direct opponent card, rendering it unable to attack in that round.
- **Mutual suicide** In the next attack phase, instead of attacking, the card will perform a ritual causing both itself and its direct opponent to be destroyed, independent of attack stats or remaining health. The only way this can be countered is by applying a "freeze turn" effect to its opponent, rendering this card unable to target the other card during the ritual and subsequently only destroying itself.
- **Swap** Allows the player to swap this card with any other card on the battlefield.

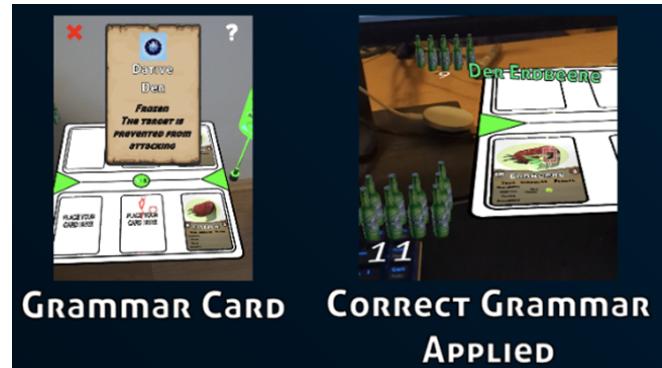


Figure 13: A grammar card used correctly



Figure 14: A wrong grammar card used

To allow more strategical play, the effect associated with a particular grammar card is known to the player immediately after drawing the grammar card.

In case the player chooses a wrong object card, a wrong text effect is shown and the object cards which fit the grammar are highlighted. These object cards may include cards of the opponent as well to boost the learning aspect.

**4.2.6. Set effects.** Once an entire column of an object card's grammar table is filled, it is granted a special set effect (more health for the singular column and more attack for the plural column). Upon completing the entire grammar table, the card is granted even more health and attack, making it a behemoth on the battlefield that has to be carefully considered when devising a strategy.

### 4.3. Accessibility options

There are two accessibility options that can be toggled on or off through the settings. They are both disabled by default in order to fit best with the intended experience.

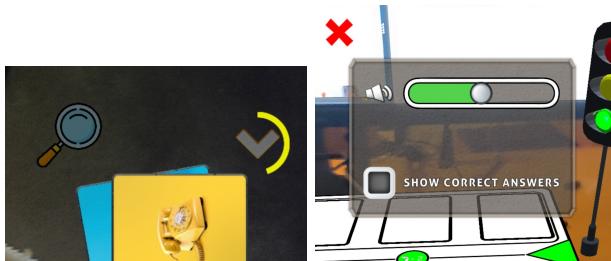


Figure 15: Accessibility options

**4.3.1. Card near view.** While any object on the battlefield and in particular placed object cards can be inspected from arbitrarily close, we recognize that this might not be accessible, practical, or outright possible in some situations. By enabling the magnifying icon as shown in 15, any card the crosshair is hovering over will get displayed in an enlarged view, so physically moving closer is not necessary. The results are shown in 16.

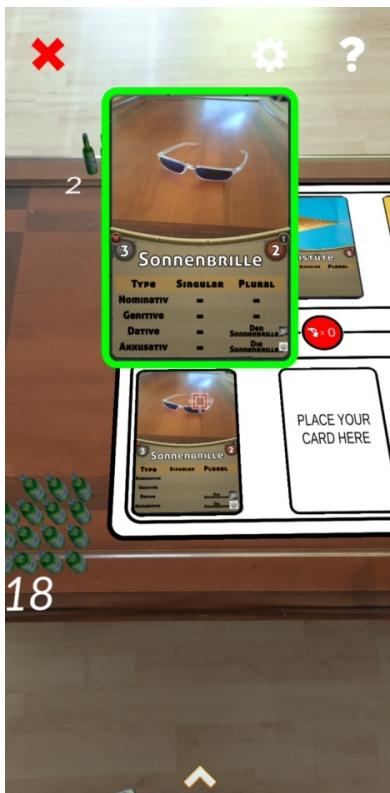


Figure 16: A closeup of an object card enabled by the "card near view" option



Figure 17: Correct answer card gets highlighted before placing a grammar card

**4.3.2. Show correct answers.** Players are required to match grammar changes in certain grammatical cases with the corresponding object cards, but if a complete beginner plays the game, they might not be able to correctly spot these connections. This can cause frustration and prevent any learning effect from happening. To counteract this, the option "show correct answers" can be enabled as shown in 15, causing "correct" object cards to be highlighted when selecting a grammar card as shown in 17. As this option removes a significant portion of the learning aspect and leaves the strategical usage of cards as the sole skill-based component of the game it is suggested to use this option sparingly and only when absolutely necessary.

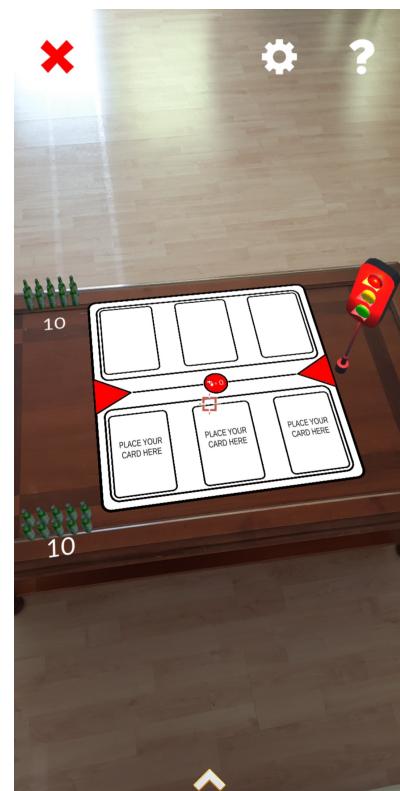


Figure 18: The empty battlefield at the start of phase two

## 5. User Study

We conducted a user study to investigate the effectiveness of Deutsch Brawl in facilitating language learning, specifically focusing on the German language. The study involves evaluating participants' language proficiency levels before playing the game and the participant's learning outcome after playing, as well as collecting feedback on the game's usability and educational value. The study seeks to determine whether the game serves as an effective tool for language acquisition and comprehension, thus contributing insights into the potential of AR technology in language education.

### 5.1. Participants

Participants aged 18 to 30 were recruited for this study due to their familiarity with technology and potential interest in gaming experiences. Recruitment efforts targeted university campuses, and social media platforms. The sample aimed to include individuals with varied levels of German language proficiency, from beginners to intermediate learners, to assess the game's efficacy across different skill levels. Ethical guidelines were strictly followed, ensuring informed consent, confidentiality, and voluntary participation.

### 5.2. Experiment Method

#### 5.2.1. Hypothesis. *Deutsch Brawl helps users to learn German*

We hypothesize that engagement with the game will lead to measurable improvements in participants' language proficiency levels. Through interactive gameplay involving vocabulary acquisition, grammar application, and contextual understanding, we expect users to experience enhanced language comprehension and retention. This hypothesis is based on the premise that gamified learning experiences, coupled with AR technology's immersive nature, can effectively stimulate cognitive processes and facilitate language acquisition in an engaging and enjoyable manner.

- **Null Hypothesis (H0):** There is no relationship between the game and learning German.
- **Alternative Hypothesis (H1):** The game can help users learn German vocabulary and grammar.
- **Alternative Hypothesis (H2):** Besides passive learning, this game can motivate users to learn German actively in addition to their gaming time.

#### 5.2.2. Steps. The experiment procedure has 3 main steps:

- **Introduction and Consent:**

The experiment commences with an introduction to the AR card game, where participants are briefed on its objectives and gameplay mechanics. This includes instructions on how to interact with the cards and navigate the AR environment. Following this introduction, participants are asked to provide informed consent before proceeding to

the gameplay session.

- **Gameplay Session:**

During the structured gameplay session, participants engage in vocabulary and grammar learning tasks facilitated by the game. These tasks involve matching object cards with corresponding grammar articles.

- **Evaluation:**

Finally, after the gameplay session concludes, participants are asked to fill out a questionnaire. This questionnaire is designed to gather feedback on their experience with the game, covering aspects such as overall enjoyment, perceived effectiveness in learning German language concepts, and suggestions for improvement. This concise and systematic procedure ensures a thorough evaluation of the AR card game's efficacy as a language learning tool, while also prioritizing participant engagement and ethical considerations.

#### 5.2.3. Variables. This study adopts a within-subjects experimental design.

The independent variables are as follows:

- Prior German proficiency
- Prior English proficiency
- Prior Experience with AR Apps/Games

The dependent variables are as follows:

- Effectiveness of Learning Vocabulary
- Impact on Daily Language Use

#### 5.2.4. Questionnaire and Feedback. We designed a questionnaire in order to get meaningful feedback from the participants. The questions are as follows:

- **German Proficiency Level:** This question aims to gauge participants' familiarity with the German language, helping to assess their baseline language skills before engaging with the AR card game.
- **English Proficiency Level:** Understanding participants' proficiency in English provides context for interpreting their feedback and responses, as the questionnaire is administered in English.
- **Gaming Experience:** By identifying participants' gaming background and experience level, this question helps in analyzing how gaming habits may influence their interaction with the AR card game.
- **Types of Games Played:** This question explores participants' gaming preferences, offering insights into the types of gameplay experiences they find engaging and enjoyable.
- **Gaming Platforms Used:** Understanding which gaming platforms participants prefer provides valuable context for assessing their familiarity with different gaming environments, which may impact their experience with the AR card game.

- Prior Experience with AR Apps/Games:** This question assesses participants' familiarity with augmented reality technology, helping to understand their comfort level with AR experiences.
- Effectiveness of Learning Vocabulary:** This question evaluates whether participants perceived an increase in their German vocabulary after engaging with the AR card game, providing insights into the game's language learning potential.
- Ease of Use:** Assessing the ease of use of the game helps identify potential usability issues and areas for improvement in the game's interface and interaction design.
- Understanding of Game Flow:** This question evaluates participants' comprehension of the game's mechanics and progression, indicating whether the game's instructions and objectives were clear.
- Accuracy in Matching Articles:** By assessing participants' accuracy in matching articles with nouns, this question measures their grasp of German grammar concepts introduced in the game.
- Recollection of German Words:** This question gauges participants' ability to recall German words learned during gameplay, indicating the game's effectiveness in facilitating language retention.
- Recollection of Articles:** Assessing participants' ability to recall the articles used with specific nouns provides insights into the game's impact on their understanding of German grammar rules.
- Perception of Learning Benefits:** This question explores participants' beliefs regarding the game's potential to aid in vocabulary and grammar retention over time, informing future iterations of the game's design.
- Engagement with Gameplay:** Participants' perception of the game's entertainment value offers insights into its overall appeal and potential for sustained engagement.
- Impact on Daily Language Use:** This question assesses participants' expectations regarding the game's influence on their daily German language usage, providing insights into its practical relevance and long-term impact.

What is your german level?

 A1  
 A2  
 B1  
 B2  
 C1  
 C2  
  

What is your english level?

 A1  
 A2  
 B1  
 B2  
 C1  
 C2

Figure 19: Questions to get demographic information

At any point did you feel it was not a game but rather a book/lecture session?

 Yes  
 No  
  

How fun or exciting was the game overall?

1 2 3 4 5

Not fun/exciting at all                    Very fun/exciting

Do you think the vocabulary and articles learnt from the game will help/improve your daily German?

 Yes  
 No

Figure 20: Questions to evaluate the game

We go through an iterative development process using the feedback. 19 shows the questions that capture demographic information and 20 shows questions that evaluate the game. 21 and 22 show some of the responses.

[Link to the questionnaire](#)

[Link to the responses](#)

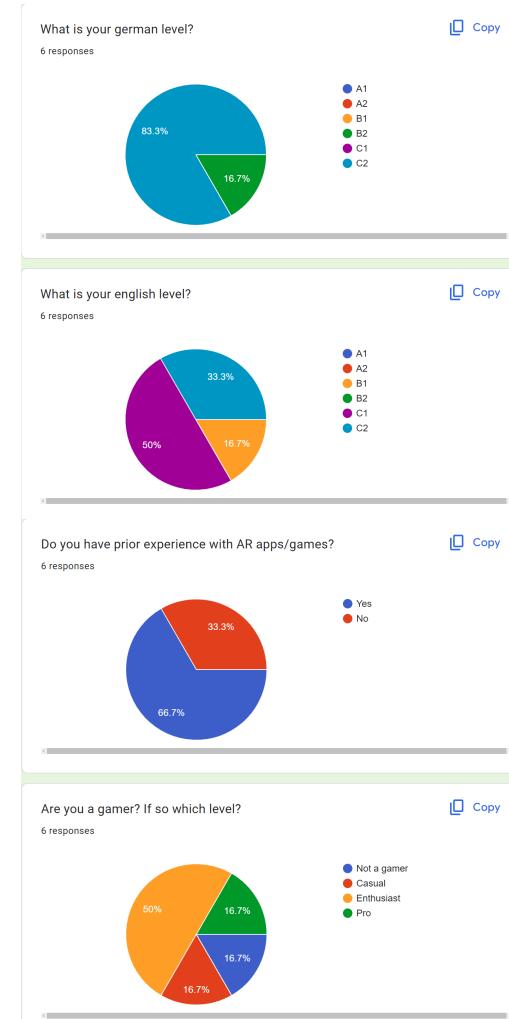


Figure 21: Responses

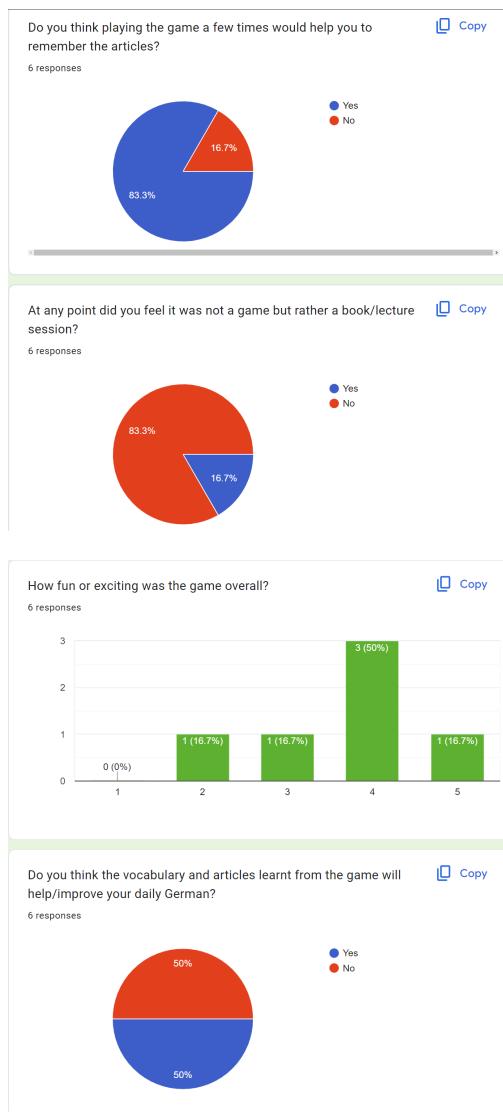


Figure 22: Responses

**5.2.5. Analysis.** To test the hypothesis that the game helps players learn German, we would use inferential statistical analysis. Specifically, we could use ANOVA.

## 6. Project Record

### 6.1. Report Authorship

The authorship of the report can be attributed as follows:

- Ankur Deria: 1.2, 1.3, 1.4, 2.5, 2.6, 2.7, 3.3, 4.2.3, 4.2.4, 5 - 5.2.5.
- Maximilian Mitterrutzner: 3.9 - 4.2.2, 4.2.5 - 4.3.2, 6.1.
- Yiyang Xie: Abstract, 1 - 1.2, 1.5 - 3.2, 3.4 - 3.8, 4.5.2 - 4.6, 6.2 - 7.

The following is our work schedule and specific weekly tasks:

### 6.2. Preparatory Phase - Oct 2023

- \* Understand math, physics concepts used by Unity, such as 3D Euclidean space, Translations, Rotations, Perspective projection.
- \* Understand the concepts of Serious Games, AR, MR, VR, Game Engine, etc.
- \* Learn the basics of Unity including Material, Camera, InstantOC, SkyBox & Lighting System, Sound, Rendering Pipeline, Prefab, Animation, Particle Effects.
- \* Learning the basics of C# and writing scripts.
- \* Implementing Stupid Shooter Game.

### 6.3. Phase I - Nov 2023

- \* Familiarize ourselves with VR headsets, Basics & Setup Varjo XR.
- \* Understand Interaction Design concepts.
- \* Attempt to implement a VR version of Stupid Shooter.
- \* Confirmation of serious games for group and project design.
- \* Identify the basics of what needs to be accomplished in a serious game.
- \* Search for relevant literature and materials.
- \* Learn about the principles of AR technology and familiarize ourselves with the way for object recognition.
- \* Brainstorm, propose a draft design for the project, and write a proposal.
- \* Determine the realization direction, game framework and game logic.

### 6.4. Phase II - Dec 2023

- \* Build the basic scenarios for the serious games in the project, create cards, card slots and other related prefabs.
- \* Complete the basic object recognition function and card conversion function.
- \* Realize the demo, implement the basic logic of the game.
- \* Add other functions that increase the fun of the game.
- \* Test the game, then improve design and redesign.
- \* Find problems in the development process, record the problems, and communicate with the tutor through the weekly meeting.
- \* Modify and improve the expected functions.

### 6.5. Phase III - JAN, FEB 2024

- \* Improve design and redesign.
- \* Enter final software refinement phase, consider basic interaction design for serious games.
- \* Personalize the model design based on the serious game.
- \* Add special effects and sound effects to improve the initial beauty of the game.
- \* Write reports, prepare presentations, and confirm file formats.
- \* Test game and the final debugging to minimize bugs in the game.

## 6.6. Weekly Records of the Project

### \* 1-3. Week (20.10 - 8.11):

- \* Introduction and courses of Praktikum: Serious Games in Extended Reality.
- \* Introduction to Serious Games and Extended Reality.
- \* Unity & Varjo Tutorial.
- \* Introduction to Usability, User Experience and User Studies.

### \* 4. Week (9.11 - 15.11):

- \* Communicate with tutor Mengdi Wang to learn how to turn 3D games into VR games.
- \* Brainstorm in a team of three and come up with 4 draft designs.

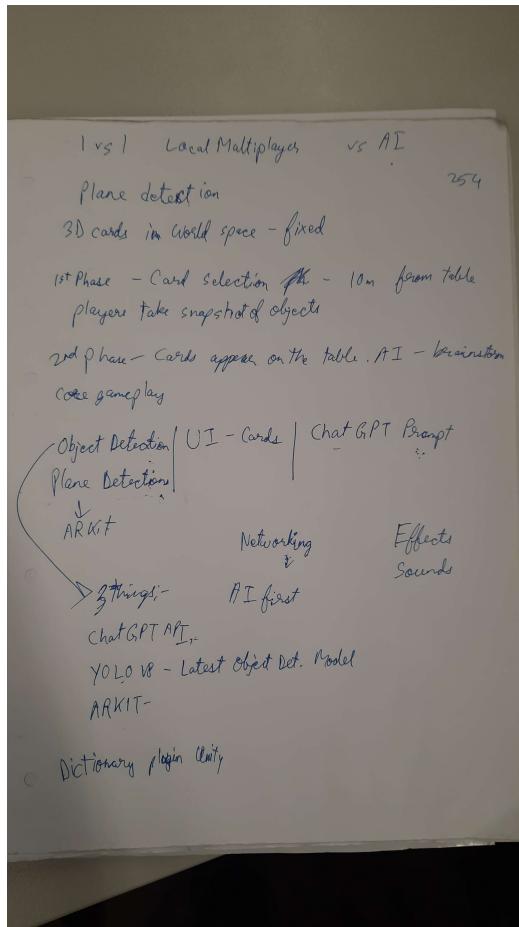


Figure 23: draft

### \* 5. Week (16.11 - 22.11):

- \* Communicate with tutor Mengdi Wang to determine the design theme and initial development of the project.
- \* Write a proposal, schedule for the following weeks (milestones, prototypes, features, ...).
- \* Create Unity project, GitLab project repository, Google Docs, Milanote, Trello.
- \* Build basic scenarios for a serious game in the project, create cards, slots, and other prefabs.

- \* Learn Object Recognition, Plane Detection, Prompt Engineering.
- \* Determine the name of our serious game, Deutsch Brawl.

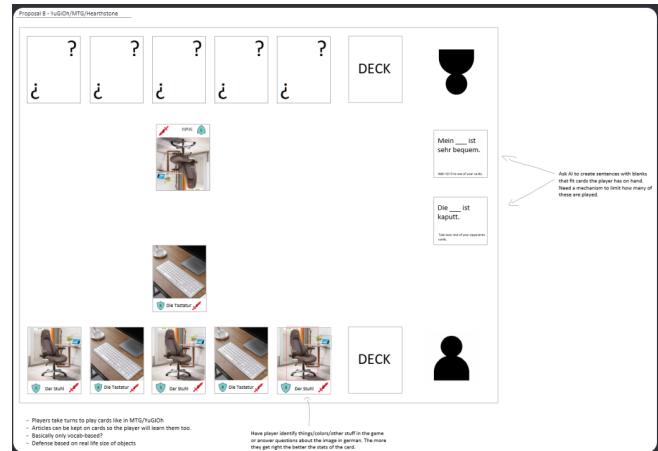


Figure 24: game logic

### \* 6. Week (23.11 - 29.11):

- \* Learn to build interactive UI interfaces, UI mockups.
- \* Download Material for menus.
- \* Start learning Unity + Vuforia + AR development
- \* Local multiplayer on the same network in mobile (Add Photon).
- \* Test using ChatGPT for Object Recognition in Unity.
- \* Ask Mengdi for an API Key for ChatGPT, Use API to send pictures to ChatGPT 4 and get it to identify objects in the image.

### \* 7. Week (30.11 - 6.12):

- \* Local multiplayer on the same network in mobile (Add in editor testing setup).
- \* Test using ChatGPT for object detection in Unity (Integrate functionality in a Unity Project, Test build on phone).
- \* Design the UI for joining game, UI mockup.
- \* To ensure that the development hardware device is normal, connect Android Studio to the phone, open USB debugging and enable developer mode. build and install the APP, the use of the effect is normal.

### \* 8. Week (7.12 - 13.12):

- \* Setup basic connection and info exchange (Local multiplayer on the same network in mobile).
- \* Test using ChatGPT for object detection in Unity (Test using ChatGPT to crop images, Test using the API to crop images).
- \* Design the UI for selecting cards, UI mockup.
- \* Write basic game logic.
- \* Add buff functionality to tabletop game cards.

- \* Integrate all the UI we made into 1 scene together.
- \* **9. Week (14.12 - 20.12):**
  - \* Setup basic connection and info exchange (Local multiplayer on the same network in mobile).
  - \* Test using ChatGPT for object detection in Unity (Test reducing waiting time).
  - \* Download the Material for the particle effects needed for the buff cards.
  - \* Test multiplayer and crosstalk between different devices.
  - \* Design the battle board, the card slots and object cards.
  - \* Write basic game logic.
  - \* Import the extension for grammar testing in German.
  - \* The icon gives the option of keeping or removing the object cards.
- \* **10-12. Week (21.12.2023 - 10.1.2024):**
  - \* Simplify the game to make it easier to use.
  - \* Use different colors/effects to highlight card slots.
  - \* Add a crosshair in the center of the screen and thus easier to play (set cards, set buff, etc.).
  - \* Beautify the game's UI to make the game look more flashy.
  - \* Adjust the settings in the particle system of the effect prefabs.
  - \* Show life value with the bottles.
  - \* Show the remaining time of a turn with a traffic light.
  - \* Redesigned the Materials (e.g. crosshairs, bottles) needed for the above features.
  - \* Download Material for sound effects for buff cards.
  - \* Add more buffs to increase game playability.
  - \* Improve the grammar detection bug for foreign languages (ChatGPT doesn't seem to answer grammar questions correctly and reliably).
  - \* Improve the game to make the game smoothly, especially the process for the object recognition.
- \* **13. Week (11.1 - 17.1):**
  - \* Learn to add sound effects and write the AudioManager scripts.
  - \* Add the sound effects/files to the audio manager in Unity.
  - \* Test the game, using a cell phone and a computer as two players against each other.
  - \* Redesign for the 2. version.
  - \* Adjust game values and game duration more reasonable.
  - \* Improved some features, drag and drop to reposition models.
  - \* Choose whether to synchronize the real world position or not, if the function is

enabled and detected, it will show the indicator icon.

\* **14. Week (18.1 - 24.1):**

- \* Test the game to try to minimize network interruptions or bugs that occur with Android.
- \* Learn a bit about the existence of prefabs with child objects and parent objects, etc.
- \* Create a new prefab after changing a prefab that satisfies a state.
- \* Write a summary of the Report, development tools and environment, development analysis, summarization.
- \* Prepare for DemoDay?
- \* Adjust sound effects.

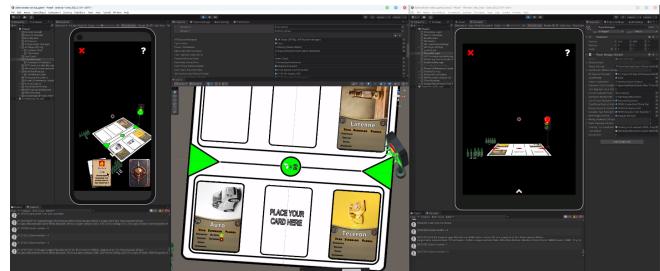


Figure 25: designing process of the 2. version



Figure 26: the tutorial for the game

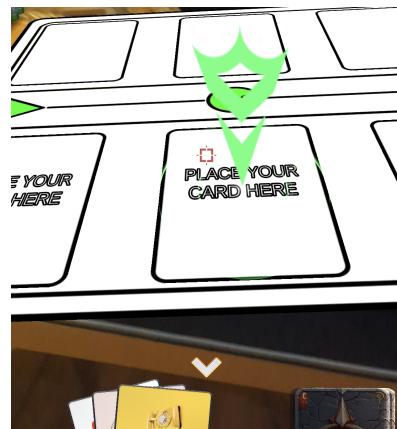


Figure 27: 2. version for swapping the cards

\* **15. Week (25.1 - 31.1):**

- \* Write report: overall project functional design, design and implementation, values

- and functions, project design completion record.
- \* Design the presentation.
- \* Test the game. Add the tutorial to the game.
- \* Prepare for DemoDay?
- \* Design the questionnaires.
- \* Collect the comments from the players after playing our serious game.
- \* Solve the bug for skipping the attack phases.
- \* Improve the Freeze and Frozen effects.



Figure 28: 2. version of the game



Figure 29: the online questionnaire

- \* **16. Week (1.2 - 7.2):**
- \* Offline meetings in the practicum lab, preparation for presentations.
- \* Complete the report, self-checking, and confirm the format of the document.
- \* Test the game and the final debug, solve the bug of mutual suicide.
- \* Improve the Freeze and Frozen effects.

## 7. Conclusion

Throughout the project, we have implemented a serious game with the name of Deutsch Brawl. The entire game for the practicum has been set up in the game engine Unity3D. The related scripts for the entire design have been created in Visual Studio and JetBrains Rider. The programming language used is C#. The AI extensions, such as ChatGPT 4, are used for object recognition, plane

detection and German grammar testing. The developed game is completed through requirements analysis, general design and detail design. After that the game system testing is completed. In the end of the project, we have developed a serious game that recognises 3D objects in real time, forms board game cards through picture and natural language conversion, and finally combines German grammar to realize attack and defense. The game UI is simple, easy to understand, interesting and interactive.

The overall project design is summarized below:

- 1) Ensure the playability and functionality of the game through software requirement analysis.
- 2) Ensure the rationality of the game through the framework design and process design of the game. Then gradually script and add components to each module.
- 3) After the completion of each module of the game, the game was tested to ensure the normal operation of the game and to meet the requirements of the project for serious games.
- 4) Develop the 2. version and beauty the game UI, in order to attract more players and increase their interests for learning German.
- 5) Add particle effects, sound management, and other functions to increase the fun of the serious game.

Eventually, we have designed a serious game that can fully realize the real-time interaction between the real scene and the game scene, the cross talk of different devices, the recognition of real objects, the placement of game cards, the generation and superposition of Buff, the detection of foreign language grammar, the addition and subtraction of life values and scores, etc. The overall AR game has a strong sense of immersion, the game is highly playable and easy to understand. The duration of a single game is reasonable. The learning of a foreign language (German) through the game satisfies the requirements of serious games. In conclusion, we have followed the iterative design process for one software and achieved the expected project requirements.

During this project, we have encountered various problems and many failures. However, the difficulties were solved through the Discord discussion within the group, our review of videos, documents, websites and repeated attempts, two regular meetings per week (once with the tutor and once without). We are sincerely grateful for the help of our mentors Mengdi Wang and Süleyman Özdel. The whole project is full of difficulties, but also the fun of innovation and breakthrough.

## References

- [1] BCG, "Game Changer: Accelerating the Media Industry's Most Dynamic Sector," <https://www.bcg.com/publications/2023/drivers-of-global-gaming-industry-growth>, 2023, [Online; accessed 23-Jan-2024].
- [2] B. Marr, "Game On! The Top 10 Video Game Trends In 2024," <https://www.linkedin.com/pulse/game-top-10-video-trends-2024-bernard-marr/>, 2023, [Online; accessed 23-Jan-2024].
- [3] "Unity CEO says half of all games are built on Unity," <https://forum.unity.com/threads/unity-ceo-says-half-of-all-games-are-built-on-unity.551857/>, 2018, [Online; accessed 23-Jan-2024].

- [4] A. Dave, "Why is Unity your Best Bet for Game Development in 2024," <https://300mind.studio/blog/unity-for-game-development/>, 2022, [Online; accessed 23-Jan-2024].
- [5] S. Mukherjee, "Games as a Service: Everything You Need to Know in 2024," <https://clevertap.com/blog/games-as-a-service/>, 2024, [Online; accessed 23-Jan-2024].
- [6] D. M. Romano, "Serious Games: An Introduction," <https://www.slideshare.net/DanielaRomano1/4-serious-games-an-introduction>, 2014, [Online; accessed 23-Jan-2024].
- [7] K. D. A., "Experiential learning: Experience as the source of learning and development," *FT press*, 2014.
- [8] C. Yoshimura A., Borst, "A Study of Class Meetings in VR: Student Experiences of Attending Lectures and of Giving a Project Presentation," *Frontiers in Virtual Reality*, p. 2.10.3389/fvrir.2021.648619., 2021.
- [9] A. S. B. R. d. F. S. K. K. D. G. A. Bellotti F., Ott M., "Designing serious games for education: from pedagogical principles to game mechanisms," *Proceedings of the 5th European Conference on Games Based Learning*, pp. 26–34, 2011.
- [10] M. C. Fleming N. D., "Not another inventory, rather a catalyst for reflection," *To improve the academy 11(1)*, pp. 137–155, 1992.
- [11] Microsoft, "Introduction to C," <https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/tutorials/>, 2022, [Online; accessed 23-Jan-2024].
- [12] Unity, "Get started with Unity," <https://unity.com/learn/get-started>, 2024, [Online; accessed 23-Jan-2024].
- [13] Microsoft, "Getting started with Visual Studio - Microsoft," <https://visualstudio.microsoft.com/zh-hans/vs/getting-started/>, 2022, [Online; accessed 23-Jan-2024].
- [14] JetBrains, "Get started - JetBrains Rider Documentation," <https://www.jetbrains.com/help/rider/Introduction.html>, 2022, [Online; accessed 23-Jan-2024].
- [15] OpenAI, "Introducing ChatGPT," <https://openai.com/blog/chatgpt>, 2022, [Online; accessed 23-Jan-2024].
- [16] "Kyrylo Kuzyk Github Link," <https://github.com/KirillKuzyk/Asset-Store-Support>.
- [17] L. Cranor, "Usable Privacy and Security, Slides: Usability and design Designing usable privacy and security experiments," *Spring 2022, Carnegie Mellon University*, 2022, [Online; accessed 23-Jan-2024].
- [18] e. T. De Freitas S., "The use of 'exploratory learning' for supporting immersive learning in virtual environments," *Computers education*, 52(2), pp. 343–352, 2009.
- [19] L. S. M., "Virtual Reality," *Cambridge University Press.*, 2023, [Online; accessed 23-Jan-2024].
- [20] K. B. S. M. C. Ben Kenwright, Abrar Ullah, "Introduction to Interaction Design," *Usability and User Experience 2021-2022, Heriot Watt University*, 2022.
- [21] A. K. B. M. S. Hoffman D. M., Girshick A. R., "Vergence accommodation conflicts hinder visual performance and cause visual fatigue," *Journal of Vision 2008*, 8(3):33, 2008.
- [22] "Vision - OpenAI API," <https://platform.openai.com/docs/guides/vision>, [Online; accessed 03-Feb-2024].
- [23] H. Weerathunga, "Unity Augmented Reality with Yolo | Realtime Object Detection with Augmented Reality," <https://www.youtube.com/watch?v=zeLzRsnVWIE>, 2021, [Online; accessed 23-Jan-2024].
- [24] S. A. J. H. A. T. Adam Crespi, Cesar Romero, "Unity\_Detection2AR," <https://github.com/derenlei/Unity-Detection2AR>, 2020, [Online; accessed 23-Jan-2024].
- [25] —, "Use Unity's computer vision tools to generate and analyze synthetic data at scale to train your ML models," <https://blog.unity.com/engine-platform/use-unitys-perception-tools-to-generate-and-analyze-synthetic-data-at-scale-to>, 2020, [Online; accessed 23-Jan-2024].
- [26] Matthew Hallberg, "Machine Learning with Synthetic Data | @unity + @TensorFlow," <https://www.youtube.com/watch?v=lHXcYhyhHGA>, 2020, [Online; accessed 23-Jan-2024].
- [27] "German Wiktionary," <https://de.wiktionary.org/>, 2022, [Online; accessed 08-Feb-2024].
- [28] M. A. P. Team, "MediaWiki Wiktionary API," [https://www.mediawiki.org/wiki/API:Main\\_page](https://www.mediawiki.org/wiki/API:Main_page), 2023, [Online; accessed 08-Feb-2024].