



Hochschule für Technik
und Wirtschaft Berlin

University of Applied Sciences

Design and Development of an Online Multiplayer Card Game using Unity Netcode for GameObjects

Bachelor Thesis

Name of the Course of Studies

Internationale Medieninformatik Fachbereich 4

submitted by

Laura Unverzagt

date:

Braşov, 10.03.2023

First Assessor: Prof. Tobias Lenz

Second Assessor: Mr. Claudiu Barsan-Pipu

Affidavit

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig verfasst habe, dass ich sie an keiner anderen Hochschule und in keinem anderen Studiengang als Prüfungsleistung eingereicht habe und dass ich keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Alle Stellen der Arbeit, die wörtlich oder sinngemäß aus Veröffentlichungen oder aus anderweitigen fremden Äußerungen entnommen wurden, sind als solche kenntlich gemacht. OpenAIs ChatGPT (Feb 13 Version) wurde zur Verbesserung von Texten genutzt.

I hereby declare that I have written the present paper independently, that I have not previously submitted it to any other university or course of study as an examination and that I have not used any sources and aids other than those indicated. All passages of the work, which were taken literally or analogously from publication or from other foreign statements, are marked as such. OpenAI's ChatGPT (Feb 13 Version) has been used for improvements of the texts.

Abstract

This paper presents the design and development process of Obsidia, a multiplayer card game created using Unity Netcode for GameObjects. The game's ruleset, structure, and software architecture are described, along with an implementation of agile work methodologies and Game Development Life Cycles in the project's development.

Multiple testing iterations were conducted by third-party testers, who found various bugs and provided insightful feedback on the game. The paper aims to provide insights into the production workflow of a multiplayer game, to present a possible structure for a game made with Unity Netcode and to offer a resource for developers interested in utilising Unity Netcode.

As a finding, the study emphasises the importance of preparation and planning when creating networked games and highlights the usability of Design Patterns in networking structures. Through the game's software architecture, the paper provides a reference for future developers worldwide and raises awareness for the complexity of creating multiplayer games. The game's development has been a success, as the prototype is functional and enjoyable, although limitations remain.

Furthermore, the paper discusses player feedback, debugging, and testing of the game, and presents insights gathered about Unity Netcode and the project management tools utilised. It concludes with recommendations and suggestions for the industry of the tool's usability and plans for the game's future.

Contents

1. Introduction	5
2. Methodology	7
2.1. Development	7
2.2. Timeline	9
2.3. Testing and Bug Fixing	11
2.4. Tools and Programs	12
3. Game Design	14
3.1. Research and Design	14
3.2. Rules	15
3.3 Deck	17
3.4. Score System	17
4. Programming and Architecture	18
4.1. Card Deck Structure	18
4.2. Unity Engine Tools for the Process	19
4.3. The Game Structure	19
5. Unity Netcode	23
5.1. Networking Structure	23
5.2. Encountered Issues	24
5.3. Usability of Unity Netcode	25
6. Testing and Bug Fixing Results	26
6.1. Development Bugs	26
6.2. First Playtest Iteration	27
6.3. Second Playtest Iteration	29
7. Conclusion	31
8. Recommendations	33
8.1. Guidelines	33
8.2. Future Work	33
9. Limitations	35
10. Sources	36
11. Appendices	39
Appendix A - Prototype	39
Appendix B - Questionnaire	40
Appendix C - Game structure	42
Appendix D - Test results	43

1. Introduction

Multiplayer card games present unique challenges for game developers due to the complexities of network programming and game mechanics. To implement multiplayer aspects, a multitude of tools that improve the development progress and simplify complex networking topics are accessible on the market.

Unity Netcode for GameObjects is a recently published high-level networking library built for Unity. It allows easy programming of multiplayer games and provides networking frameworks. Despite being in production for several years, Unity Netcode was released to the public in June, 2022, which explains the limited resources and video game projects using this library [\[1\]](#). The most notable game utilising Unity Gaming Services so far has been “Among Us”[\[2\]](#) [\[3\]](#) [\[4\]](#) by Innersloth in 2018. Many of Netcode’s features, modules and tools have only recently been released, leading to few resources.

This paper aims to fill the gap by covering the process of creating Obsidia, a multiplayer card game designed using Unity Netcode for GameObjects. Based on the classical German card game “Skat” [\[5\]](#), Obsidia takes references from “Skull King” made by Schmidt Spiele [\[6\]](#) as well as “Wizard” by Ken Fisher [\[7\]](#). By utilising the Unity Netcode library, Obsidia provides a multiplayer game for up to five players per game session. It offers a host-client connection from multiple devices, guides the players server-sided through the game rounds and calculates points as well as the winners of the game.

Research Objective

This paper presents the complete process of programming the game, points out struggles and gives practical recommendations for using Unity Netcode. It provides insights into a video game's production workflow and suggests a structure for a game made with Unity Netcode.

According to Marc Prensky[\[8\]](#), the first element of what makes games engaging is the ‘fun’ element, driven by enjoyment and pleasure. Hence, two objectives were formulated for this paper:

- **Playability and functionality.** Create a functional game prototype that utilises Unity Netcode and document the full game development cycle in order to provide recommendations that add value to the industry.
- **Enjoyment.** Bring entertainment and involvement to Obsidias players by evaluating the enjoyment of the player experience.

Relevance of the Research

The research objective is to raise awareness about the complexity of creating multiplayer games and to offer a resource for developers intending to utilise Unity Netcode. By presenting the software architecture of the game, this paper provides a reference for future developers worldwide.

Structure of the Research

The following chapters will start by covering the project management methodologies used to plan the process, the project milestones achieved along the way, and the tools and programs that contributed to its creation.

A comprehensive guide to the design of the game will follow and Obsidia's design, ruleset and score system will be introduced.

Chapter four will delve into the software architecture and programming to provide an overview of the structure design decisions made during the prototyping process.

Furthermore, the paper will explain the network structure behind the game, detailing its features. Afterwards, it examines the insights gathered about Unity Netcode, assessing its effectiveness for future games and identifying attributes to consider.

In chapter six, this paper will summarise player feedback and share discovered bugs from the repetitive game testing. Feedback will be analysed and evaluated.

The final chapters will conclude the findings and offer practical ideas and suggestions for future research, while also addressing the paper's limitations.

2. Methodology

The Project Management section describes the planning of this paper and prototype (see Appendix A). It details the milestones and timeline that have been made for this project and shows the methodologies used for an agile workflow. Tools and programs utilised for this purpose are explained and listed below. Finally, this chapter provides a detailed account of the testing process, the role of the gameplay testers, and the measures taken to address the identified bugs, focusing on enhancing the overall gameplay experience.

Objective

The objectives of this paper (see chapter 1), will be measured by analysing the playtesters feedback and used to determine the success of the research. The feedback will be gathered via a questionnaire (see Appendix B) that is distributed among the sample. The gathering of player feedback is detailed in subchapter 2.3., as the feedback will be assessed according to two factors:

- Playability and functionality of the game build
- Entertainment and enjoyment of the playing experience

2.1. Development

For the development of the game, agile methodologies, specifically SCRUM, have been combined with Game Development Life Cycle Guidelines (GDLC). Different GDLC guidelines have been used and combined for an efficient development process [\[9\]](#).

GDLC describes a development guideline or strategy that envelops the process of game creation which is split into phases. Diverse models are known within the industry that differ in their phases and execution. For this project, the timespan, scope and approach needed to be taken into consideration. A combination of the GDLC proposed by Blitz Games Studios and the GDLC proposed by Arnold Hendrick has been tested and used for this purpose.

As the pitch was the proposal of this paper, it has not been taken into consideration for this development cycle. Additionally, due to time constraints, preproduction and production have been combined into a minimal viable product (MVP). Therefore, the production pipeline has been designed as shown in Figure 1.

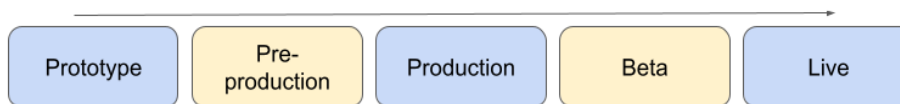
Blitz Games Studios GDLC:**Arnold Hendrick's GDLC:****Obsidia's GDLC:**

Figure 1. Comparison of the Blitz Games Studios, Arnold Hendrick's and Obsidia's GDLC

To further organise and plan the development cycle, agile development with Trello as its tool has been utilised (see the Tools and Programs chapter for a full overview). As explained in the paper by Al-azawi, Ayesh and Obaidy, agile methodology is “[...] an iterative and incremental approach, and it achieves quality and productivity through iterations. Each [iteration] of sprint phase[s] includes a software development team working through a full software development cycle including planning, requirements analysis, design, coding, unit testing, and acceptance testing [...]” [\[10\]](#).

Sprints have been organised according to the Game Development Life Cycle, meaning each sprint ended with each new milestone, and with it, a new sprint started. Toward the end, the methodology of ABC-sprints was used. This method describes the splitting of sprints in Alpha, Beta and Completion sprints and aligns with the GDLC described above. [\[11\]](#)

Additional inspiration for development has been taken from Model-Driven Game Development (MDD). During the Production Phase, Unified Modeling Language (UML) structure diagrams have granted an overview and greater insight into the complexity of the game. Though the development technique has not been specifically used for this project, the core intent of creating structures before development has helped greatly during the planning process for sprints and development cycle phases. [\[12\]](#)[\[11\]](#)

The development methods described above have been used to prevent some of the hurdles in the development process, such as scheduling problems, crunch time and scope and feature creep problems [\[10\]](#). To further avoid feature creep problems, milestones for each sprint have been concretely defined in tasks and goals. The project milestones and a basic overview, as well as a time schedule, have been provided in the “Timeline” section to grant further insight into the process of development.

2.2. Timeline

The project started on the 18th of November, 2022 and was finished on the 10th of March 2023. It took 16 weeks to complete this paper and prototype, including testing, programming, planning and writing this work. To utilise the given time to its fullest, milestones following the phases of the GDLC mentioned above have been planned.

Each phase of the GDLC aims to provide a functioning game, meaning a version that works as bug-free as possible and still offers fun. Below described are the details involved in each iteration of the game, as well as explanations of the reasoning behind features and priorities. In Figure 2, the timeline and length of sprints are presented, showing the effort and time put into every phase and milestone.

Production/MVP

For the minimal viable product and the first iteration of the game, the aim needed to be a feasible and functional game in the engine. As the design was already prepared, the game mechanics were clear: The goal was to avoid overshooting and instead create a solid basis for the game and, therefore, a working card game. This didn't mean a complete game but rather a simplified version with only colour cards.

The networking aspect had priority, as it was the core of the game. Implementing a functioning connection between at least two, but preferably multiple people with a host-client architecture was the first step to achieving functionality.

After establishing a connection, the functionality of a game loop had to be ensured. This involved the playability of cards, the successful calculation of a winner of the trick and player feedback regarding the winner of the round. Said calculation did not require to consider a game-winner, but rather the momentary successor after every played card. This meant the game itself was rather simple at the time but provided the basic structure to build any further features.



Figure 2. Timeline for Obsidia

For player feedback, a simple user interface (UI) was required. Card colour, card value and buttons to play cards were desired, as well as a pile showcasing played cards of the round.

Alpha

Goal of the alpha was the inclusion of all necessary features explained in the design of the game. This means gameplay aspects, extended connection options for multiple players and improved UI.

Since the most enjoyment of the game comes from its multiplayer aspect, varying amounts of possible player connections for up to five players were important in this iteration. In addition, a proper relay system needed implementation because multiple-player sessions could not be created without it. If friends wanted to play together, they also needed the ability to connect to a specific host.

This led to another important feature: a menu with connection options, a rules section and a way to exit the game was required. This concluded with updated UI functionality and additional player feedback during the game.

More gameplay features were required in the game. Of particular importance to game balance was the functionality to change the player who plays the first card each round. Furthermore, the special cards and their functionality were needed. And to keep fairness and include the goal of the game, a proper win-and-lose system with special cards has been taken into consideration.

Lastly, a point system with player feedback, calculating points depending on your announced tricks was needed, and thus a possibility to announce your tricks before each round.

Beta

Though many additional features were planned, many were cut due to time constraints. The beta aimed to add “nice-to-have” features, meaning no more crucial additions but rather small features that increase the players’ experience and enjoyment.

The art style was improved and some more assets were added, UI was touched upon and the replayability function resetting the game has been implemented. In addition, features requested by testers have been included in the release candidate. Many UI improvements have been added, such as an exit button when players are connecting, visuals for the announced trick, the round number and the number of won tricks as well as improved rule explanations with an in-game info menu explaining the rules.

2.3. Testing and Bug Fixing

Software testing is a critical process for ensuring the quality and reliability of games. To mitigate potential biases and ensure the accuracy and validity of testing results, the expertise of third-party gameplay testers has been leveraged.

The testers for this paper were selected using a convenience sampling approach, a non-probability sampling method, from a pool of volunteers from diverse backgrounds. The majority of the testers were sourced from gaming communities on the Discord platform, while others were selected from personal circles or companies in the game industry. This approach ensured that the testers had a good understanding of the software and were able to provide detailed and precise feedback on the game's performance. To avoid potential biases, all testers had no prior experience with the game prototype before testing and were not made aware of the rules beforehand. This approach ensured that the testing results were unbiased, reliable, and representative of the end-users perspective.

By involving external testers, the aim was to minimise potential biases arising from the familiarity with the project and ensure that the testing results were reliable, and representative of the end-users perspective. The use of third-party testers allowed the approach of testing from an external perspective, thereby gaining a more comprehensive understanding of the game's strengths and weaknesses.

Testing has been planned in two iterations and two test groups. The first iteration and test group involved the product after the Beta phase (see chapter 2.2 for more information). The main goal of this testing group was to find bugs and report and categorise them via a questionnaire, the so-called Bug Report (see Appendix B). Additionally, they were asked to leave feedback on their first impression of the game and suggest features they have been longing for during testing via a different questionnaire, henceforth referred to as the Gameplay Report (see Appendix B).

After bug fixing and implementing the suggested features, both test groups were to test the release version. Though another bug questionnaire was given to both groups, the main purpose of this testing iteration was to gather feedback on the game. By creating two test groups, different perspectives have been given on the game. One group had never seen the game before, while the other knew its Beta and how it looked before improvements. This way, broader feedback was expectable.

The feedback during both iterations was gathered through a questionnaire (see Appendix B) that covered questions involving game balance, the aspect of fun, the play time of the testers and the overall satisfaction of the players.

2.4. Tools and Programs

The following points describe tools, programs and applications used for the development of this project and paper. They cover organisation, production and evaluation.

Unity and Unity Netcode

Unity is a cross-platform game engine developed by Unity Technologies. It is the main platform of development for the game. Additionally, Unity Netcode is a first-party, mid-level networking library built for the Unity game engine that will be used for the development of the multiplayer aspect [\[13\]](#). C# is one of the supported languages of Unity and will be used for all scripts.

Unity and Unity Netcode have been chosen as the main production technologies, not only for this paper but also due to personal interest in the technologies and the familiarity of this author with the engine in the past. As Unity's intended purpose is the creation of games, and since Netcode was specifically invented for multiplayer aspects, they have been chosen for the creation of Obsidia [\[14\]](#).

Google Drive, Documents and Google Forms

The google services, specifically Google Drive [\[15\]](#), Google Documents [\[16\]](#) and Google Forms [\[17\]](#), are cloud-based services providing storage, survey administration software and document editing [\[18\]](#). The organisation of documentation for this paper, the rules of the game and notes on the programming progress were essential. Google Docs provided a remote option for easy access from any device and enabled structure thanks to Google Drives folder structures.

Additionally, Google Forms was used for the testing iterations, providing questionnaires for testers and gathering their answers.

GitHub

GitHub [\[19\]](#) is a cloud-based source and version control website [\[20\]](#). Used by students, programmers, web and game developers and many more, it is widely known and free to use. Having used it in my studies before, it became the most important tool for ensuring code stability and the possibility of continuing my work in different locations without issues.

ParrelSync

ParrelSync is described and explained on its Github website as follows:

“ParrelSync is a Unity editor extension that allows users to test multiplayer gameplay without building the project by having another Unity editor window opened and mirror the changes from the original project” [\[21\]](#).

With ParrelSyncs features, the creation of multiple copied projects is possible, which are updated after saving changes in the original Unity Editor window or scripts. Testing the game, code, assets, and performance is essential in the process of game development. With ParrelSync, debugging on both the host and client-side is possible.

Trello

Trello is a web-based, kanban-style, project management tool [\[22\]](#). It is free to use and features multiple boards with customisable tasks.

Within each list, you can create cards to represent individual tasks, such as creating a specific character model, implementing a new game mechanic, or fixing a bug. You can add labels to these cards to indicate priority or type of task, and you can also assign due dates and members to each card [\[23\]](#).

One of the key advantages of Trello is its flexibility. Customisable boards, lists, and cards are adjustable to the needs of its user. The drag-and-drop interface simplifies the movement of tasks between lists as they progress.

To keep an overview of objectives for the week, tasks have been organised on boards specific to their milestones and deadlines. Should a task not be finished by the end of the milestone, it was moved to a new board for the next milestone.

3. Game Design

The following passage describes the game, its rules and its scoring system. It details the objective of the game as well as the deck and its cards. Understanding the mechanics of Obsidia is essential to comprehending the choices that have been made in the progress of the prototype's creation and following the architecture that has been designed.

3.1. Research and Design

The idea of this game came from my personal interest in physical card games. Whether for a family game night or a pub game, card games have always held an important role in our society. As an engaging activity, they were used for gambling, socialising and entertainment [\[24\]](#). Many modern card games come hand in hand with digitalisation. A good example is the “Uno” game, at first created as a physical card game based on “Mau Mau”, but is nowadays digitised by Ubisoft [\[25\]](#). Inspired by said modernisation, the goal was to create an online multiplayer card game. Prototypes were physical at first, to ensure a stable design and basis of enjoyment in the game. After the creation of a sufficient design, the development and research of the prototype were digitalised.

For Obsidia, a card game based on the principles of “Skat” and the “Stich”, also called trick, was created, where each player plays a single card each turn and wins the played card stack if their card has the highest value [\[5\]](#). As Skat is a rather complicated game with many mechanics and special cases, simplified versions have been looked into. Inspiration was found in the games “Wizard” [\[7\]](#) and “Skull King” [\[6\]](#).

The intention was to keep the mechanic of predicting your total number of tricks each round. Similar to “Wizard”, four different colour cards with values from 1 to 13 were created, to keep the balance of the game. The first played card in a match defines the colour to be served. Since colour has priority over the value of cards, dynamics within the game change. Through this mechanic, the first player has the strategic advantage of predicting and planning their next moves, as well as dominating the tone of the game. This also results in a power imbalance. With inspiration from Skull King and “Wizards Special Edition” [\[26\]](#), Special cards have been implemented to break the power imbalance of the first player and make the game more dynamic.

Elemental, Fiend and Celestial triumphing over all colour cards break the otherwise flat hierarchy. Due to the rock-paper-scissors principle between the three cards and the different amounts of cards, a win is not guaranteed, yet again shifting the balance, complexity and strategic planning of the game.

The Gate card, under normal circumstances, is a guaranteed failure. Inspired by the “Escape Card” in Skull King, it gives the option of intentional loss, therefore avoiding an overload of won tricks. In Skull King, it gives the player safety, as whatever else happens in the game doesn't affect the card. This security was intentionally broken in

Obsidia with the Fey card. With only one in the deck, the Fey is unlikely to show up in the first few rounds. But strategies change with it in the deck, as it ensures a guaranteed win for the first Gate card played, therefore, breaking the otherwise automatic failure.

With these five cards, the flat hierarchy has changed, strategic aspects have been introduced to the game and balance has shifted. The player's next moves require more careful consideration and engage the player further in the game.

3.2. Rules

Objective of the Game

The objective of the game is to defeat your friends by earning the most points. Points are earned by correctly guessing the number of tricks you will win in each round. The player with the highest score at the end of the ten rounds wins the game.

How to Play

The game is played in ten rounds. At the beginning of each round, players receive a number of cards equal to the round number. For example, in the first round, each player receives one card, in the second round, two cards, and so on. Players keep their cards hidden from their opponents throughout the game.

At the start of each round, players must announce the number of tricks they believe they will win. The highest number a player can guess is the number of cards they hold. Points are earned based on how close the announced trick count is to the actual number of tricks won. If the announced trick count does not align with the actual number of tricks won, the player loses points.

Each round consists of several sub-rounds, equal to the number of cards held by each player. In each sub-round, each player must play one card, and a trick is won by the player with the highest card.

Sub-rounds and the Trick

The game begins with the host playing a card on the pile. This card determines the colour that must be served for the current trick. Subsequently, each player places one card on the pile. This sequence of card plays is known as a trick and concludes a sub-round.

At the end of the trick, the player who played the highest-valued card wins the trick and earns the right to play the first card for the next sub-round. The value of the card depends on both its colour and its number. For instance, if a green 3 is played first, the

colour green must be served. This means a green 6 would trump a blue 12. If none of the other players has a green card to play, the first player wins the trick.

If all cards have been played, the round ends and players receive points based on the number of tricks they won and the bet they made at the beginning of the round. The player who correctly guessed the number of tricks they would win earns points, while those who incorrectly guessed lose points.

Special Cards

The game features five special cards, each with unique properties that affect gameplay:

Elemental	Elementals defeat all colour cards and beat Fiends. However, they are overtrumped by the Celestial card.
Celestial	The Celestial card can defeat Elementals but is defeated by Fiends due to their rugged nature.
Fiends	The Fiends can defeat the Celestial card but can easily be shut down by Elementals.
Gates	Gates are quick escapes, always losing the trick unless the Fey has been played.
Fey	The Fey trumps all other colour cards but is beaten by Elementals, Celestials and Fiends. When a Gate is played, the Fey reverses the Gate and makes it the highest-valued card in the trick, higher than even other special cards. This is the case whether Fey has entered the trick before or after. If multiple Gates have been played, the first Gate wins.

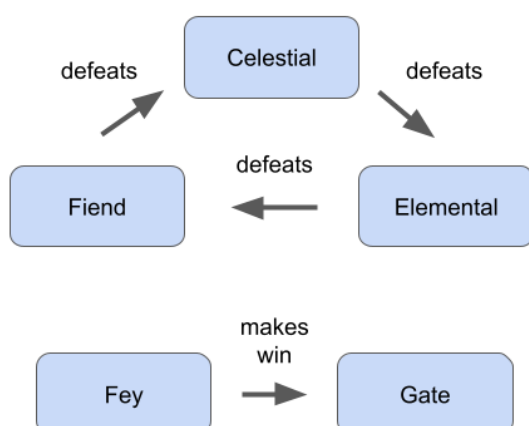


Figure 3. Card effect diagram.

If Elemental, Celestial, and Fiend have been played, the cunning Fiend trumps them. If multiple cards of the same type have been played, the first card wins. The gate would win, if all types of special cards were played (see Figure 3 for more information).

3.3 Deck

The deck for the prototype contains the following cards and card amounts:

- | | |
|-------------|---------------|
| - 13 blue | - 5 Elemental |
| - 13 red | - 2 Fiends |
| - 13 purple | - 1 Celestial |
| - 13 green | - 5 Gates |
| | - 1 Fey |

3.4. Score System

If a player guesses their tricks correctly, they gain 2 points, plus an additional 2 points for each trick they won during the round.

If a player loses the round, they lose 1 point for each extra or missing trick. For example, if a player bets on winning 3 tricks but only won 2, they would lose 1 point. If they bet on winning 3 tricks but won 4, they would also lose 1 point.

If a player correctly bets on winning 0 tricks, they gain 2 points plus additional points equal to the current round number. For example, in round 3, correctly betting on 0 tricks would earn the player 5 points ($2 + 3$).

4. Programming and Architecture

The structure of the prototype is split and explained in this passage. Beginning with the overall game structure, it presents Unity Engine tools made for the project and software architecture within the prototype. Going over the scripts and their functionality, describes design choices from a technical standpoint. In addition, the networking structure with Unity Netcode is defined and issues that have been encountered with the Netcode library are shown, along with personal advice for programmers using the said library.

4.1. Card Deck Structure

To create a card game, one needs a card deck. It is a core part of the project. To keep adaptability and an easily changeable deck, both the cards and card deck (in the Unity editor named “card list”) are scriptable objects. Scriptable objects are data containers that are independent of class instances [\[27\]](#).

Each card contains an index, to identify it easily within the code. Additionally, enumerations (enum) were used to differentiate between different card types. One enum specifies the colour of the card, with special cards owning their own “colour” called “special”. The other specifies the type of card, with all colour cards owning the enum “humanoid”. These tags simplify efforts of differentiating cards in the Solver, where the winner of a round gets defined.

Card Lists on the other hand contain a list of card scriptable objects. This way, they act as the deck and open easy access to all cards. This method not only allows for easy flexibility of cards within a deck, but it also opens the possibility of multiple different decks that can be changed by either the programmer or maybe even players later on from any point onward, both in and out of the game. For even easier populating of card lists, an editor function gathering all cards from a specified folder has been created.

To display cards to the player, two prefabs have been made that change their visuals depending on the card. Changes are taken from the logic provided by the scriptable card object and change the text and colour of the visual prefab. This offers a performance-friendly solution, as played or drawn cards don’t need to constantly be spawned in or destroyed, but rather activated and deactivated within the scene. While one prefab visually represents drawn cards that the player can play, the other shows played cards that lie on the pile, visible to all players. All prefabs are already in the scene and are activated and deactivated when needed.

4.2. Unity Engine Tools for the Process

Since creating a full deck of over 50 cards is very time-consuming, a card creation tool for the unity editor has been made. After selecting a target path for the Scriptable Objects to be created, it has two core functions: create singular cards and create multiple cards. Creating singular cards can create any card desired. Options to select and specify all data contained within the scriptable card object provide customizability of the card. This is especially useful for special cards that only exist in limited amounts.

Meanwhile, the option of creating multiple cards is particularly practical for colour cards. The way it works is that it takes over all data values for created cards specified by the user, but changes the value of the card automatically. Starting from one, it increases and creates cards up to the user-restricted amount. This means if the user wants cards with a value of up to 15, it creates 15 cards, starting at value one, then two, then three and so on. This saves time and automates unnecessary work.

4.3. The Game Structure

The software architecture and game structure are described further below. Scripts and components are shown and explained in detail, presenting their importance to the system. For a visualisation of the architecture, see Appendix C, Fig. 4 for a UML of the full game structure.

TurnManager

The functionality of the game is controlled by the TurnManager class. Said class is the context of a finite state machine based on the state pattern. The state pattern is a behavioural design pattern that allows objects to alter their behaviour upon the change of their state. Using the abstract base called “Turn_Base”, instances of concrete states, labelled with the prefix “Turn_” are created within the TurnManager. States upon finishing their functionality, autonomously invoke the method within the TurnManager to change to the next state [\[28\]](#) [\[29\]](#) [\[30\]](#). See Figure 5 for a detailed view of the state pattern implementation in Obsidia.

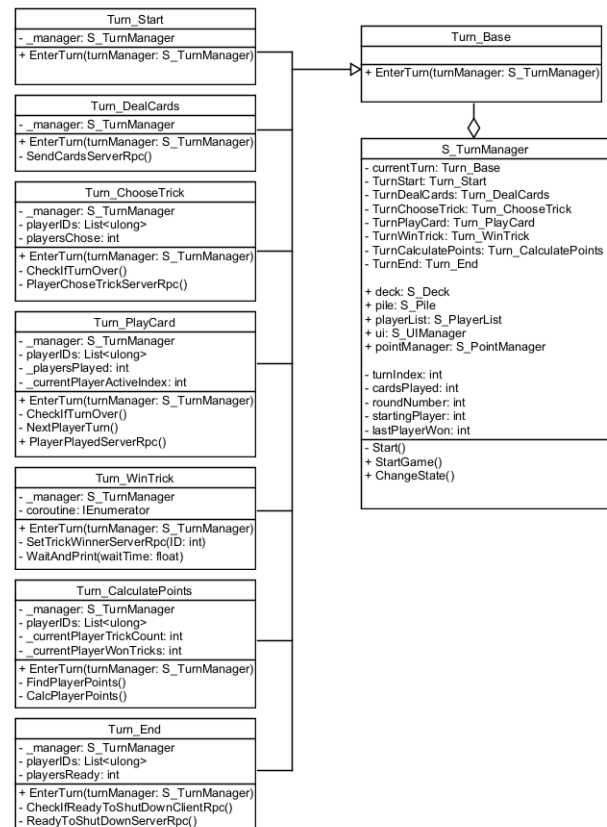


Figure 5. State Pattern structure used in Obsidia

Utilising the state pattern offers multiple advantages. Most card games already consist of a turn-based play methodology, resulting in usable states to control the flow of the game and the current turn every player is in. Additionally, it offers control for the programmer for debugging and testing, as each state encapsulates its methods and functionality. Furthermore, thanks to the context, references to other classes are provided with ease.

Listed below are the turns and their basic functionality:

Turn_Start	First State of the game, begins the turns
Turn_DealCards	Deals cards from the deck to all players
Turn_ChooseTrick	Gives players the UI to select their predicted trick count and sends the prediction to the server
Turn_PlayCard	Allows the first player to play one of his cards. Goes through every player and lets them play a card to send to the pile, disables all other players from playing until it is their turn. Finishes when all players have played
Turn_WinTrick	It controls the inspection of all cards and requests the winner from the Pile and Solver
Turn_CalculatePoints	Requests points for all players from the PointManager upon finishing a round
Turn_End	Resets the UI and disconnects the host from the server, resulting in players disconnecting

Deck

Responsibility of the deck is to draw random cards for all players and hand out cards depending on their index. It owns a reference to the CardList scriptable object and with it all cards. Should decks be changed or switched, this reference is responsible for the CardList that is in the game.

The method DrawCardIndex(int cardsToDraw) draws cards for a player. It uses a list of all current cards still within the deck, and then randomly selects a card, checking whether it is in the current pull already to avoid drawing the same card multiple times. This method gets called for each player, providing all the cards for the round.

In addition, since scriptable objects cannot be sent through the network (see “Encountered Issues”), the deck provides all information about cards with the GetCard(int index) method. Accordingly, a search algorithm through the deck has been implemented. Binary search is the used option, as it is feasible and quick [\[31\]](#). Since its

complexity reaches $O(\log n)$ with n being the number of cards within the deck, the current complexity is around $O(\log(66))$.

Pile and Solver

The Pile and Solver are responsible for collecting all played cards and returning the winner of the trick.

The Pile has four important methods. One provides it with the card and the player who played it, one calls the solver to calculate the winner, one returns the index of the winner and lastly, a method resets its lists. The Solver's only reference lies within the Pile.

The Solver contains one method being called within the Pile called "SolverCalc". Using guard clauses, it has multiple iteration cycles. The first iteration cycle contains all played cards. It defines the colour that is to be served and specifies the first gate as the winner, should the edge case of only gates within the pile occur. The next iteration deals with colour cards, defines the winner among them and sorts out special cards. Should no special cards occur within the pile, this is where the method returns the winner. Otherwise, it will continue and specify the winner among the special cards, considering all possible scenarios.

Splitting the actual winner calculation from the pile allows for a simpler structure and clearer communication between the pile, the server and the client. It in addition supports the SOLID principles, specifically the single-responsibility-principle and brings forth an improved structure in the code [\[32\]](#).

PlayerHand

The PlayerHand is part of a prefab that gets spawned into the scene upon connection to a host, or upon starting a host. It is representative of the Client and can be found through multiple different methods from the NetworkManager.

The PlayerHand manages the cards that a player draws, sends selected cards to play to the pile and updates a player's UI. It is the main reference and object for the client and connects the actions of the player through the UI to the logic of the game.

Since the NetworkManager controls the spawning of the PlayerHand and assigns it to a specific client, many advantages are given. Not only is it simple thanks to Netcode-specific methods to find the local Player Object, it additionally ensures the specific amount of objects needed is within the scene.

PointManager

The PointManager's responsibility is to track both the predicted tricks as well as the won tricks of all players. Multiple Lists have been utilised to track every player's ID,

prediction and win count. The win count gets updated in the WinTrick turn and resets upon finishing the round.

The only reference to the PointManager is within the TurnManager and the UI, and only the server tracks the predictions and won tricks. This gives the system structure and avoids unnecessary calculations on the client side.

The basic function of the PointManager is as a storage system since all calculations for points happen in the turns and their functionality, specifically the CalculatePoints turn.

PlayerList

The PlayerList contains all player IDs and registers players upon connecting in a List. It is the main provider of player count and player IDs for all other scripts. Scripts such as the deck use it via the TurnManager. It mostly functions as a container to track the current players. Players upon connecting, call the script and register themselves with their local player ID.

User Interface

Goal of the user interface (UI) and all scripts utilising UI elements is to keep it as disconnected from the logic of the game as possible. That means any calculations and data being sent or shared were to be separate from buttons and texts. Thanks to this approach, the separation led to easier bug fixing and problem identification, as identifying whether something was a visual bug or a logic issue was simplified.

The responsibility of the user interface has been split into three scripts. The ConnectionManager deals with the relay system and all functionality to connect with the host. It not only binds different menu panels upon connection but also ensures that the connection logic corresponds to the UI and that the game starts when all players are ready.

The UI_Cards Script deals with updating the visuals for both the cards as well as the pile cards. It ensures the correct colour, value and type of the card being displayed on the card or pile prefabs, taking said information from the Card Scriptable Objects.

Lastly, the most important script is the UI Manager. It handles the management of all UI within the game, including the activation and deactivation of cards and pile cards. It ensures the proper display of points and predicted trick texts and win-and-lose texts. Furthermore, it enables and disables the prediction buttons during the ChooseTrick turn of the game, manages to disconnect the player after the game and closes the application when the player decides to leave.

5. Unity Netcode

The upcoming chapter focuses on Unity Netcode and the networking architecture used in Obsidia's project. It examines the capabilities of Unity Gaming Services and their implementation in the networking structure. The section also discusses the challenges faced during the implementation of Unity Netcode and evaluates its usability for the project.

5.1. Networking Structure

For the project, Unity Gaming Services have been used, specifically Unity Relay [\[33\]](#). A universal Relay server functions as a proxy for this method and connects players through it [\[34\]](#). Proxy servers are explained as follows:

“[...] a **proxy server** is a server application that acts as an intermediary between a client requesting a resource and the server providing that resource [\[35\]](#).”

The proxy delivers messages between connected players. In the specific case of the Relay server, it utilises low-latency datagram exchange between game clients, which means that multiple players don't connect directly to each other. This enables the usage of a Host-Client-Structure, meaning one player hosts the server for everyone else. Using the proxy additionally avoids revealing sensitive information to other players, like the host's IP address, granting further security [\[36\]](#).

To be able to reach the proxy though, the second component of Unity Relay is the Allocations service. It is the entry point to the Relay Servers and enables creating and joining gaming sessions. The Allocation flow is described as follows in the Unity Documentation: “The allocation flow, or the connection flow, is the process by which the Allocations service allocates a host player and any number of joining players to a game session [\[37\]](#).”

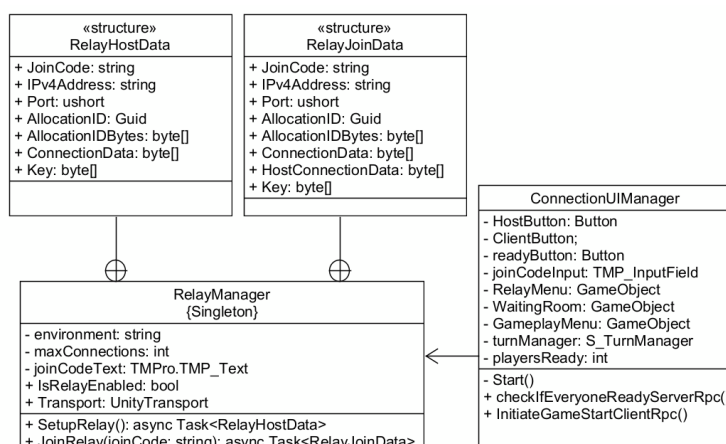


Figure 6. UML of the Networking Structure

The structure avoids the issue of hosting or maintaining game servers for players and additionally avoids the issue of third-party solutions outside of Unity. It is a simple solution to a complex topic that requires a lot of research outside of game programming. Instead, it enables creativity and quick implementation of systems with multiplayer aspects.

To implement this in Unity, the Network Manager and Unity Transport Script, provided by the Unity Netcode Library, utilises the Relay Unity Transport.

To establish a connection, the Connection UI Manager and RelayManager have been created. The UI Script establishes not only the correct visuals for players upon connection, but also calls upon the RelayManager.

The RelayManager contains two tasks as seen in Figure 6, SetupRelay and JoinRelay. To ease access to the script, it utilises the Singleton pattern [\[28\]](#). SetupRelay is called when starting to host and creates a new allocation with a new randomly generated code, that acts as an identifier for the created allocation. It takes the data from the allocation and sets it in the Relay Server provided. JoinRelay instead tries to find a connection with an already existing host by finding the allocation with the join code.

To ease the process of sharing data, the RelayHostData and RelayJoinData structures have been created. They function as a storage system for IP addresses, ports, allocation IDs and more. The main difference between the structures is that RelayJoinData additionally contains the data of the connected host.

5.2. Encountered Issues

Due to the limitations and restrictions of networking, many rules have been encountered that needed to be taken into consideration.

One of the most important rules of Unity Netcode is always to use a Network Object component for objects that require syncing in the scene. Additionally, scripts that require networking, need to inherit from NetworkBehaviour to sync it in the network [\[38\]](#). The NetworkBehaviour can be used on parent classes as well [\[39\]](#) [\[40\]](#). An example within the project for this is the Turn_Base, which all turns inherit from. This still requires each child object in the scene to contain a Network Object component, for otherwise, a null pointer can occur, as the object will not be recognised in the network.

Another important issue that has been encountered is the matter of open-ended variable types. Due to the way networking works, only closed types like integers, ulongs and arrays can be sent through the network, to avoid data leaks. This means not only Lists but also Scriptable Objects cannot be sent in a Client or ServerRPC.

This resulted in multiple workarounds for different situations. The deck and Scriptable Card Objects as examples, as cards could not be sent to players. Instead, each card has been given an index that can be sent either as an integer value or as a group of cards in the form of an array. This is the reason for the deck sharing all other information about the Scriptable Card Objects, as only the deck has access to them. A similar issue occurred with Lists for IDs, resulting in again utilising arrays for transport. JSON files could be used for larger information, though in this case, it would not have been worth the effort, since the data was small and not complex.

ClientRpcs are needed to contact a client from the server. It also is the case that ClientRpcs are executed on all clients, especially when contacting a single client. This behaviour is inconvenient and redundant, as the method gets executed for every client, whether they are targeted or not. Especially when one client tries to contact another, structure in the project is very important, as you can only get all ConnectedClients from the Server and have to request the ID of a client every time.

Finally, one of the biggest hindrances during this project's testing is: debugging with built-in debugging tools is almost impossible. When using breakpoints in the code of the client or host, often the connection between the parties can time out and get interrupted. This causes challenges, costs time and induces "Debug.log" use. Additionally, Unity rarely gives the programmer specific error messages regarding Unity Netcode issues. This is especially problematic when forgetting core functionality, like the implementation of NetworkBehaviour. Should it be forgotten, no ServerRpc or ClientRpc will work.

5.3. Usability of Unity Netcode

Unity Netcode has been a great asset to this project. It is a remarkable basis for Unity and is easy to implement. It is a good tool for beginners that don't want to spend too much time learning networking structures and instead dive into game programming, as it simplifies many processes.

As described in this paper though, there are still many issues with Netcode, from complicated debugging to general issues and many unwritten rules to consider while coding, that you cannot find in the documentation but instead have to learn while implementing. Not many resources exist yet, which makes the otherwise beginner-friendly Unity Gaming Services and Unity Relay much harder to get into and learn. The Unity documentation is a good place to get an overview, but even that is lacking.

Sometimes, repetitive code is required for Server-Client interactions, many solutions are still unclear and even the Unity Netcode library itself generates warnings in the console that come with the package. From the current standpoint, Unity Netcode requires more development, but has the potential of becoming a great competitor to other solutions on the market.

6. Testing and Bug Fixing Results

This chapter aims to examine the development bugs and bugs found by testers during the software development life cycle. It analyses the different types of bugs that were reported (see Appendix D for the feedback data) during the development process and the methodologies used to detect and resolve them, including their impact and their severity levels. Additionally, it discusses the feedback from testers and how it has been considered and evaluated during the development process.

6.1. Development Bugs

During the development of the game, a few bugs of a bigger scale have occurred that are worth noting. A bug to mention specifically is the deck bug.

The deck bug has been an issue since the first iteration of the game that has gone unnoticed for multiple weeks. The issue was the repeated pulling of cards from the deck, which was not supposed to be possible. Occasionally null pointers would occur, as if the deck list was looking for an unreachable index.

Two issues were noticed in the code. The first rather quick fix was that the deck was never reset, resulting in too few cards that could be drawn. This was a rather simple fix, as the reset for the deck was implemented in the turn system. The second and bigger concern was that whenever a card was drawn, it used the index of the list. By doing so, all other elements from the list have been moved, meaning all their indexes were misaligned by 1, resulting in the repeated drawing of cards and the drawing of incorrect cards. The same issue appeared when leaving the cards in the list since other clients also had to draw cards with new random numbers. To fix this bug, the index for each card has been implemented that is aligned with the original position in the ScriptableObject Card List. This way, with a binary search, cards were uniquely identifiable.

Another issue rather than a bug was that during development an additional turn was needed, as tricks needed to be announced before the beginning of the round. This had not been taken into consideration at first. Luckily, the systems and codes architecture was created in a way that it was easily adaptable, resulting in no larger issues.

Since multiple devices have been used during the creation of the project prototype, another issue with Github has been encountered. A script name was not properly capitalised during its creation but has been renamed in a later commit to Github. Github does not recognise capitalisation in the commit changes and encryption though, resulting in the correct name within the script, and the wrong name for the script. Unity requires script names and class names to be identical, which resulted in later issues when pulling the project from the Github repository. After noticing this, the issue was resolved by completely renaming the script.

6.2. First Playtest Iteration

The first playtest iteration brought the initial feedback to the game as well as numerous reported bugs that needed fixing.

Over 50% of the reported bugs turned out to be gameplay-related (Fig. 7.), many of them breaking the game (see Appendix D for full data). Many of the issues that the testers found had not been spotted before since earlier testing mostly happened with two players. One of the most reported gameplay bugs was described by a tester as follows:

“If a different coloured card is played after the first card, and then a card that should beat the first card is played, the different colour will win”

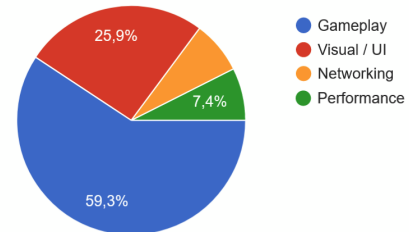


Figure 7. Diagram categorising reported bugs of the first testing iteration

As an example, if a blue 4, a red 2 and a blue 9 in that order have been played, the red 2 would win, regardless of its value. This bug was caused by a core issue within the code. Due to the guard clauses within the solver, the counter considering the winner ID of the round often did not get increased before continuing the loop, resulting in the wrong person winning. This issue occurred with multiple special cards as well, but has been an easy fix otherwise.

Yet two more notable bugs occurred with the special cards as mentioned by the testers:

“If a fey is played first, and a fiend is played in the round, the fey still wins. Applies to [fey, colour, fiend], as well as [fey, fiend, colour]. Not tested with other special cards.”

This issue has occurred since fey, fiend, celestial and elemental have been part of the same “if - else if” statement. Creating two if-statements was the simple fix for this issue (Fig. 8).

```
if (isFeyInPile) specialCard = CardType.Fey;  
if (isElemInPile) specialCard = CardType.Elemental;  
else if (isCelesInPile) specialCard = CardType.Celestial;  
else if (isFiendInPile) specialCard = CardType.Fiend;
```

Figure 8. Code from the S_Solver Script, demonstrating the fixed bug

“When the first player places a gate, and the second one a color [sic] (virtually becoming a trump), the last player to play a higher of the same trump color [sic] loses regardless.”

This bug was a result of the default special card value being a humanoid. Normally a guard clause would have prevented the comparison to run, but since a gate is considered a special card, said code would run regardless, making the first humanoid, meaning the first colour card, win. As shown in Fig. 9., this issue was solved by implementing yet another guard clause, preventing humanoid-typed cards from changing the winner card.

Most other bugs that were reported derived from UI issues and visual bugs, such as “misaligned text”, “hard-to-read text”, “too slow scrolling for rules” or “words overlapping”. Another bug that was reported multiple times was the “leave-button not working”, which happened due to a missing connection to the script. Most of these have been overall simple fixes that were quickly implemented.

```
counter = 0;
foreach (SO_Card card in SOcards)
{
    if (specialCard == CardType.Humanoid) break;

    if (card.cardType == specialCard)
    {
        winnerCard = card;
        winnerPlayer = playerIDs[counter];
        return winnerPlayer;
    }
    counter++;
}
```

Figure 9. Functional code from the S_Solver Script with a guard statement

A few additional networking bugs were reported, explaining how the game breaks upon a player joining during a running game and similar. Furthermore, the PointManager breaking for clients has not gone unnoticed by testers, which were known issues, yet to be fixed.

First Playtest Feedback

Overall despite the bugs, testers of the first iteration considered the game a positive experience. With an average of five to ten played rounds per tester, a sufficient overview has been gathered of the game. Players reported excitement about the game, but also their impeded experience due to the issues caused by bugs:

“Really fun! Once we sorted out the bugs that we knew, I enjoyed my time playing the tactics of the game.”

“i liked the game, interesting concept, decent execution”

“i like the core game. but all the UI is really buggy [sic] and would need to be better for the game to really be funn [sic]”

“I like the design and the Idea, just had gameplay bug issues”

Testers additionally came back to the game, asking further questions about the development and the planned features. The bugs were, as said by most, the main issue the game had. Otherwise, as shown below, the players reported being happy with the balance and game overall:

“Balance feels good, allows for different strategies”

“I think it was decently balanced for what it is so far”

Some testers suggested features which were implemented into the second playtest iteration:

“The Rules. They could have an illustrated example for each rule set. It helps understanding probably. Maybe also a little chart like : Fey>Gate>Fiend>Celes>Element > Color>Gate [sic]”

“I would like to see which round i am on and possibly what you betted on.”

“I would like scores on the side of the screen telling you, what round it is, how many tricks you have right now, and how many you clicked in the beginning. Just to keep track of all that.”

6.3. Second Playtest Iteration

With the second playtest iteration, the goal was to find the last lurking bugs, but also get more feedback on the functional game. For this test, two groups have been organised: one group contained the testers of the first playtest iteration, while the other contained new testers that have not seen the game before. Regarding bugs, most noticed bugs have been of a visual nature (see Appendix D for full data). Over 60% of playtesters reported bugs are grounded in UI issues, while less than 30% are gameplay bugs (Fig. 10).

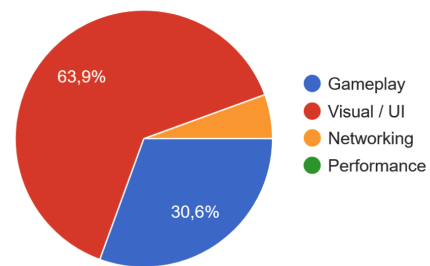


Figure 10. Diagram categorising reported bugs of the second testing iteration

The reported gameplay bugs boil down to one big issue: the point manager. A tester described the issue as follows:

“The point calculation is weird... If it is right (in some rounds) it is right for everyone, regardless of lose or win. if it is wrong, it is wrong for everyone equally. Sometimes more, sometimes less!”

This bug has not been able to be located within the code base in the time frame provided. Moreover, it was not prioritised, but will be handled in future iterations of the game. Other issues as mentioned are mainly visual in nature, including responsiveness of the game on smaller screens, centering of UI texts or general visual complaints about the art.

A small amount of Networking issues were reported, including issues when joining in the middle of the game, as the game seemingly breaks. This was also a problem if you started with only one player. These bugs have been known, but not been worked on so far, as they do not impede the testing of the networking or game prototype itself. It was not mandatory for the frame of this paper and due to time restraints, they have been taken out of consideration.

To prove the success of the functionality of the game, none of the bugs from the first playtest iteration persisted in the second iteration. Therefore, the iterative process provided significant additional value.

Second Playtest Feedback

With the second test iteration, not only did early testers get the chance to see the improvements in the game, but also see some added features that were suggested in their gameplay reports. The testers of the first iteration have been excited over the changes, and described themselves as “lik[ing] the new things added!”, they reported the following:

“The info chart on the side is really good!”

“The new visuals are a huge improvement”

“The cheat sheet helps a lot with the flow of the game, having easy access to the more abstract rules makes it much easier to play. The scroll function moving faster makes it much easier to see your whole hand of cards, as well as scroll through the rules.”

Most new testers considered the game enjoyable as well, although players in this group noted the weaknesses of the limited number of players. A pair of players considered the game as follows:

“Not fun, didn't feel like i was in control of anything. [sic] Very much based on luck on what cards you get and on luck of predicting what the other player has.”

“As a player in a 1v1 setting, the most fun we had came from the conversation about how unpredictable it is and if there is even a strategy with the current implementation and rules. So the multiplayer aspect definitely was the fun source in a 1v1 environment.”

“No, in the 1v1 situation we had it was more then [sic] likely in the first couple of rounds and later rounds that the person who goes first wins easily because of the color rules [sic] [...]”

The game has been made with the intention of enabling a group of players to play, rather than two or fewer. Though it is playable with only a pair, the experience and balance are not at their fullest. Future iterations need to consider this weakness and disbalance. Thanks to this feedback, design concepts for two-player games can be taken into consideration for future iterations.

In summary, most players enjoyed the game, but mentioned an impeded experience due to bugs. Many bugs have been found thanks to testers, and shortened the development time due to the delegation of bug search to testers. With more time in the future, visual issues and more features can improve the game feel thanks to feedback and suggestions given to the players.

7. Conclusion

The objectives raised in the introduction chapter of this research can be concluded as successful according to the feedback provided by the playtesters, and is furthermore reinforced by the recommendations provided in the following chapter. Despite the overall success, certain limitations remain which will be addressed below and in chapter nine.

Playability and functionality. The prototype made with Unity Netcode in its final state is functional, filling the gap through a well constructed and debugged structure. Additionally, the documentation and production workflow of the game with agile methodologies provide valuable insight for the industry.

Enjoyment. The game and its design, its rules and special cards are enjoyable and unique, as testers have reported in feedback. As presented in the first iteration feedback, the interest in the game as well as its replayability have been proven. While the majority of the testers reported positive and 'fun' emotions during the playtesting experience, a distinct limitation remains when a lower number of players are in the game session.

Architecture

The programming and system architecture of Obsidia is providing the worldwide playability of the game and is effective and functional. However, when addressing future improvements to the system, there is a noteworthy improvement to be made in regards to performance.

Instead of spawning in PlayerHands, it is beneficial to have premade PlayerHands in the scene, which are then assigned to clients when the game starts. Though this would mean a set amount of objects with most likely more objects than players in the scene and extra effort to assign clients to their PlayerHands, it avoids the issue of having to find references to other scripts like the deck within the scene in the start method, therefore, being more performance friendly. Said references can be set up before the program starts, avoiding a lot of processing power.

First Playtest

With an average of five to ten played rounds per tester, a sufficient overview has been gathered of the game during the first playtest. The test iterations have provided grand insight into the dysfunctions and bugs of the game. The reports given by the play-testers have pointed out issues, and with that shortened development time.

The overall feedback for Obsidia was great as well as provided inspiration and additional suggestions for the second iteration. The feedback proved replayability and long-term enjoyment that kept players interested. Moreover, the reports prove that

the design and concept of the game are sufficient, captivating and fulfilling their aim to create a fun experience.

Second Playtest

To prove the success of the functionality of the game, none of the bugs from the first playtest iteration persisted in the second iteration. Therefore, the iterative process provided significant additional value. Moreover, the playtesters' suggestions for additional features were implemented to the game for the second iteration, which improved the enjoyability of the game according to the feedback provided.

During the second playtest, a noteworthy weakness was identified whereby participants reported reduced enjoyment of the game when played with a small number of players. The game's primary objective is to allow a group of players to participate, rather than two or fewer individuals. However, the game's balance and overall experience are not optimal when played with a small group. This feedback suggests that future iterations of the game should address this issue. To improve the game, design concepts for two-player games could be considered in future iterations.

An additional weakness shown during testing were larger bugs such as the point manager are still within the prototype. Since time was limited and since the goals of this paper were not impeded by this bug, it was not prioritised.

In summary, most players enjoyed the game, but mentioned an impeded experience due to bugs. Many bugs have been found thanks to testers, and shortened the development time due to the delegation of bug search to testers. With more time in the future, visual issues and more features can improve the game feel thanks to feedback and suggestions given to the players.

Unity Netcode

Unity Netcode has shown both its strengths and weaknesses. Due to its lack of resources and recent release, it provides a tool of great potential with current weaknesses. It is reliable and stable in its connection and network functionality, as well as easy to learn for beginners in the field of network programming. Its greatest weakness is the lack of testing in released games and prototypes that will change over the following years. Future research, development and use of the library will show its comparison to similar tools on the market.

8. Recommendations

Overall, this game and paper have the potential of raising awareness of the complexity of creating games, specifically multiplayer games. It provides a code example and resource for developers intending to utilise Unity and Unity Netcode.

8.1. Guidelines

When working with Unity Netcode or any networking library, framework or system, one has to always create the structure of the game or program first. UML diagrams can be useful for this purpose. One should always start with the networking aspect and not with the game itself, otherwise subsequent changes can become complicated and may require a complete change of the implemented systems.

Furthermore, decisions of what elements of the code should be on the server and what on the client side should be made as early as possible, preferably in the preproduction and research or production phase. This counts specifically for data and variables, as their storage needs to be specified, as well as who they get shared with and who can modify them. A variable that needs to be available for everyone should be stored on the server as an example, while player-specific information might want to be stored on the client side instead. If the variable needs to be reachable, one might want to create a manager for it on the server side instead. And should it be modifiable, only the owning client, the server or every client should be able to change it. Aspects like this need to be taken into consideration when creating the structure of the game.

Sharing data is also important. One has to decide early what variable types are needed in the project, as not all can be easily sent in the network, such as Lists or Scriptable Objects. Another option is the creation of JSON files, for transport and storage of larger information, which have to be planned before implementation.

And lastly, testing new features and their access to the clients is of great importance. Should a client have access to the wrong methods, the entire game could break. Ensuring that only the server has access to important functionality that the client does not need to know of.

8.2. Future Work

The end of the project does not mark the end of Obsidia. Several new features have been planned for the future of the game.

With the feedback provided in both test iterations, the first aim will be to fix the remaining bugs within the game, mainly related to gameplay and visual aspects. Afterwards, an overhaul of the current structure and architecture of the game is

planned for better performance, including the changes of PlayerHands that have been mentioned in the conclusion. The new structure will incorporate several of the suggested features from testers.

Although steps were taken in design to reduce the imbalance, the game would benefit from additional testing to assess the power balance of the design of the game in depth.

Future research on this project can involve the implementation of different networking systems, such as a lobby system. Additional game modes with different decks and changed rules as well as customisation of the game combined with more networking functionality can be considered by future researchers.

This project has not only enhanced my game development skills, but also demonstrated the potential of contemporary technology in game development. It has served as a testament to my abilities and propelled my personal journey. The knowledge and experience acquired through this project serve as a catalyst for future game creation and ensure success in the game industry.

9. Limitations

Time constraints. Game development takes time, as every prototype requires research, development and testing. Due to only 16 weeks, not all aspects could be covered. Some bugs as reported by testers were unable to be fixed due to priorities as mentioned in the playtest iteration points of this paper. Several non mandatory features needed to be cut for this reason. The same counts for bugs that have not been impeding this paper's goals.

Limited resources. The author of this paper needed to consider the prototype as well as the paper. Limited resources can hinder the progress and quality of research and prototype development. A singular developer may face time constraints, limited scope, lower quality outcomes, and limited expertise. Collaborating with others and seeking additional resources can help overcome these limitations.

Lack of previous research studies on the topic. The lack of academic research on Unity Netcode makes it challenging to identify its capabilities and limitations, hindering game development. Without comparative studies, it's difficult to determine its strengths and weaknesses in comparison to other networking solutions.

Addressing this gap by conducting more research studies can provide valuable insights into the capabilities and limitations of the library, which can help game developers maximise its potential. More research is needed to provide insights into the library's functionality and scalability, facilitating informed decision-making.

Tester biases. When testing a prototype with a third party, there are several limitations to consider that can impact the quality of feedback and the overall effectiveness of the testing process. One of the primary limitations is the experience of the testers. Testers who have prior experience with similar games or genres may have preconceived biases that can influence their opinions and feedback. This can lead to a skewed perspective on the prototype's potential appeal to a wider audience.

Sample size. The testing approach utilised a relatively small number of testers, which could impede the ability to identify all potential issues with the prototype and fully address the experience's enjoyability aspect. The involvement of a larger and more diverse group of testers would have yielded a wider range of feedback, thereby enhancing the identification of areas for improvement that may have been overlooked with a smaller sample size.

Sampling method. Convenience sampling, a non-probability sampling approach, was utilised which may limit the generalisability of the findings. The chosen sampling method may also result in the over- or under-representation of certain groups, leading to skewed results. Moreover, convenience sampling may not ensure that all members of the population have an equal chance of being selected, which may further compound sampling bias.

10. Sources

- [1] Unity and Armstrong, L. (2023) *Netcode 1.0.0*. (Accessed: March 9, 2023)
<https://docs-multiplayer.unity3d.com/releases/netcode/1.0.0/index.html>
- [2] Innersloth (2018) *Among Us, Steam Page*. (Accessed: March 9, 2023).
https://store.steampowered.com/app/945360/Among_Us/
- [3] Unity (2021) *Among Us and Unity | Unity Gaming Services*. YouTube. (Accessed: March 9, 2023).
<https://www.youtube.com/watch?v=wrjmxIRbyt0>
- [4] Unity (no date) *How Among Us connects 500+ million players , Among Us | Unity Case Study*. (Accessed: March 9, 2023).
<https://unity.com/case-study/innersloth-among-us>
- [5] Euroskat (no date) *Ein Skat Stich tut nicht weh!* (Accessed: December 5, 2022).
<https://www.euroskat.com/info/skat-stich>
- [6] Brettspielwelt (no date) *Skull King - Anleitung*. (Accessed: December 5, 2022).
<https://www.brettspielwelt.de/Hilfe/Anleitungen/SkullKing/>
- [7] AMIGO Spiel + Freizeit GmbH, (no date) *Wizard Spielanleitung*. (Accessed: December 5, 2022).
https://www.brettspiele-report.de/images/wizard/Spielanleitung_Wizard.pdf
- [8] Marc Prensky (2001) "Chapter 5", from McGraw-Hill (ed.) *Digital Game-Based Learning | Fun, Play and Games: What Makes Games Engaging*. (Accessed: March 9, 2022).
<https://marcprensky.com/writing/Prensky%20-%20Digital%20Game-Based%20Learning-Ch5.pdf>
- [9] Ramadan, R. and Widayani, Y. (2013) *Game development life cycle guidelines*. (Accessed: March 3, 2023).
<https://jiharu.github.io/lowpoly/img/06761558GameDev.pdf>
- [10] Al-azawi , R., Ayesha, A. and Obaidy, M.A. (2014) *Towards Agent-based Agile Approach for Game Development Methodology*. (Accessed: March 3, 2023).
https://www.researchgate.net/profile/Rula-Al-Azawi/publication/286562348_Towards_Agent-based_Agile_approach_for_Game_Development_Methodology/links/56c5612008aea564e304e5d8/Towards-Agent-based-Agile-approach-for-Game-Development-Methodology.pdf
- [11] Kristiadi, D.P. et al. (2019) *Game Development with Scrum methodology*. (Accessed: March 3, 2023).
<https://ieeexplore.ieee.org/abstract/document/9144963>
- [12] Reyno , E.M. and Cubel, J. (2008) *Model Driven Game Development: 2D Platform Game Prototyping*. (Accessed: March 3, 2023).

https://www.researchgate.net/profile/Jose-Carsi/publication/221024315_Model_Driven_Game_Development_2D_Platform_Game_Prototyping/links/0deec52bc129d3e60e000000/Model-Driven-Game-Development-2D-Platform-Game-Prototyping.pdf

[13] Unity (no date) *Networking and Netcode Software Solution* | Unity. (Accessed: March 3, 2023). <https://unity.com/products/netcode>

[14] Unity, *Unity Real-Time Development Platform* | 3D, 2D, VR & AR Engine. (Accessed: March 10, 2023). <https://unity.com/>

[15] Google, *Google Drive Homepage*. (Accessed: March 10, 2023). <https://drive.google.com/>

[16] Google, *Google Documents Homepage*. (Accessed: March 10, 2023). <https://docs.google.com/document/>

[17] Google, *Google Forms Homepage*. (Accessed: March 10, 2023). <https://docs.google.com/forms/>

[18] Wikipedia (2023) *Google Drive*. (Accessed: February 2, 2023). https://en.wikipedia.org/wiki/Google_Drive

[19] Github (2023) *Erste Schritte mit Github - GitHub Docs*. (Accessed: March 1, 2023). <https://docs.github.com/>

[20] Wikipedia (2023) *Github*. (Accessed: February 21, 2023). <https://en.wikipedia.org/wiki/GitHub>

[21] VeriorPies (2022) *ParrelSync - (Unity3D) Test multiplayer without building*. (Accessed: February 20, 2023). <https://github.com/VeriorPies/ParrelSync>

[22] Trello, *Trello Homepage*. (Accessed: March 3, 2023). <https://trello.com/>

[23] Wikipedia (2023) *Trello*. (Accessed: February 21, 2023). <https://en.wikipedia.org/wiki/Trello>

[24] Dummett, M. (1993). *The history of card games*. (Accessed: March 3, 2023). <https://www.cambridge.org/core/journals/european-review/article/abs/history-of-card-games/E9386F7675C8D9CA283502A888D46BF8>

[25] Ubisoft (no date) *UNO* | Ubisoft. (Accessed: March 3, 2023). <https://www.ubisoft.com/en-au/game/uno/uno>

[26] AMIGO Spiel + Freizeit GmbH, (no date) *Wizard Jubiläumsedition*. (Accessed: March 3, 2023). <https://gzhls.at/blob/ldb/e/c/6/5/1704fb20da04dce5caf0222e288dad8cf78f.pdf>

[27] Unity (2018) *Unity- Manual: ScriptableObject*. (Accessed: March 3, 2023). <https://docs.unity3d.com/Manual/class-ScriptableObject.html>

- [28] Gamma et al. (1998) *Design Patterns Elements of Reusable Object-Oriented Software*. (Accessed: March 3, 2023).
<http://www.javier8a.com/itc/bd1/articulo.pdf>
- [29] Refactoring Guru (no date) *State*. (Accessed: March 3, 2023).
<https://refactoring.guru/design-patterns/state>
- [30] iHeartGameDev (2021) *How to Program in Unity: State Machines Explained*. YouTube. (Accessed: March 3, 2023).
<https://www.youtube.com/watch?v=Vt8aZDPzRjI>
- [31] Wikipedia (2023) *Binary search algorithm*. (Accessed: February 27, 2023).
https://en.wikipedia.org/wiki/Binary_search_algorithm
- [32] Wikipedia (2023) *SOLID*. (Accessed: February 27, 2023).
<https://en.wikipedia.org/wiki/SOLID>
- [33] Unity and Armstrong, L. (2023) *Relay | Unity Multiplayer Networking*. (Accessed: March 1, 2023).
<https://docs-multiplayer.unity3d.com/netcode/current/relay/relay/index.html>
- [34] Unity (no date) *Unity Relay*. (Accessed: March 1, 2023).
<https://docs.unity.com/relay/en/manual/introduction>
- [35] Wikipedia (2023) *Proxy server*. (Accessed: March 1, 2023).
https://en.wikipedia.org/wiki/Proxy_server
- [36] Unity (no date) *Relay Servers*. (Accessed: March 1, 2023).
<https://docs.unity.com/relay/en/manual/relay-servers>
- [37] Unity (no date) *Allocation service*. (Accessed: March 1, 2023).
<https://docs.unity.com/relay/en/manual/allocations-service>
- [38] Unity and Armstrong, L. (2023) *NetworkObject*. (Accessed: March 1, 2023).
<https://docs-multiplayer.unity3d.com/netcode/current/basics/networkobject/index.html>
- [39] alexandremoraes (2015) *Inheritance issue using UNET NetworkBehaviour*. Stackoverflow. (Accessed: March 3, 2023).
<https://stackoverflow.com/questions/30871494/inheritance-issue-using-unet-networkbehaviour>
- [40] biboserge (2019) *Inheriting from a class that inherits from NetworkBehaviour*. (Accessed: March 3, 2023).
<https://answers.unity.com/questions/1619919/inheriting-from-a-class-that-inherits-from-network.html>
- [41] *Reference for author's inspiration for the game*. (Accessed: March 7, 2023).
<https://www.youtube.com/watch?v=dQw4w9WgXcQ>

11. Appendices

Appendix A - Prototype

Link to the Github Repository:

https://github.com/LiloTM/BachelorThesis_Obsidia.git

Visual assets for cards

<https://assetstore.unity.com/packages/2d/gui/card-shirts-lite-165698#content>

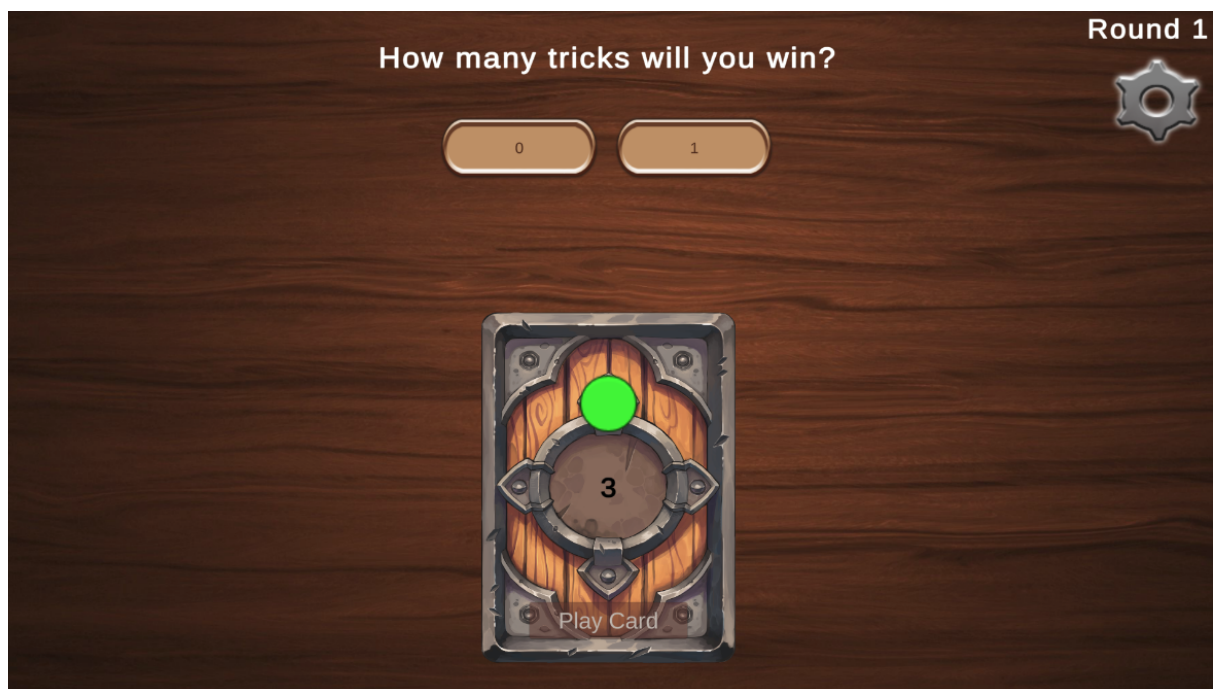
Background asset

<https://www.pinterest.co.uk/pin/730990583241462378/>

UI assets

<https://assetstore.unity.com/packages/2d/gui/icons/fantasy-free-gui-147682>

Screenshot of Obsidia:



Appendix B - Questionnaire

Bug Report Form

Please report any bug encountered during your play-throughs.
Be as direct and clear as possible.

Please categorise your bug:

- Gameplay
- Visual / UI
- Networking
- Performance

Please rate its severity:

- Gamebreaking
- Major
- Minor
- Trivial

Please explain the bug.

Textfield

How did this issue impact you?

Textfield

Can you replicate it?

Yes/No

If replicable, please explain how.

Textfield

(Optional) Please enter your contact details in case of further questions:

Textfield

Please provide a screenshot or short video, if applicable.

upload file

Gameplay Report Form

Don't shy away from going into detail. Write multiple sentences for each question if needed

1. How much time were you playing for? Alternatively, how many rounds have you played?
2. How was the overall player experience?
3. What did you like and dislike about the game?
4. Did you encounter any issues or bugs while playing? (YES/NO)
5. Was the game well-balanced, in terms of fairness?
6. Did you feel the game was too easy or too difficult in certain areas? Was there any aspect of the game that felt repetitive or boring?
7. Was there anything that felt confusing or unclear in the game?
8. Was there anything you felt was missing from the game?
9. What features would you add or change to improve the game?
10. Do you have any other comments?

Appendix C - Game structure

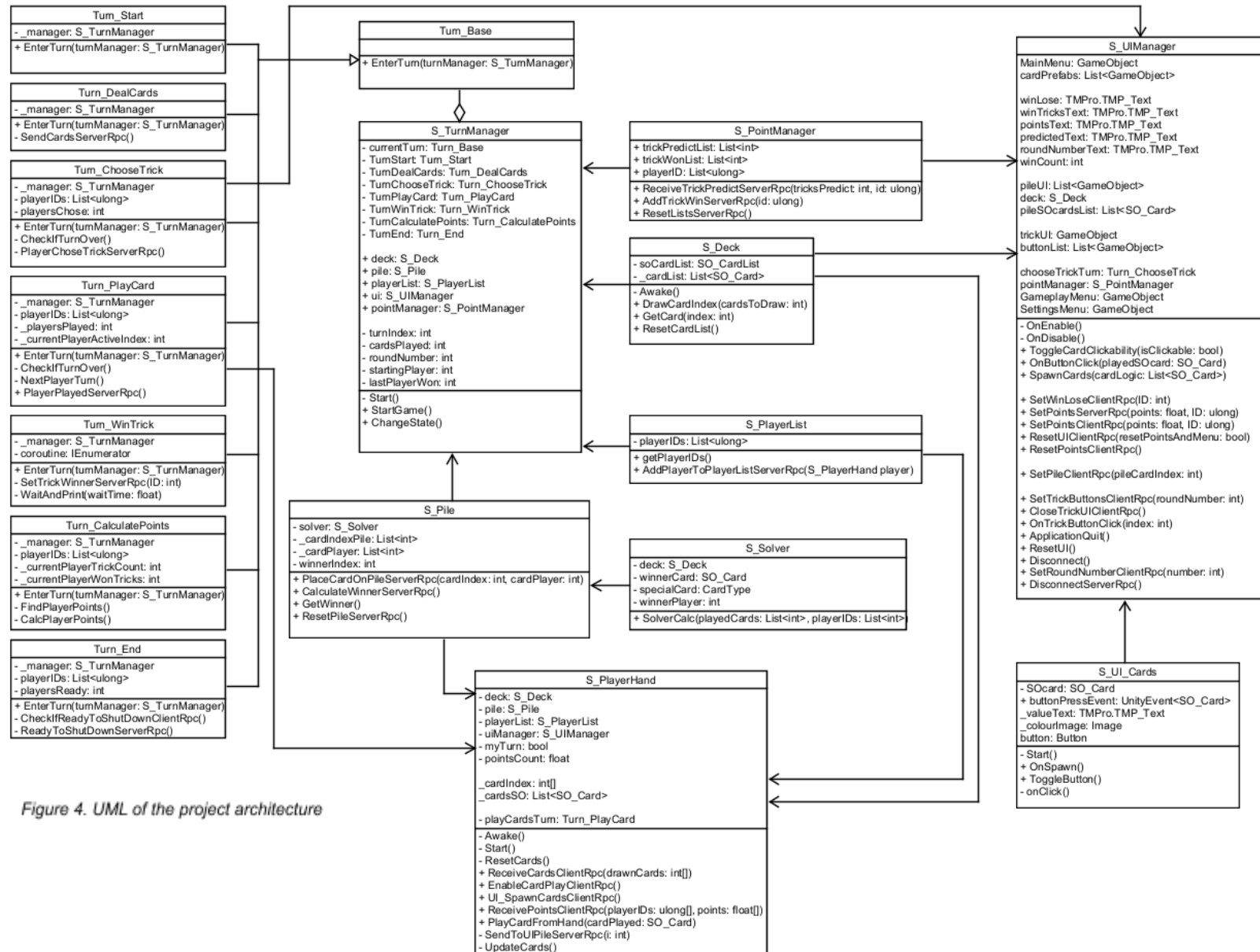


Figure 4. UML of the project architecture

Zeitstempel	Please categorise your bug.	Please rate its severity.	Please explain the bug.	How did this issue impact you?	Can you replicate it?	If replicable, please explain how.	(Optional) Please enter your contact details in case of further questions.
2023/02/04 9:02:44 PM OEZ	Gameplay	Major	The second card wins regardless of colour or number if it is a different colour from the first one played and the last card played is the same colour as the first that is higher.	Win's are technically random	Yes	As explained above.	
2023/02/04 9:04:05 PM OEZ	Visual / UI	Major	The EXIT button doesn't work	I cannot close the game except forcing it with the X int he top right corner	Yes	It's always there.	
2023/02/04 9:06:01 PM OEZ	Visual / UI	Trivial	The bottom part of the cards is cut off	Not at all	Yes	It's always there.	
2023/02/04 9:07:05 PM OEZ	Visual / UI	Minor	Scrolling via mousewheel through the rules or cards is very slow.	Not a lot, since you can drag instead of scroll.	Yes	It's always there.	
2023/02/04 9:08:27 PM OEZ	Visual / UI	Major	Leave button on the main menu does not function	Unable to exit game within the game's UI	Yes		
2023/02/04 9:09:54 PM OEZ	Visual / UI	Trivial	The points are black on a brown background, so it's not that easy to see.	Very minorly	Yes	It's always there.	
2023/02/04 9:12:55 PM OEZ	Gameplay	Gamebreaking	If a higher number of the serving colour gets played than the first one after a different colour in between, the different colour wins.	It's gamebreaking	Yes	By purposefully putting "low serving-not serving-high serving"	
2023/02/04 9:13:27 PM OEZ	Gameplay	Major	If a different coloured card is played after the first card, and then a card that should beat the first card is played, the different colour will win		Yes	As above	
2023/02/07 1:37:53 AM OEZ	Visual / UI	Minor	Using the mouse wheel to travers the rolls or your hand of cards is tedious to the point of useless. Sensitivity should be increased.		Yes		
2023/02/08 11:24:27 AM OEZ	Gameplay	Gamebreaking	The point manager only works for the host, the clients' points are all the same.	It doesn't give accurate results.	Yes		
2023/02/08 11:26:54 AM OEZ	Visual / UI	Minor	On the ready screen, the words are overlapping with the button	It doesn't look too clean	Yes		
2023/02/08 11:44:18 AM OEZ	Gameplay	Gamebreaking	If a fey is played first, and a fiend is played in the round, the fey still wins. Applies to [fey, colour, fiend], as well as [fey, fiend, colour]. Not tested with other special cards.	It breaks the game	Yes	Play fey first, and then fiend somewhere in the same round.	
2023/02/08 1:14:02 PM OEZ	Gameplay	Major	The leave button in the main menu is not functional	Not that much, the window is closable but it should work	Yes	Closed and opened the game again, still didn't work	this is Eve, feyishVen on discord
2023/02/08 1:16:16 PM OEZ	Gameplay	Minor	The leave button wont let me leave the game	Its okay I just have to press the X	Yes	I'm stuck	
2023/02/08 1:33:25 PM OEZ	Performance	Major	Cannot leave the game	... erm... I couldn't quit the game with the button	Yes	I restarted the game several times, it persists	
2023/02/08 1:36:01 PM OEZ	Gameplay	Gamebreaking	Would have won, had the higher number with the correct colour but still lost	Well, I did not gain points and lost even tho I would have won	Yes		
2023/02/08 1:44:16 PM OEZ	Gameplay	Major	When you have a colour with a number and another person plays the same number in a different colour and then the third has the same colour as the first and has a higher number, the second with the different colour wins	The wrong person wins	Yes	tried the same with different colours	this is eve again
2023/02/08 2:13:44 PM OEZ	Gameplay	Gamebreaking	When a higher card of the same color of the trump is played after any other color, it doesn't count it as a win.	Let's me misjudge the winning possibility and lose the game	Yes	We played several rounds and coordinated our plays so that we can replicate it.	
2023/02/08 2:15:43 PM OEZ	Performance	Trivial	Scrolling with the mouse is way too slow. Maybe it needs some "speed multiplier"?	It is just a little annoying	Yes	You just need to enter the rules window	

2023/02/08 2:18:23 PM OEZ	Gameplay	Major	When the 1st player places a gate, and the second one a color (virtually becoming a trump), the last player to play a higher of the same trump color loses regardless.	It makes winning impossible	Yes	We coordinated our play to replicate this, it worked	
2023/02/08 2:21:14 PM OEZ	Gameplay	Major	When a color card is played first, setting a trump, and a fiend is played next, even if the following card is of the higher value and of the trump color, if loses.	It makes winning impossible and is not consistent with the rules, as it never says that fiends trump colors.	Yes	We coordinated our play over discord to replicate this.	
2023/02/08 2:31:51 PM OEZ	Networking	Gamebreaking	It is possible to join ongoing games and totally break the game	I joined the game, so it did not affect me much, but the others. Their game got reset to my round, but still had the card values from theirs.	Yes	We just played many round and joined the round already started	
2023/02/08 2:55:48 PM OEZ	Networking	Major	As a host one can start a game round totally alone and wins it every time	Well... It's not a real game if you dont have other players in this one.	Yes	Simply join a network game and be the host, click ready and start playing.	
2023/02/08 3:35:52 PM OEZ	Gameplay	Gamebreaking	There are various bugs concerning precedence of rules and who wins. Sometimes higher card values win of wrong color, sometimes lower values win after gates. See the screenshots. Some have descriptions in them and others just the states, from one can deduce the mistake hopefully	It makes turns hard to predict	Yes	Just play enough	
2023/02/08 3:36:55 PM OEZ	Gameplay	Gamebreaking	There are various bugs concerning precedence of rules and who wins. Sometimes higher card values win of wrong color, sometimes lower values win after gates. See the screenshots. Some have descriptions in them and others just the states, from one can deduce the mistake hopefully	Makes the game unpredictable	Yes	Just play it a few times	
2023/02/09 9:09:17 PM OEZ	Gameplay	Gamebreaking	the point's dont seam to work	you cant really play the game if you can see who wins	Yes	play the game	Hex
2023/02/09 9:11:49 PM OEZ	Gameplay	Gamebreaking	the colord cards some time dont work as it suposte to. i win and loos when it soude be the oposite	hard to play the game when you winn and lose randomly	No	-	

Zeitstempel	How much time were you playing for? Alternatively, how many rounds have you played?	How was the overall player experience?	What did you like and dislike about the game?	Did you encounter any issues or bugs while playing?	Was the game well-balanced? Was it fair?	Did you feel the game was too easy or too difficult in certain areas? Was there any aspect of the game that felt repetitive or boring?	Was there anything that felt confusing or unclear in the game?	Was there anything you felt was missing from the game?	What features would you add or change to improve the game?	Do you have any additional remarks or observations to share
2023/02/07 1:37:25 AM OEZ	~approx 2 hours	Overall good, with some issues like bugs and UI.	Simple ruleset with fast paced gameplay.	a few (impeding your experience)	Balance feels good, allows for different strategies	N/A	Certain rules take some time to adjust to, an optional rule card/visual during the game could solve this for newer players	When creating a game, there is no display of how many players are connected, making it difficult to see if there are any people struggling	Reduce the size of cards in game, to make it easier to see your whole hand on one screen.	Game is very harsh on the GPU. Hopefully to be fixed in the future, but still ran very well on my pc.
2023/02/08 2:12:17 PM OEZ		10 Was quite nice to explore	I like the design and the Idea, just had gameplay bug issues	a few (impeding your experience)	It is fair when the bugs are fixed :D	Neither to easy or to difficult, I can really imagine playing this more often when its finished	Had some issues with the rules about the special cards like fey, but maybe thats just my brain	no	I would add Player names to see who plays a card and who wins, also the scrolling takes ages, to see what your trick bet was would be also nice	The colour could be put on the screws on the Cards as different shaped gems, would look more elegant and would be nice for people who are colour-blind
2023/02/08 3:19:22 PM OEZ	we played for about 2 hours and played 2 complete and 3 stopped rounds	Really fun! Once we sorted out the bugs that we knew, I enjoyed my time playing the tactics of the game.	Please fix the scrolling speed on the rules, it makes it very annoying to scroll through them. I really liked the strategy of loosing on purpose to get the exact trick, which is core to the game. I liked playing off the different cards to achieve your desired goals.	a few (impeding your experience)	since the rounds with the special cards are unwinable with colour cards, it can really screw someone over, especially if one person gets many specials.	It is difficult to remember what the special cards do and how they work, mostly because they are more than one.	Where do I see who is in the game? If I'm the host, I would like to be sure that everyone has entered the game before starting. The numbers for the tricks its hard to get what they mean at the start, please add at least some text that indicates what they are No one knows who is currently able to play. You don't need to show everyone, but at least let the person that it's their turn. I was expecting that only the host could start the game? but	I would like to see which round i am on and possibly what you betted on. As well as, it is easy to get confused about who's card has been played. It would be good if there was an option to change your tip before the round starts, maybe an extra button to state that you are ready for the round. Once you click it you can't change the tip	Maybe options for a shorter game to get used to the rules. It was way easier to understand the rules after I played it but 10 rounds is really long.	
2023/02/08 4:57:00 PM OEZ	2,5h - about 10 rounds	rather good	I liked the general artstyle. I'm not sure about the buttons for placing bets. They are a bit to wide. Maybe use a slider and input field combination instead? or a button set for + and - values	a few (impeding your experience)	Generally, it is ok but I feel that there are a bit too many specials. Or they should be a bit more tied together. The Gate serves as a Joker in "Wizard", but isn't reliable much to predict it's value, as the Fey changes it. Celestials, Elementals and fiends work good with each other, as they all can trup one another. But Fey and Gate somehow break that balance. Either reduce the specials to 3 maybe (anything nore can be rather complicated to	The difficulty overall is ok (especially once the mechanics are more solid). It is repetitive by nature, but still engaging enough. Sometimes keeping track of which card trumps what is a bit confusing...	The Rules. They could have an illustrated example for each rule set. It helps understanding probably. Maybe also a little chart like : Fey+Gate>Fiend>Celestials>Elementals>Gate	Sounds, Menues for a few settings: like resolution and volume (if sound was added)	Most are UI or UX related Visual cues of play order. Here I would also remove (make them completely invisible) the "play card" buttons when it is NOT a players turn. Additionally reduce the cards saturation down to 50% (gray them out), maybe. Instead of the "Play Card" button, make them completely clickable? When the mouse hovers over a certain card, it could "hover up" and "zoom in" to	Did you use the Wizard-Special edition as an inspiration? It has similar "special cards" with flipping and changing values. I have all the editions of that game series and, the most basic one was still the most engaging. But it is still a great game example and I can imagine it will be a Gwent for D&D players soon. It just needs a catchy name ;)
2023/02/09 7:45:59 PM OEZ	10 rounds	i liked the game, interesting concept, decent execution	I liked the concept and rules, i dislike the lack of a working leave button and no win screen	very little (barely impeding your experience)	I think it was decently balanced for what it is so far	Rules are decently easy to follow, and there is some good planning ahead for this game	i didnt feel that anything was unclear	maybe an optional tab for rules to check during the game?	Working leave button, win screen scores displayed	no

2023/02/09 9:06:19 PM OZ	2	ok, but really scuft	i like the core game. but all the UI is really buggy and would need to be better for the game to really be funn	a lot (heavily impeding your experience)	i only get to test with 1v1 so as far as that is ok.	did not get to test to much about it	how to leave. who is starting the trune. how the point's work.	yes. you can't see your bett and trics of the round. you can't see when people join the host, you cant see who's turn it is. you cant see the oder playres bets. you cant see oder playres point you can see who win.	the scrool weel is soo slow it pointtest to use. having a reminder of the rules in the game what beats what	fix the UI
2023/02/09 9:18:07 PM OZ	4 rounds	Mostly fun.	Easy to understand, casual and a good party game. No particular dislikes. Only a few quality of life issues. (Slow scrolling, resolution issues)	a few (impeding your experience)	Mostly yes. Only a few bugs that got in the way sometimes.	The rounds can drag towards the end.	No.	A lobby to see what players are ready or have joined the game.	No particular suggestion.	No.
2023/02/13 3:38:20 PM OZ	About seven or eight or so? I played three different times in the first test round.	It was fun to figure out the bugs	It was for sure still a bit rough so i can't really give a proper assessment about that. I liked the lore aspects of it, why certain cards were beating others, and their names.	a few (impeding your experience)	I would say so; we will see once we play bug free.	I haven't made an actual attempt at winning yet because of the score bug, so I can't speak to the difficulty.	No, once you get a hang of the rules, it went well. It took a round or two though, because there are quite a few.	I would like scores on the side of the screen telling you, what round it is, how many tricks you have right now, and how many you clicked in the beginning. Just to keep track of all that.	See above; additionally, an exit button during the game round would be good, so you don't have to close and re-open the application.	I'm looking forward to the next rounds of testing.

Zeitstempel	Please categorise your bug.	Please rate its severity.	Please explain the bug.	How did this issue impact you?	Can you replicate it?	If replicable, please explain how.	(Optional) Please enter your contact details in case of further questions.
2023/02/15 4:06:20 PM OEZ	Visual / UI	Minor	Under Special Cards in the Rule menu, the text overlaps for a few words	Cant read	Yes		
2023/02/15 4:32:36 PM OEZ	Visual / UI	Major	When you rescale the window, the cards & buttons stay the same size.	Not really since my screen is big enough, but once the prediction buttons get to many, they cover the score	Yes	Resize the Screen, and play until at least round 5 or six	This is Eve
2023/02/15 5:22:03 PM OEZ	Gameplay	Major	When the round is an even number everyone gets a wrong score	I didn't get the correct score for my game	Yes	Its difficult, I don't know how the score gets calculated wrongly but it happens on the even rounds	Its Eve
2023/02/15 5:47:04 PM OEZ	Visual / UI	Trivial	why is the leave button called "leave" and not "exit"	it confused me	Yes		Liz
2023/02/15 5:48:45 PM OEZ	Visual / UI	Major	The buttons to select the prediction for each round were at one point overlayed over the points, so you couldn't see how much points you had.		Yes	Play the game until approx. round 5 or 6, then it appears on its own.	
2023/02/15 5:49:05 PM OEZ	Visual / UI	Minor	there should be a tiny bit of space between the text and the text box in the how to play section	look weird and made it a bit harder to read imo	Yes		Liz
2023/02/15 5:49:36 PM OEZ	Gameplay	Gamebreaking	The point system didn't always work as it was supposed to, most of the time points were given at random instead of how they were supposed to be given.		No		
2023/02/15 5:50:07 PM OEZ	Gameplay	Major	there is no way to leave a running game, you have to close and reopen the whole window		Yes		Liz
2023/02/15 5:52:00 PM OEZ	Visual / UI	Minor	the client code window is directly over the client button for some reason? it would make more sense if you had to click the client button and then a new window opens where you have to insert the code and then press submit	it look kinda clunky and it was confusing to me what i have to press	Yes		Liz
2023/02/15 5:54:14 PM OEZ	Gameplay	Minor	it would be better if i still could change my predictions while i'm waiting for everyone else to give their predictions	i sometimes accidentally clicked one prediction and then immediately realized i would like to choose something else but then i couldn't change anymore	Yes		Liz
2023/02/15 5:55:53 PM OEZ	Visual / UI	Minor	the played cards look kinda squished compared to the others	it was just weird to look at	Yes		Liz
2023/02/15 5:57:43 PM OEZ	Visual / UI	Major	moving the cards around with the mouse wheel results in very jagged weird movement	it was annoying and confusing	Yes		Liz
2023/02/15 5:59:48 PM OEZ	Visual / UI	Trivial	the gear button gets squished if the window is resized	not much tbh i just noticed it	Yes	simply rescale the window	Liz
2023/02/15 6:00:52 PM OEZ	Visual / UI	Minor	the colored circle on the cards look VERY low poly	it just ugly tbh	Yes		Liz
2023/02/15 6:05:43 PM OEZ	Visual / UI	Minor	the info panel has very weird edges which look cut off	it just look ugly	Yes		Liz
2023/02/15 6:07:11 PM OEZ	Visual / UI	Major	the won games counter doesn't show a 0, it just shows nothing if you haven't won yet	it was kinda confusing since for a second i thought the counter was broken	Yes		Liz
2023/02/15 6:08:21 PM OEZ	Visual / UI	Minor	it's the missing 0 counter for won games again, i just forgot to add the screenshot		Yes		Liz
2023/02/15 6:09:57 PM OEZ	Visual / UI	Trivial	maybe make the elementals/friend/celestials and the portal/fey a different color or maybe shape	it was confusing to differentiate from each other	Yes		Liz

2023/02/15 6:12:01 PM OEZ	Visual / UI	Major	Make it more obvious that it's your turn. maybe by hiding the greyed out "play card" buttons completely when it's not your turn. Or maybe make a bigger test (similar to the you loose/win) which says YOUR TURN	i often didn't even realise that it was my turn yet	Yes		Liz
2023/02/15 6:13:12 PM OEZ	Visual / UI	Minor	the you win/loose game isn't centered	it was just a bit annoying	Yes		Liz
2023/02/15 6:16:27 PM OEZ	Visual / UI	Minor	maybe show after each round how many points you got instead of only showing the absolute points		Yes		Liz
2023/02/15 6:20:09 PM OEZ	Visual / UI	Major	the way the card are displayed needs to be different (especially when it's many cards) either overlap them a bit, or display them in two rounds or something	all your cards should be visual at the same time or else it is quite difficult to guess how many wins you will have and to choose what you should play next	Yes		Liz
2023/02/15 6:21:18 PM OEZ	Gameplay	Major	show a final score and maybe a comparison of all player scores in the game	did i win, did i loose? who knows the final points never existed	Yes		Liz
2023/02/22 8:08:35 PM OEZ	Gameplay	Minor	A player can start the game even if they are the only player in the lobby. Which is an issue because you can not check if someone else has joined.	It caused my friend to play on his own.	Yes	Host lobby, press ready	
2023/02/22 8:23:12 PM OEZ	Gameplay	Gamebreaking	Screen got stuck at the final round on the you lose screen. didn't send me back to the main menu so i had to close the game and restart it.	I couldn't play anymore	No	Haven't tried yet.	
2023/02/22 8:24:06 PM OEZ	Gameplay	Gamebreaking	Couldn't press the host or client button after a game against myself. Made me unable to play another game unless i restarted the game.	Couldn't play anymore.	No		
2023/02/22 8:25:13 PM OEZ	Visual / UI	Minor	Point counter bugged out displaying the wrong points, after the round where points got added/removed it went back to normal again.	Couldn't see my actual points for a round.	No		
2023/02/23 3:29:48 PM OEZ	Visual / UI	Trivial	It's now possible to resize the window, which is great on smaller screens... but the UI elements (Buttons, Cards etc) behave wierdly. Buttons resize to sheer comical proportions until the become unreadable, while cards remain always he same size no matter how small the screen is... that makes the game almost unplayable. Resizing is ok, but the ratio should remain the same still.	Not much, it is just a visual thing. Everything still functions, even though it can be hard to see on screen.	Yes	Simply play the game in very small resolutions. Resize the window to weird proportions and you'll see it	
2023/02/23 3:35:48 PM OEZ	Visual / UI	Trivial	Starting from the 5th round the trick prediction amount picker buttons obscure parts of the surrounding UI. Probably not existing on larger screens, but still: the buttons take WAY too much width on the screen.	not much, it just obscures the points unnecessary	Yes	just play the game on 1920x1080 (my maximum size) till round 5 and later	

2023/02/23 3:46:33 PM OEZ	Visual / UI	Trivial	even on a 1920x1080 16:9 screen, many elements of the game overlap each other at a certain point (either in higher rounds or with many players) The played cards on top overlay the surrounding UI and the cards one wants to play. Some of us had super-wide screens with higher resolutions, where this problem almost never showed up (compare the screenshots). This shouldn't be bound to absolute screen sizes. All elements should scale and certain areas (where the UI is) shouldn't be used as a playable area in my opinion.	Not at all, just an observation.	Yes	Everytime I played the game, this was a thing I had to keep in mind.
2023/02/23 3:53:05 PM OEZ	Networking	Major	When one (me) joins to an ongoing round, the game for the others isn't reset to one's 1st round. Instead now the tables are flipped: the new player kinda joins the current round but only has the cards of the first round (as it is their first round) or sometimes just less than the others ... This resolves once all are "dragged" down to the same internal level, it seems. The high round players cannot effectively really continue, and the new player kinda loses any round they play until all are on the same round. sadly, I did not make any screenshots.	Well, it breaks ongoing games and even for the new person joining it isn't really fun as it is weird behavior	Yes	Just join an ongoing game.
2023/02/23 5:03:06 PM OEZ	Gameplay	Gamebreaking	The point calculation is weird... If it is right (in some rounds) it is right for everyone, regardless of lose or win. if it is wrong, it is wrong for everyone equally. Sometimes more, sometimes less! I've added extensive description in the screenshots and cut many screenshots together, as this form only allows 5 files, and we had way more than that, and it also is visually better to see them in succession.	It makes cards unreliable. If one cannot rely on the points and predictions to yield correct results, one may also just play anything without even thinking: the computer will throw seemingly arbitrary points at you at any time - so why bother trying to play correct? That's why I regard it game breaking. But it makes for a great laugh, as it is almost unpredictable and a makes for a great guessing game and social experiment (due to rage quits after the weirdness!)	Yes	Kinda, we played a few rounds just to try to find out when it occurs. We first thought it is the even rounds, or just when one loses, but... nope :P Hence the extensive descriptions in the screenshots themselves. One needs to see it, to make some sense of it. I'm still trying to figure out the mathematical mistake there :D
2023/02/23 5:05:51 PM OEZ	Gameplay	Trivial	Not a bug per se. But being thrown out the game to the main screen directly after the 10th round barely finished, is breaking the gameplay. One never sees the final points can't brag about them or find out who ever won - if at all (due to the weird point calculation)	Why would I play the game, if I never see an outcome? It IS fun in of itself, but the goal is kinda lost. And a game needs rules and goals.	Yes	just play till the end
2023/02/26 12:12:04 AM OEZ	Visual / UI	Minor	in the rules under the "Special Cards" section is some text overlapping	couldnt read it	Yes	open rules and scroll to "Special Cards" I am your Brother.
2023/02/26 12:27:32 AM OEZ	Networking	Minor	Open way many Lobbys at once	not much	Yes	You spam the "Host" button (for further information watch the attached video) I am your Brother

2023/02/28 10:13:12 PM Gameplay
OEZ

Major

Score calculations do not seem to be working correctly. As shown below, the additions and reductions are inconsistent.

0

OT W 3

OT W 4

OT W 5

OT W 9

1T W 4

1T W 12

1T W 16

1T X 8

OT W 11

OT W

Makes it very difficult to track who is winning and losing.

Zeitstempel	How much time were you playing for? Alternatively, how many rounds have you played?	How was the overall player experience?	What did you like and dislike about the game?	Did you encounter any issues or bugs while playing?	Was the game well-balanced? Was it fair?	Did you feel the game was too easy or too difficult in certain areas? Was there any aspect of the game that felt repetitive or boring?	Was there anything that felt confusing or unclear in the game?	Was there anything you felt was missing from the game?	Are you happy with the improvements and added features in the game?	What features would you add or change to improve the game?	Do you have any additional remarks or observations to share
2023/02/15 5:42:40 PM OEZ	2 full games	Fun	I like the new buttons and the option menu where you see the rules	a few (impeding your experience)	The point calculation does not work, so what is fair? :D	No its totally fine!	The points but I think thats because the calculation doesnt work	We need an leave the game button when you want to leave a round	Yes, so far I like the new things added!	Still the colours and the option to leave the rounds	
2023/02/15 6:19:35 PM OEZ	1,5 hours, and we played 2 full games	Still very fun, and i still want to play this during a dnd round	I like the "loosing on purpose" strategy that everyone adapts a lot,	a few (impeding your experience)	After playing it again, i think its fine with the many special cards. you really just need to estimate your cards well, which comes with time and hope that your opponents don't put down a card that doesn't benefit you.	during the round the only repetitive is when you only want to loose, have low colour cards and you don't have to start. and even then that is not really repetitive or long. The 10 rounds do get tiring after a while since it takes so long. 5 rounds would be enough too.	The new cheatsheet is great but the symbol is confusing, because it symbolizes settings, not hints. Please switch it out to something symbol like a question mark, explanation mark or light bulb icon.	An endscreen where you can see your points at least. also a way to quit the game to the menu	Yes, the info chart on the side is really good!	Please put a different indication to who's turn it is than just the "play card" button on the bottom. We had multiple instances where we were confused who's turn it is	My pc had a weird high noise, that only happens there. I don't think you could do something about it but i wanted to let you know.
2023/02/23 3:22:27 PM OEZ	15h; 2 rounds	very good	The Gameplay and the aesthetics	a few (impeding your experience)	It is well balanced and fair.	The game is ok as it is. The repeating parts allow for strategy	With the new diagram of beating order, everything is even more clear	A real menu: the new gear button suggests options but shows only instructions.	Yes, the new visuals are a huge improvement	Make the prediction buttons smaller, they only contain a number and can be perfectly round, not wide as they are now - the even obscure part of the screen now. I still think the "Play Card" buttons should be completely deactivated until it is one's turn. It's a much cleaner communication of turn order... until there is a "Your Turn!" message on the screen (which would be overkill and hand-holding. A good visual design [hiding buttons,	Can we get a name for the game? something catchy! :D
2023/02/23 3:57:18 PM OEZ	15, 2 rounds	Good	The Gameplay and aesthetics	a few (impeding your experience)	It is balanced and fair	It is ok as it is, it allows for strategy	no, as I know similar games. And also the new play order graph helps	Real options.	Yes I like them a lot	Now this is where I wanted to add something: Make a play tester "god mode" or something (I know, this far in development this is riddiculous), so players can "pick" the cards and rounds they want to play, as it allows for tracking down weird errors in various card combinations and play orders.	Sorry, now I made two feedbacks just to add this suggestion I forgot.
2023/02/28 11:28:53 PM OEZ	Playing for approximately an hour and a half, 3-4	Overall experience was good, with major improvements over the previous iteration. Still a few minor issues to be ironed out.	I think the ruleset is simple to follow, and makes for fast paced games.	a few (impeding your experience)	The game is well balanced, and the "cheat sheet" in game was helpful for assisting with remembering rules. The fey card offers an interesting dynamic with its interactions between with gates.	The random nature of the game keeps everything fresh, and it is well balanced for short games.	The point system was not functioning, which made it confusing to understand who was in the lead at any point.	A button/function to copy the session code when hosting the game. Would make it much easier to share the code with others to join. The ability to leave an ongoing session without completely exiting the game.	The cheat sheet helps a lot with the flow of the game, having easy access to the more abstract rules makes it much easier to play. The scroll function moving faster makes it much easier to see your whole hand of cards, as well as scroll through the rules.	Bring more of a visual flair to the cards, such as adding pieces of art to the fey, elemental, fiend, and celestial cards.	

Zeitstempel	How much time were you playing for? Alternatively, how many rounds have you played?	How was the overall player experience?	What did you like and dislike about the game?	Did you encounter any issues or bugs while playing?	Was the game well-balanced? Was it fair?	Did you feel the game was too easy or too difficult in certain areas? Was there any aspect of the game that felt repetitive or boring?	Was there anything that felt confusing or unclear in the game?	Was there anything you felt was missing from the game?	What features would you add or change to improve the game?	Do you have any additional remarks or observations to share
2023/02/15 5:44:30 PM OEZ	1.5 hours, 2 full games	it was very fun	i liked how intentionally loosing is a valid strategy	a lot (heavily impeding your experience)	hard to tell since the points weren't calculated correctly and because there was no final score/ranking	no not really	i sometimes felt confused which special card wins over what	no	final scores, scoreboard	
2023/02/15 5:46:33 PM OEZ	1 hour 30 minutes, Fun two full rounds		I liked that its a multiplayer game, the way it looks and the flow. I disliked that the rules are a bit hard to understand at first. I also disliked that you can't see your final results.	a few (impeding your experience)	The special cards appear fairly often, but outside of that it was fair.	No.	The rules. Also, at one point the buttons for the predictions one makes overlayed the number of points one had.	No	It would be nice to be able to leave an individual game/round without having to close the entire game. Also, the Play Card button should only appear when it actually is a players turn, otherwise it gets confusing.	
2023/02/22 8:21:05 PM OEZ	20 rounds with someone, 10 rounds on my own :(	(Played in 1v1) Not fun, didn't feel like i was in control of anything. Very much based on luck on what cards you get and on luck of predicting what the other player has. Game also left you very much on your own at what cards there are and how many of them there are, are there 2 elementals or just 1, are the duplicates of colored cards, whats the highest number there is?	You didn't feel in control at all, rules could have been explained more simply with pictures or drawings of the rule, game played like a try to guess what other people have in their hand without anything to go off.	a lot (heavily impeding your experience)	No, in the 1v1 situation we had it was more then likely in the first couple of rounds and later rounds that the person who goes first wins easily because of the color rules, since you don't have a lot of cards, the chances you have the same color and a high number at the same time where slim which makes it so that you basicky already lost the round supported by the facts that the person who wins the sub round also gets to go first another time, which means he	Not sure to be honest, so i didn't feel in control at all but my opponent also didn't it felt like we were just putting cards down whenever.	turn orders, not sure when someone goes first or not in rounds. The point system also felt really confusing at times where even my point counter was bugged sometimes.	Winner screen, if someone didn't look at their points at the end of the game who knows who would have won.	Winner screen, pictures to explain the rules more easily. Something to see what cards are in the game and how many there are of them.	Nope
2023/02/22 8:34:07 PM OEZ	20 rounds in a 1v1, 10 rounds alone for testing	As a player in a 1v1 setting, the most fun we had came from the conversation about how unpredictable it is and if there is even a strategy with the current implementation and rules. So the multiplayer aspect definitely was the fun source in a 1v1 environment. We also got annoyed that the first player to place a card has a substantial competitive advantage. Meanwhile, if they win, they go first again.	Like: The multiplayer aspect adds a lot to the game since you get social interaction, and there is a big gap in the market for fun co-op / social games. The game is simple to learn, and there are some strategies you can learn to win more often, but the winner's advantage needs to be fixed to make it actually fair. Dislike: That the winner of a sub-round goes first again. Rulebook can be much better	a few (impeding your experience)	It felt like it was too random, as you could achieve only a little bit using strategy. Since the first card advantage is too big. The first card advantage being too big is because of the color being too deciding, especially with the little number of special cards we usually got. Other card games like "bullying" fix this by allowing you to either match color or number. In this case color was too important for it to be fair.	The game quickly felt repetitive, as you do the same loop every sub-round. And since the actual impact the player can have is limited, at least in a 1v1 setting, it starts to feel too much of a game of chance. And yes, there are a few tactics, like if other players could not beat this card, they probably can't beat it again. But that is often more a trial and error.	Since the rulebook is so big, it requires a while to process. Aside from our few discussions about the gate system, it was clear. Although since it was only specified for the game, we were expecting a draw if we had the same card aside from gates.	The visual presence of other players. It felt like I was playing alone since we only see cards randomly appear, which could come from an AI as well without us knowing. I expect a turn order visual and indicator of waiting for other players, etc.. to be added.	I suggest adding more control to your ability to learn information. In real-life card games, this often comes from being able to see facial expressions, so games made from card game concepts often add features to make up for losing this ability.	The networking worked fine, and aside from the turn order, and a lack of player feedback / just A LOT of UI work still needs to be done for this to be considered a polished product. The gameplay is functional and straightforward; however, the fun factor would be missing for a large player base that can barely adjust the game's outcome.

2023/02/26 12:08:35 AM O EZ	2 hours alone against me and one game other people.	good but also confusing	i liked that you get more cards the longer the games goes on i think that the cards should be a little smaller because their are huge	a few (impeding your experience)	i mean it is a luck based game it kinda cant be fair cause of luck... or i am just stupid	the point system should get explained better	the point system.	A scoreboard, names for player (or just "Player 1" "Player 2"...), a "leave the game button", a text that tells you (while you are giving your prediction) when you start the trick,	everything that i said in "Was there anything you felt was missing from the game?" and extra designs for the cards (as an example: an elemental on the elemental card or the whole card has the color and not just a bit on the top)	No but i wish you great luck and fun with the rest your working time ^-^
--------------------------------	--	----------------------------	---	-------------------------------------	---	--	-------------------	--	---	---