BATTLE BEASTS Technical report

PSIT4 | IT18a_ZH

Aleksandra Timofeeva
Cyril Wanner
Erwin Tran
Lars Höhener
Ilbien Paul
Marc Berchtold
Marcel Brennwald
Marius Niklaus
Michael Schlaubitz

Version 1.2 11.05.2020

Table of versions

Version	Date	Changes
1.0	26.03.2020	Initial Version
1.1	27.04.2020	Outline of the technical report finished.
1.2	11.05.2020	Finalizing document.

Table of contents

1. Introduction 1
  1.1. Introduction 1
  1.2. Base line 1
  1.3. Goals and requirements 2
    1.3.1. Visual design 2
    1.3.2. Game modes 2
    1.3.3. Opponent AI 2
    1.3.4. Game content 2
    1.3.5. Mobile friendly 2
    1.3.6. Payment 2
2. Results 3
  2.1. AI opponent 3
    2.1.1. AIConnector 3
    2.1.2. AI Rules 3
    2.1.3. Random deck generation rule 3
    2.1.4. Card type deck generation rule 4
    2.1.5. Card value deck generation rule 4
    2.1.6. Random placement rule 4
    2.1.7. Card value placement rule 4
    2.1.8. Knapsack placement rule 4
    2.1.9. Random attack rule 5
    2.1.10 Weakest first attack rule 5
    2.1.11 Rated round attack rule 5
  2.2. Payment for cards and deck spaces 5
  2.3. Mobile friendly 7
    2.3.1. Cards 7
    2.3.2. Deck Editor 7
    2.3.3. Hamburger Menu 7
  2.4. Spell cards 8
3. Discussion and forecast 9
4. References 10
5. Table of figures 10

Abstract

Battle Beasts is a turn-based card game developed by a small team of software developers based in Zurich. It revolves around players creating decks of beast cards, which they then use to fight each other in one-on-one online matches. Battle Beasts' first iteration was very well received and thus the decision was made to continue working on Battle Beasts and make it even better.

The goal of the project of enhancing Battle Beasts was to add more features. Those features include an enemy with artificial intelligence, a payment system for buying new beast cards, deck spaces and a redesign for mobile phones.

The actual AI enemy is a computer opponent that lets the player play the game without needing another human. It has three different levels of difficulty. All difficulty levels consist of rules to generate a deck of cards, place cards and attack cards according to defined criteria. The rules' complexity ranges from simple to more elaborate but they all achieve a convincing result as if one was playing against another human player.

The payment system uses PayPal as payment service. New beast cards and deck spaces can be bought via a PayPal account to get better cards and more deck spaces for varying strategies.

For the mobile redesign new rules and designs were implemented, to adjust the components to smaller devices and improve the user experience.

1. Introduction

1.1. Introduction

Since the early days of computers people are playing solitary [1] on their PC. New card games were ported and new more complex ones were created. Well known for their on PC ported card game was Yu-Gi-Oh! [2] which then spread around the globe.

When online gaming was made available for every household online card games spread even more and with the possibility to be played on mobile devices, reached an all-time high.

Two of the most popular turn-based card games are Hearthstone [3] and Yu-Gi-Oh! [4]. Both let you fight with monster or magic cards against another player. Those two card games dominate the market and entertain a huge player base [5][6].

Using the popularity of online card games and with bringing in new ideas a first basic version of Battle Beasts was created. The main goal was to create a turn-based card game which is, in difference to all already existing card games, playable in the browser against other players on computers and mobile phones. It differentiates itself style wise from other games by being child friendly and even includes educational elements.

1.2. Base line

Prior to this project, a basic version of Battle Beasts was created. Battle Beasts is an online turn-based card game in which players fight against each other using beast cards, which are depictions of animals. The beast cards can be reinforced with spells and equipment. The players take turns and let their animals fight each other until one player has no more life points left.

The game came as far as to have a working environment, meaning it had an existing registration and login system, decks could be created with a variety of different cards and a matchmaking mechanism was implemented. Furthermore, it was already possible to play against real opponents and for beginners a tutorial was added.

Due to the great appearance, the good feedback of test players, and the booming mobile game market, it was decided to continue this project.

To get a better understanding of our project, it is hosted on https://battlebeasts.frostbolt.ch. Where you can test it, play rounds or read the rules.

1.3. Goals and requirements

The goal of this project is to continue the development of the prior developed card game Battle Beast by creating new features, implementing third party functions and optimizing already existing features. The goals were described and put into six different categories:

1.3.1. Visual design

The game will receive more visuals and animations. It will be possible to see the opponent's moves, attack animations for the cards will be implemented and attack indicators will be included which ensure that the player can directly recognize which cards are able to attack and which are not.

Furthermore, the deck creator will receive a button to directly buy new cards. A drag and drop function will be added for the deck creator. Deck names will receive a new design. Cards will receive different colors and pictures in the deck making list.

1.3.2. Game modes

It will be possible for the player to choose between game modes such as casual and ranked. Depending on the players win rate in ranked mode the player will get a player rank and will be shown on a leaderboard.

1.3.3. Opponent AI

The game will feature a self-developed AI to play against to train before playing against real opponents. The AI can create its own deck and will respond reasonably to the opponents moves. The player will be able to choose between different difficulties of the AI to optimize the training effect.

1.3.4. Game content

The game will be expanded with more animal cards, spell cards with different effects will be added and a new function to attack an opponent's card with multiple cards.

1.3.5. Mobile friendly

The game will receive a mobile friendly design, a new game board that is optimized for smaller screens and the cards will be redesigned and optimized for mobile screens.

1.3.6. Payment

The game will receive a third-party payment system which will give players the opportunity to buy cards and deck space.

2. Results

The results of Battle Beasts advancement include an AI opponent, the payment system and a redesign for the mobile phone.

2.1. AI opponent

The aim was to create an opponent that players could play against which is completely controlled by the server application. To achieve this without having to rewrite the pre-existing codebase, which was focused on providing a multiplayer experience, the adapter class AIConnector was created. Furthermore, the behaviour of the AI opponent is controlled by rule classes of which there is at least one for each game phase and which is defined by a corresponding interface. Depending on the selected difficulty more or less sophisticated rules are applied.

2.1.1. AIConnector

The AIConnector class imitates a real player by mocking a SocketIO connection object. This is being achieved by listening to emit commands to receive events and by creating events itself to execute actions as a game opponent. Thanks to this approach almost no backend source code had to be modified to support non-human players.

2.1.2. AI Rules

To control the AI, a rule class is applied for each of the major game phases. This rule class executes all the needed calculations and logic. It also instructs the AIConnector class to emit the events to execute the chosen actions. By combining the different rule classes, one for each of the game phases, the AI opponent difficulty can easily be adjusted. This approach also allows to easily implement new AI strategies.

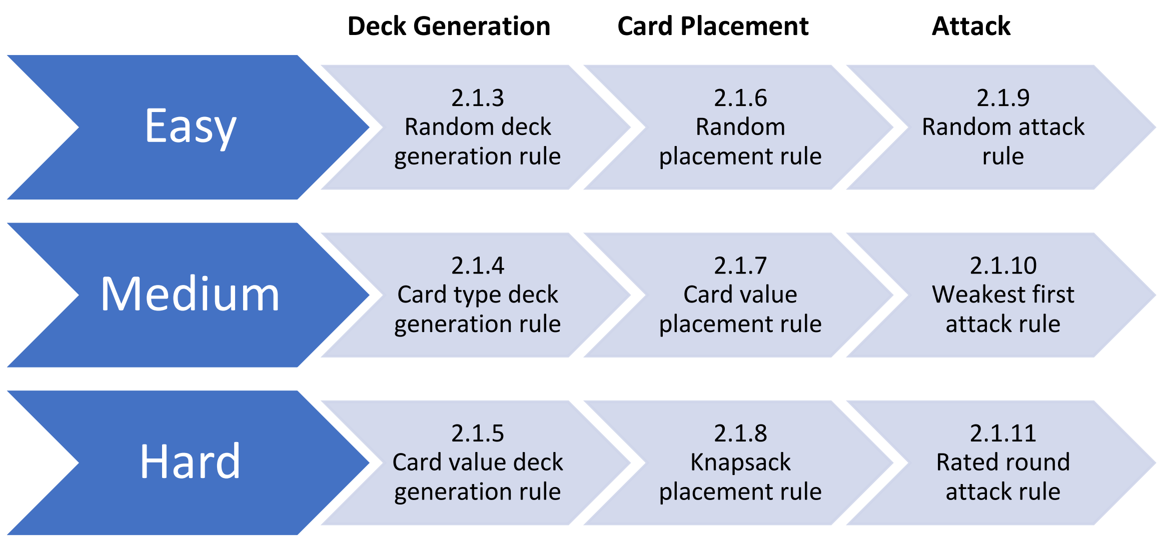
Deck Generation
Card Placement
Attack
Easy
2.1.3 Random deck generation rule
2.1.6 Random placement rule
2.1.9 Random attack rule
Medium
2.1.4 Card type deck generation rule
2.1.7 Card value placement rule
2.1.10 Weakest first attack rule
Hard
2.1.5 Card value deck generation rule
2.1.8 Knapsack placement rule
2.1.11 Rated round attack rule

Figure 1: AI Rules

2.1.3. Random deck generation rule

The simplest method to generate a deck for the AI is to just pick random cards until the deck is full. Which is exactly what this AI rule does.

2.1.4. Card type deck generation rule

To provide the AI with better cards than just randomly picked ones, this AI rule selects a specified amount of cards per type. There are four defined card categories: Attack cards (cards with higher attack points than defense points), defense cards (cards with higher defense points than attack points), equipment cards and spell cards.

2.1.5. Card value deck generation rule

This deck generation rule tries to generate a flexible deck that only consists of high value cards. The value of a card is defined by the ratio of its attack or defense points and its action points. Since there will be cards that are far more powerful than others, for their cost, the rule picks cards from three different action points categories: low, medium, and high cost. This prevents the rule from only choosing very high or very low cost cards, which would result in an easily beaten deck. From these three categories the highest value card is found and picked multiple times. Since equipment cards are only applicable to certain types of cards, they get chosen dynamically depending on the card type of the highest cost cards. The spell cards aren't chosen by a specific criterion as it is hard to assign them a value.

2.1.6. Random placement rule

As the name suggests, this placement rule places cards randomly from the hand onto the board. It does so until the AI has no more action points left. This behaviour can lead to very bad card placements as the rule does not consider whether playing a certain card is valuable or not. Since all animal cards have to be placed in attack or defense mode, the rule classifies the hand cards into attack and defense cards like the Card type deck generation rule and places them accordingly.

2.1.7. Card value placement rule

First, the rule calculates the card value for each card in the same way the card value deck generation rule already does. Spell cards, which are special and don't have those required attributes for calculating the card value, have a manually predefined value.

Once the values for all the cards on the AI's hand are calculated, it starts placing the cards until there are no action points left, starting from the highest value to the lowest value.

2.1.8. Knapsack placement rule

To choose cards to place for this rule, the knapsack problem gets solved by the AI in every round.

This means that the AI simulates all possible combinations of card placements and rates every placement as a whole. In the end, it executes the placement with the best rating and discards the rest.

To rate every placement, the card value (same calculation as in the card value placement rule) of each card in the placement is summed up. The difference to the card value placement rule is that two lower rated cards together could be stronger than one higher rated card which exactly gets taken into consideration by the knapsack placement rule as it tries to find the best possible combination for the available action points.

2.1.9. Random attack rule

The random attack rule is the simplest and weakest attack rule. Given a hand of cards, the AI randomly chooses one of its animal cards on the board to attack an enemy board card. This is only attempted for cards that are actually beatable by its own card. This can lead to very bad choices as a very strong card could be used to destroy a weak opponent card instead of a stronger one, wasting valuable attack points. If possible, the AI always prefers to attack board cards of the opponent before attacking the player directly.

2.1.10 Weakest first attack rule

This attack rule checks every attack card of the AI board from weakest to strongest in order of their attack points. So first it checks the weakest AI attack card and attacks the strongest player attack card it can beat. If there are no beatable attack cards or none at all then it attacks the strongest beatable defense card. If there are no defense cards left it attacks the player. If there are defense cards but they are not beatable the AI attack card is skipped and gets checked again in the second loop. After checking every AI attack card twice, it ends its turn.

2.1.11 Rated round attack rule

The aim of this attack rule is to find the mathematically best strategy by simulating all possible moves, evaluating and rating each, and finally choosing and executing the best one.

First for each of the AI board cards that are in attack mode all possible attack targets are gathered and added to a map variable which organises them based on the AI board card unique play id.

As a second step all possible attack strategies are generated by calculating all permutations from the possible targets map. Using this list of possible attack strategies each strategy is rated based on the amount of opponent cards that are destroyed (amount multiplied by 5 to weight them) plus the amount of damage done to the opponent itself. The best strategy is then executed.

2.2. Payment for cards and deck spaces

The aim of adding a payment system to the game of Battle Beasts is to profit off users who want to buy better and stronger cards and more deck spaces to be more flexible by creating a deck with different strengths. Therefore, a connection to the payment system of PayPal was created. The decision to use PayPal was made because PayPal has proper documentation and a sandbox system. A sandbox system means that as a member of PayPal, one can create sandbox accounts which have not real money as deposits. This is optimal for testing the connection because no real money has to be invested.

The process for both buying new beast cards and new deck spaces is the same. The process is as following:

1. User presses “Buy mystery card” or “Buy more deck space” on the My Decks site.
2. The server of battle beast initializes the payment via PayPal and redirects the user to the log in page of PayPal.
3. The user logs into his PayPal account and chooses the payment method (card or PayPal deposit). After that the user is returned to the website of Battle Beasts.
4. The user needs to confirm the payment. Otherwise, the user has the option to cancel the payment.
5. The server of battle beasts executes the payment via PayPal.

Below one can find the detailed process of Battle Beasts frontend and backend interacting with the PayPal service for the process of buying cards. The process consists mainly of three parts.

First, the payment is initialized which means that the PayPal service creates a payment which the specified amount of money. The payment receives a payment id from PayPal. After the initialization of the payment, the user is redirected to the PayPal website, as described above, logs into his account and chooses the payment method. Finally, the user can choose to confirm the payment or cancel the payment. By cancelling the payment no payment will go through and no money is transferred. By confirming the payment, PayPal executes the payment with the corresponding payment id and the money is transferred. The payment was thus successful. Note that the process of buying deck spaces is exactly the same one. Also note that this system sequence diagram only describes the best-case scenario and assumes that no errors happen. Possible errors that could happen are listed below

- After initializing the payment, the PayPal service could return an error, this would mean that the user would see an error message and can try to buy cards again.
- After executing the payment, the PayPal service could not approve the payment, which means that the whole process will be reset and the user also would see an error message that something went wrong with the payment.

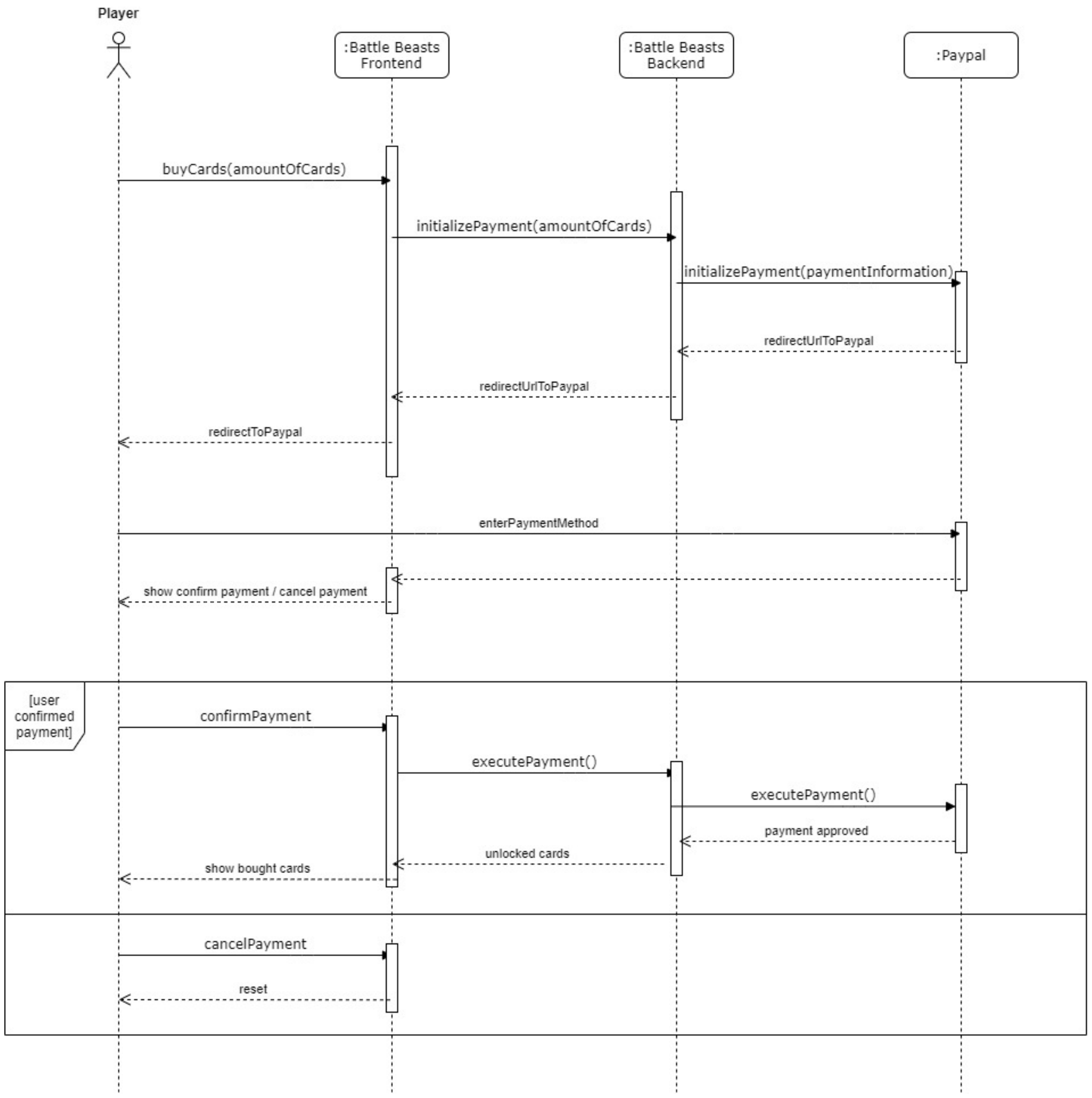

Figure 2: Payment sequence diagram

2.3. Mobile friendly

2.3.1. Cards

A major change which happened with the CardComponent, was that the team rewrote the cards from an SVG element to a HTML element. This decision has been made, because the team is more comfortable with HTML tags when it comes to responsiveness. Once that change happened, the mobile card size could be implemented, which only includes essential information such as the animal species, action points, attack- and defense values. Other information such as the description and image of the card, had to be removed or in the case of the image, put to the background, to keep the cards as small as possible. The main reason for resizing the card, was to retain a clear overview over the board on smaller device.

2.3.2. Deck Editor

For the mobile view, the editor displays the available cards and selected cards on top of each other, instead of both sides of the window in the desktop version, for a user friendlier experience. The team implemented an additional preview pop-up of the card when the screen size hits below 650px, which can be checked with: window.matchMedia('(max-width: 650px)'). This check was written in react not css, because the pop-up should only come up on smaller screens. Once the screen is small enough, hovering over the available cards no longer shows a preview and now clicking on these cards toggles the state "showModal", which if set to true, displays the ReactModal component. This component acts as the preview pop-up and in itself contains the same CardComponent with additionally the description and an add button.

2.3.3. Hamburger Menu

For mobile view the team has decided to implement a hamburger menu which toggles the navigation depending on a state called "isOpen". Using a similar check such as on the deck editor page, the hamburger menu shows up automatically if the user performs a resize of the window. The implementation had a bug on mobile devices. If a user tried to log in, the keyboard popped up but then the hamburger got closed. Debugging the code in the browser of the smartphone showed, that the handleWindowResize function got triggered which set the state of the hamburger "isOpen" to false. An extended check fixed that issue: If the window is smaller than 650px meaning the state "isMobile" is true, a check if the user is logged in is added. Furthermore, if the user is not logged in the hamburger gets closed otherwise the state is set to the previous state.

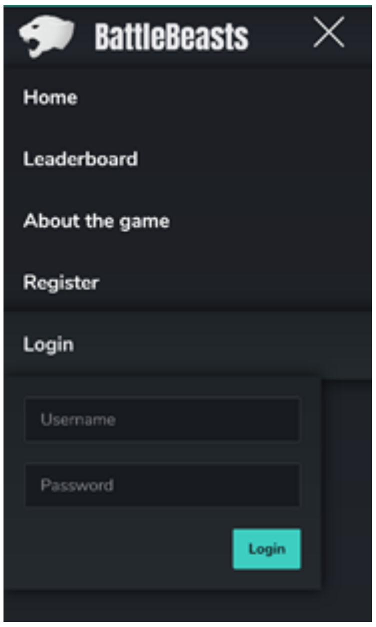

Figure 3: Navigation bar

2.4. Spell cards

In this project spell cards were added to extend the static equipment cards. Spell cards are similar to equipment cards and work in two different ways. The first way is a general effect on the gameboard or on a player, for example sending the opponent's cards back to his hand, healing yourself or damaging the opponent directly. The second way is a temporary effect on a chosen card that will vanish when the players next round starts, for example increasing the attack of a beast card.

Each spell has its own class that implements a spell interface which ensures that the 'activate spell' function is available. Each spell card has a specific name, with this name the fitting spell class can be activated in the spell card controller.

To activate a spell with a temporary effect the spell needs to be selected on the hand and a beast card on the gameboard. This is achieved by storing the spell card and its target card and sending them to the backend via a play controller. The frontend emits for clicks on the spell and target card and the backend listens on it.

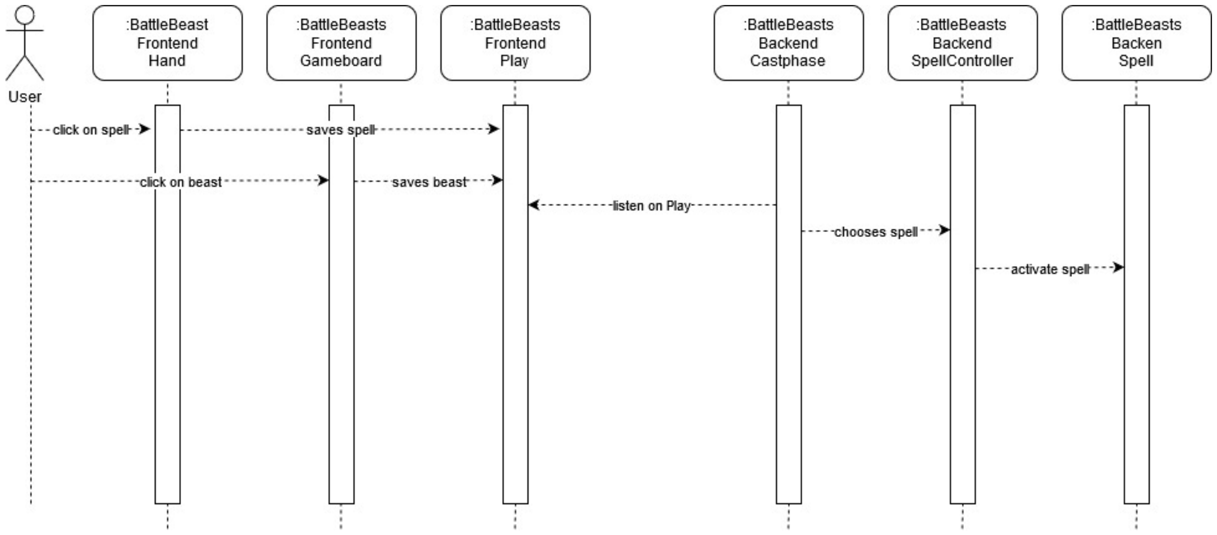
User
:BattleBeast Frontend Hand
:BattleBeasts Frontend Gameboard
:BattleBeasts Frontend Play
:BattleBeasts Backend Castphase
:BattleBeasts Backend SpellController
:BattleBeasts Backen Spell
click on spell
saves spell
click on beast
saves beast
listen on Play
chooses spell
activate spell

Figure 4: Spellcard sequence diagram

3. Discussion and forecast

The goal of this project, to expand the functionality of the prototype game application and improve the design, has been successfully implemented.

The results achieved during the project are impressive: thanks to acquired knowledge in the field of artificial intelligence the possibility of playing with a bot as well as playing ranked have been put into effect. Especially interesting are the newly developed rule strategies.

In addition, a mobile and tablet version of the game have been implemented and new design solutions were found. The project participants gained significant knowledge in one of the most popular front-end programming frameworks called React. This allows the application to be dynamic and removes the need for frequent reboots.

Furthermore, a game purchase procedure has been successfully introduced, which will facilitate the promotion of the game in the market in the future.

A further development of the prototype is planned, which will primarily include the refinement and expansion of functionality and testing. It would be advisable to conduct a survey with real users to collect better data. That means testing the game on real players.

4. References

[1] “Wikipedia,” [Online]. Available: https://en.wikipedia.org/wiki/Microsoft_Solitaire. [Accessed 05.11.2020].

[2] “YU-Gi-Oh! Card,” [Online]. Available: https://www.yugioh-card.com/en/products/games.html. [Accessed 05.11.2020].

[3] “Hearthstone,” [Online]. Available: https://playhearthstone.com/de-de/. [Accessed 05 .11.2020].

[4] “YU-GI-Oh! Duel Links,” [Online]. Available: https://www.yugiohcard.com/en/products/duel_links.html. [Accessed 11.05.2020].

[5] C. Gough, “Number of Hearthstone: Heroes of Warcraft players worldwide 2018 (in millions),” Statista, 25 03 2020. [Online]. Available: https://www.statista.com/statistics/323239/ number-gamers-hearthstone-heroes-warcraft-worldwide/. [Accessed 11.05.2020].

[6] STEAMCHARTS, «Yu-Gi-Oh! Duel Links Player Overview,» [Online]. Available: https://steamcharts.com/app/601510#All%20. [Accessed am 11.05.2020].

5. Table of figures

Figure 1: AI Rules. 3
Figure 2: Payment sequence diagram 6
Figure 3: Navigation bar 7
Figure 4: Spellcard sequence diagram 8