Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Algorithmically Balancing a Collectible Card Game

Bachelor Thesis

Jonas Krucher

jkrucher@ethz.ch

Distributed Computing Group
Computer Engineering and Networks Laboratory
ETH Zürich

Supervisors:
Prof. Dr. Roger Wattenhofer, Michael König

March 14, 2015

Acknowledgements

I thank Michael König who spent a lot of time giving me useful feedback and tips and helping me when I needed support. I also want to thank Professor Roger Wattenhofer for the opportunity to have an interesting insight into making a game and artificial intelligence. I appreciate the help of my family and friends who motivated me, were interested in seeing the final game and gave useful feedback after the playtesting session. Finally I thank Anna Balicka for reading through my unstructured thoughts and her support in stressful times during this project.

Abstract

Collectible card games are interesting to play because they usually consist of a huge amount cards allows many different strategies and variation. This successful concept is merged with that of a boardgame to a new computer game. For any card game to be interesting, the cards need to be well-balanced. In this project this is attempted to be done by an artificial intelligence. By playing many games against itself and constantly adapting the cards, the artificial intelligence modifies a randomly generated pool of cards.

Unfortunately the method did not lead to the desired balancing of the cards. Cards in the resulting set often lack a fair cost and can sometimes even be too bad to be played at all. The method is suspected to not be accurate enough and thus needs refinement. Another factor contributing to the failure could be a too small amount of played games on which the adaptations are based.

Altough the balancing did not lead to the requested results, the game concept was still well received by playtesters.

Contents

Chapter 1

Introduction

1.1 Motivation

Board games like chess are an all-time favorite for many people. They often include strategic elements and need thought and adaptation to the enemy's play. Collectible card games became more popular lately, especially games for newer devices like tablets (for example Hearthstone[1], Solforge[2] or Scrolls[3]) but also older cardboard based like Magic the Gathering[4] are still played regularly[5].

In this project parts of board games and of collectible card games were combined to form a new computer based game.

There exist other games which try to bring both collectible card games and board games together. Examples for this are Cards and Castles[6] which was released for mobile devices, or Faeria[7] which is still in beta stage.

1.2 Collectible Card Games

A collectible card game (CCG) is a type of game where the players build decks out of a typically big pool of cards. Usually there is a limited number of cards that can be put into a deck. The cards have a high variety of different effects, attributes and costs which allows many combinations and strategies. In most CCGs, the cards are drawn from a shuffled deck to be played against another deck[8].

Chapter 2

Game Design

2.1 Basic Mechanics

In this game, as it is true for the most card games, players have cards in their hands which they draw from their decks. Each deck consists of 40 cards. Players are taking turns consecutively. The game takes place on a board, which is a finite hexagonal grid. There are figures on the board which can be interacted with. Players can play the cards, which have abilities to affect the board and the figures on it. Figure 2.1 shows a screenshot of a typical scene during a game.

Figure 2.1: A screenshot showing a typical scene during a game.

2.2 Heroes, Minions and Characters

At the beginning of the game, both players each control one figure on the board which are called the *heroes* of the players. The goal of the game is to kill the enemy's hero. As the game progresses, players can place other figures on the field, so called *minions*. Minions and heroes are further referred to as *characters* and have the following basic attributes:

- Attack
- Health
- Movement
- Range, consisting of minimum and maximum range

Characters can take damage, which is indicated by the current health of a character (maximum health - damage). A character dies if it's current health is zero or below.

A character can attack other characters within their range. This means that it deals it's amount of attack as damage to the defending character. The defending character does the same thing to the attacker, if the attacker is in the range of the defending character. Characters can move on the board according to their *movement*.

A character's attack is at least zero, whereas range and movement are always numbers between one and five.

At the start of a players turn all their characters regain the ability to move and attack once this turn. When a minion enters the board, it cannot attack or move in the same turn.

Characters can be *poisoned, paralyzed, armored* and *immune*. When a character is poisoned, it takes n damage at the beginning of the controlling player's turn, where n indicates how heavily the character is poisoned. The number n then gets decreased by one (until it hits zero and therefore vanishes as it has no effect anymore). The state poisoned always comes with such a number n. When a character is paralyzed, it cannot attack, move or defend an attack. Paralyzation also has a number n attached, which indicates how long it lasts and is also decreased by one at the start of the controlling players turn. Similar to the poison, paralyzation vanishes when it reaches zero. Poison and paralyzation stack up, meaning that for example when a character is poisoned 1 and gets poisoned 1 another time, it is poisoned 2. Armor means, that whenever the character takes damage, one of this damage is prevented. When a character is immune, it cannot be poisoned or paralyzed. However, if a character was poisoned or paralyzed before becoming immune, poison and paralyzation are treated as before. In that case it just cannot get paralyzed or poisoned more than it already is.

Minions can also have effects, which are triggered at certain moments in the game. The possible triggers are:

- When a minion enters the board
- Whenever it fights with another character
- Whenever it kills another character in combat
- At the beginning of the controlling players turn
- At the end of the controlling players turn

2.3 Land

Each field on the board has a land type. A core mechanic of the game is the possibility to change the land on the board to your needs. There are five land types: fire, water, nature, earth and wasteland. Players can change a land's type in their heroes range once per turn, which is indicated by changing the texture of the desired field of the board. However they can only change the type to fire, water, nature or earth. Changing the land is crucial as players can only play cards, if the pattern indicating their cost is fulfilled by the board. There can also be cards which can change land types with their effects or turn them to wasteland, which means that the previous land type is removed and they become wasteland again. When a game is started, all the land on the board is wasteland.

2.4 Deck, Hand and Graveyard

At start of a game, the players each draw three cards from their decks. Players can have at most three copies of one card in their decks. This does not apply for randomly generated decks which can have arbitrary many copies of one card inside.

At the beginning of a player's turn, that player draws a card from their deck. If there are no more cards left in the player's deck, simply no card is drawn. There is no additional penalty, as for example in Hearthstone, where a player loses life points whenever they have to draw a card when the deck is empty[1]. Magic: the Gathering rules are even stricter: players lose the game, if they have to draw a card from their deck but can't[9]. In this game the state of no card being drawn is considered as a penalty severe enough, as the player cannot strengthen their position anymore. This will eventually also lead to an end of the game, as nothing can be played anymore at this point to turn the tide again.

At the end of a player's turn, if that player has more than eight cards in their hand, random cards are discarded until the player has eight cards in their hand.

Cards which were already played are put in the so called *graveyard* of the player who owns that card.

Decks can have cards which need any land type(s) in them, there is no restriction in terms of allowed number of land types. However it is recommended to only play with two or at most three land types. It gets harder to play cards from your hand when playing with all four different land types, because more land types have to be changed in order to be able to play the next card. However it could still be interesting to play with all the four land types, as more combinations of the cards and their effects are possible. In theory it is also possible to play cards of one land type only, but it may be hard to assemble a good deck as there are most likely only few cards with one color only due to the way the cards are generated.

2.5 Cards and Effects

There are two types of cards: namely minion cards and spell cards. Every card has at least one effect. Minion cards only have one effect, they spawn a specific minion on the board. They always have the condition that the minion has to be spawned inside the range of the player's hero.

Spell cards can have multiple effects, for example healing or dealing damage. For the cost of minion cards, one has to be look relative to where one wants to place the minion, for spell cards the pattern has to match relative to the position of the player's hero.

Chapter 3

Balancing

3.1 The Artificial intelligence

The purpose of the artificial intelligence (AI) is to play many games against another instance of itself in order to balance the cards. The cards will then get rated and modified according to how well they did during these games, meaning that they get a good rating if the player who played the card won the game and a bad rating if the player loses.

One main criterion for the AI was to play the turns fast, as the rating of the cards is highly based on the number of games played with them. The simulation is limited to one second per turn or 5000 action sequences, which are explained later in this section. The AI also stops when it explored all the possible action sequences. This can especially happen in the beginning, as there are for example no minions or only few on the board which could lead to more actions. A larger number of played games is desirable. It is not desired to do adaptations based on only a small number of ratings, as the rating of a card could also be based on its interaction with another card, or it could not have played a strong role in the outcome of a game.

The AI tries to simulate actions and then rates the outcome when these are applied to the initial game state the AI had at beginning of the turn. Possible actions are:

- Character move
- Character attack
- Land placement
- End turn
- Play card

Each of these actions can have several places/targets to be applied to, for example minions and heroes can move to the left or to the right and it is possible

to place a land of the type fire or one of the type water. The number of allowed turn ending simulations was limited to two, as otherwise there would be huge subtrees possible out of this action. Still next turns should be considered in order to be able to plan a bit further. If the AI can see that there is a very good path in the tree in the nearer future, it can already do the necessary things in this turn to be able to go in this direction. An example could be the possibility to play a stronger minion next turn, if the AI does not place a land for a weak minion to be summoned this turn. However the end turn actions are only based on what is already there, the AI plans with a very optimistic scenario of the enemy player doing nothing in their turn (besides drawing a card). But the effects of the minions which get triggered at the beginning or the end of the turns are considered.

3.2 Rating the Actions

Whenever the AI simulates an action, the resulting game state is rated. When the AI finished the simulation, due to the time limit or one of the other reasons mentioned above, it executes the action sequence with the best rating.

The rating considers all the attributes of the minions on the board with a weighting for each of them, where the attributes of the minions which are controlled by the AI are rated positively and the attributes of enemy minions are rated negatively.

$$Rating = Board_{own} - Board_{enemy} + Hand_{own} - Hand_{enemy} + Additional \quad (3.1)$$

The board ratings $Board_{own}$ and $Board_{enemy}$ are computed as in the following equation:

$$Board_{player} = M_{player} + H_{player} + L_{player} \quad (3.2)$$

M_{player} is the sum of all attributes of the minions of a player, each multiplied with a weight. A minion also gets an additional positive rating, if the enemy's hero is in it's range, because this means the minion can attack the hero and thus contribute to the final goal of defeating the enemy's hero. H_{player} is the same for the hero of the player but without the additional rating of having the enemy's hero in range. At last the number of non-wasteland land around the hero of the player is considered in L_{player}. This is rated positively, it is considered as an advantage for a player because it is more likely to be able to play cards.

It is be desirable to rate the attributes of heroes differently than the ones of minions. For example it could be that a higher range of a player's hero is much better than a higher range of one of the minions of the player. In this example the

land could then be changed from further away or minions be spawned without having to wait for the own hero to move to a field near to where the minion can be placed, which can be very powerful.

The hand ratings $Hand_{player}$ are very simple. They just multiply the number of cards in the hand of a player with a weight. A game state is better the more cards there are in the AI's hand and worse the more cards there are in the enemy's hand. With more cards in a player's hand it is more likely to be able to play something which can strengthen the own position.

Finally the *Additional* part consists of a very positive rating if the enemy hero is dead and of a very negative rating if the own hero is dead. Since the goal of the game is that the own hero survives and the enemy's hero dies, we want to take this into account very heavily. This is done by subtracting a ∞ rating if the own hero dies and adding an ∞ rating when the enemy's hero dies. As a rating of ∞ is not possible in a program, these numbers are just selected to be very big (+1000000 and -1000000). When there is a chance to kill the enemy's hero this should always be done and also if the own hero would die, we absolutely never want to consider this action sequence as an option. In the *Additional* part it is also taken into account with a positive rating if the player changed the type of a land this turn. We want to change the type of a land in every turn, as it can also contribute to cards which are drawn in the future. If the land around the hero is already suited for the hand, the AI can still just change a land to the type which it already has, which is never bad.

The exact weights and ratings used in this project can be seen in Table 3.1.

Tuning the weights is difficult, in particular because the game has not been tested for a longer period of time as for example Magic the Gathering, which was adapted since it came out in 1993[10]. The weights could be optimized by playing a lot of games with different weights and taking the best of them, but this is out of the scope for this project.

There are also a few restrictions to bound the possible actions which can be taken, because a goal is to be fast in order to simulate many games in a relatively short amount of time. Action sequences with a lower rating than the game state, which the AI had at start of its turn, minus a threshold (50) are not considered any further. This is because it is assumed that from a state this bad it is not possible to recover. However we do not want to throw away all the action sequences which are only a bit worse than the initial game state, as there can be cases which get very good later. For example a player may want to sacrifice two weaker minions in order to kill a bigger minion of the enemy player. This sequence would be worse after the attack of the first minion, as the player loses the minion and the enemy only lost a few health points of their minion. But after the second action, the player will be much better off, as they killed a bigger minion of the enemy while only losing two smaller ones, which will result in a possibly better rating than all the other sequences of actions could lead to.

Table 3.1: Weights and ratings used for rating a game state.

Minion attack weight	4
Minion health weight	4
Minion range weight	4
Minion movement weight	4
Minion immune weight	4
Minion armor weight	4
Hero attack weight	5
Hero health weight	5
Hero range weight	5
Hero movement weight	5
Hero immune weight	5
Hero armor weight	5
Minion moved rating	5
Minion attacked rating	5
Minion distance to enemy hero rating per tile	1
Own minion has enemy hero in range rating	100
Hand card weight per hand card	4
Hero is dead rating	1000000

So there is always a trade-off in deciding which sequences we want to keep due to the possibility of them leading to the best sequence and which we want to throw away due to efficiency reasons and the risk that the best sequence was not found, because there was too much time spent on parts which would not lead to anything good.

3.3 Rating Cards

At the end of a game all cards which appeared in the game are rated. Table 3.2 shows the used ratings. Cards are rated based on if they were in the hand of a player, on the board (minions only) or in the graveyard and thus already played, together with the information if the player won or lost the game or if it was a draw. These ratings are assumptions of what could make sense and lead to desired results, but could also be tested and adapted further to contribute to a more accurate adaptation of the cards.

The ratings for the minions on the board are straightforward, if the player won the game, they are rated positively and if the player lost the game the ratings are negative. Minions on the board get a rating of zero when the game ended in a draw. It might be a bit surprising at first, that when a game was won, hand cards are rated with zero. This is because it cannot be said that these cards contributed to the result of the game. In fact it cannot be evaluated if this

Table 3.2: Ratings used for cards at the end of a game.

Minion on board rating when a game was won	2
Minion on board rating when a game was lost	-2
Minion on board rating when a game ended in a draw	0
Handcard rating when a game was won	0
Handcard rating when a game was lost	-2
Handcard rating when a game ended in a draw	0
Card in graveyard when a game was won	2
Card in graveyard when a game was lost	-2
Card in graveyard when a game ended in a draw	0

card was good or bad in this game, as it was never played. This does not hold for the cards in the hand of the player which lost the game. These cards are suspected to be either not helpful or not playable and this might be a reason why the player lost the game, thus these cards are rated negatively. The cards in the graveyards (the spells which were played and the minions which died on the board) are assumed to have contributed to the final state of the game and thus are rated with a positive number when the player won and negative when the player lost.

3.4 Adapting Cards

When the AI played a certain number of games, the cards are adapted based on their ratings. In this project this was done every 100 games in order to get a meaningful amount of ratings, but still not have to run too many game for one adaption. Then the average and the standard deviation of all the cards' ratings are computed. The cards we want to adapt are the outliers in terms of rating. We consider a rating as an outlier if it has a deviation of at least 1.5 to 2.5 (depending on the case) times the standard deviation from the average rating. Additionally, the cards' win/loss ratio is taken into account and adapts the threshold from which on we speak of a rating as an outlier. Namely when a card has a negative overall rating the result of equation 3.3 is chosen as the threshold and for cards with a positive overall rating the result of equation 3.4 is chosen.

$$(wonGames/totalGames) * 2.5 + (lostGames/totalGames) * 1.5 \tag{3.3}$$

$$(lostGames/totalGames) * 2.5 + (wonGames/totalGames) * 1.5 \tag{3.4}$$

The reason for this adapting threshold is for example when a card has four bad ratings, we want it to be considered as a worse card than a card which lost six times, but also won two times. If we only took their overall rating into account, they would have been both considered to have the same card value.

We adapt cards by adapting their cost. When a card's rating is bad enough to adapt it, we remove one land cost at a random position up to a minimum of one land. When a card's rating is too good and we want to adapt it, we add another land cost of a random type which is already in the cost at a random position up to a maximum of 35.

3.5 Unplayable Cards

Due to the almost random generation of cards, there may be cards which are too bad to be played, even with the lowest cost possible (which is one single land). For example there may be a spell, which does two damage to your own hero. There is no way this card can be good (at least with the current rules, triggers and effects), so we want cards like this to be treated specially. One could argue that it should be prevented to generate such effects in the first place, but such negative effects can also be interesting to play and be put into strategies. For example there could be a spell card, which adds four attack points but also deals two damage to the same minion. This card could either be played on a weaker enemy minion to kill it, or it could be played on a stronger friendly minion to boost it. This card would probably have a lower cost than a spell which simply adds attack or deals damage, as it contains both positive and negative effects. This can make such a card very attractive to play. Other examples would be a spell which lets you draw three cards but also deals five damage to your hero or a minion which is strong and cheap, but also has a negative effect such as letting the owner discard two cards when it is played. Other card games, such as Hearthstone also have cards like this. For example a lot of cards of the Warlock class in Hearthstone have the negative effect that the player has to discard cards when he plays something, but the cards are stronger than other cards with the same cost in return[11].

We now have arguments which justify the existence of negative effects. Nevertheless there can be bad cards which just damages your own hero. We want to find such cards and replace them with new randomly generated cards. The way it is done in the game is that cards which have a rating which is bad enough to be adapted but already have cost one (the minimum cost) are marked. If a card has three marks, it is deleted and a new card is generate in its place.

Chapter 4

Implementation

4.1 Game Engine

4.1.1 Framework

The following C++ libraries were used in this project:

- SFML[12] was used for threading, networking, windowing and 2D graphics. It was also chosen because it is available for all major platforms.
- TGUI[13] provided GUI elements, such as buttons or list boxes.

Furthermore these programs were used:

- GIMP[14] was used for drawing the graphics.
- MATLAB[15] was used for plotting graphs.

4.1.2 Networking Architecture

The networking part consists of a central server, to which two clients can connect at a time. When a user hosts a game, a server and also a client are created. This client then directly connects to the server. The clients send all the actions which they want to make to the server, where these actions get checked if they can be applied and all the conditions and costs are fulfilled. If everything is correct, the server forwards the applied changes to the clients, probably splitted up in smaller parts of simpler logic. For example if a player plays a spell which deals one damage and adds two attack, the server just sends that the spell was played, one damage is dealt and two attack were added and the client can just apply that without having to check for any conditions to be true, as this was already done by the server. The architecture fits the fact, that only the server should be trusted. A client can be modified in a way that he can send anything at any

point in time to the server, so it must be possible to check if the sent actions are legal. For the future it would be desirable to have one official trustful server which manages all the games.

4.2 User Interface

4.2.1 Lobby

When a user launches the game, the first screen to be seen is the lobby. Here a user can enter a name, join a game, host a game or quit the program (Figure 4.1).

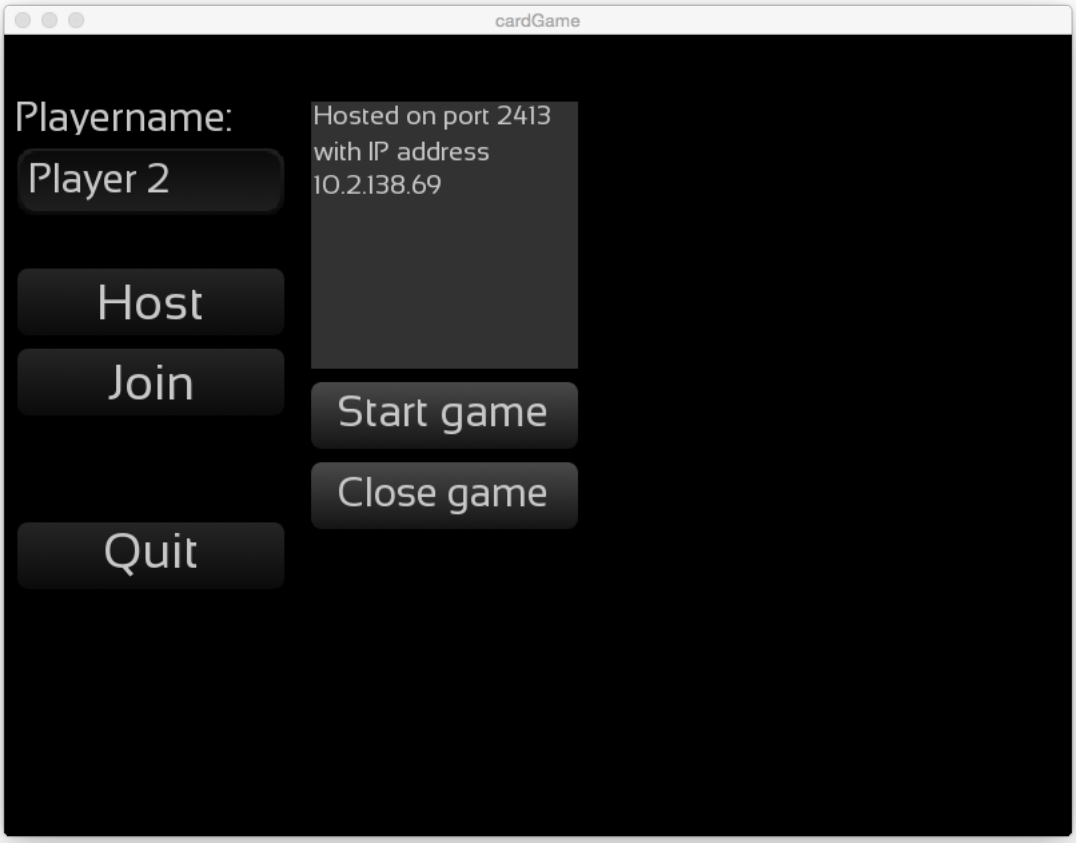

Figure 4.1: Hosting a game in the lobby.

4.2.2 Gameplay

At the start of a game the user can see the board in the middle of the screen, the cards they have in their hand at the bottom of the screen and an information box in the top left, as it can be seen on Figure 4.2.

Everything in the game can be controlled with the mouse. To select something like a card or a minion on the board, a player left clicks it. In order to perform actions, like playing a selected card, a player right clicks on the target. If something needs more than one target, they have to be right clicked in the correct order. For example if a spell deals one damage to a minion and provides another minion plus one attack, a player first right clicks on the target which should be dealt one damage and then on the target which should get plus one attack in order to play the spell. For spells which do not have a target to be specified, a player can just left click somewhere on the board to play them.

By clicking on a card in the hand, the properties of the card are displayed on the right side (Figure 4.2). In the top, it is possible to see the cost needed to be able to play the card. Right below, the name and the image of the card is shown. Minion cards have a box with their attributes and another box with their effects, if they have any. Spell cards only have the box with the effects, as they do not have attributes.

When a character on the board is selected, details are shown on the right side, similar to what is shown when a card is selected. Additional information about where the character can move and which other characters it can attack are shown with an orange / red indication on the board (Figure 4.3)

There are small circles in the colors red and blue shown next to characters (Figure 4.4). These indicate if the character can attack and move this turn, where the red circle shows that the character can attack another time this turn and the blue circle indicates that the character can move another time this turn.

Figure 4.2: Screenshot of a typical scene in the game. Detailed information about the selected card is shown on the right side.

Figure 4.3: Screenshot of the game when a minion is selected.

Figure 4.4: Closeup of a character where indicators show if it is still possible to attack and move with it.

4.2.3 Game End

When a game ended, the player can see a screen, where it is displayed that the game ended and if the player has won or lost the game. The user gets back to the lobby with the escape key.

4.3 Effect Containers, Effects and Conditions

In order to provide freedom in terms of being able to use different effects and combinations of them for randomly generated cards, the previously used term *effect* was defined to consist of the three parts effect containers, effects and conditions.

Effect containers consist of a target type, effects and optionally conditions. There are around 30 different target types, for example *minion*, *random character* or *enemy hero*. Some of them allow the player to choose a target as for example *minion*, whereas others chose a target randomly or the target is already clear. Examples for these are *random character* or *all friendly minions*.

When a condition of an effect container is not fulfilled, all effects in the container will not activate. Around 20 different conditions are included in this project, for example *the target needs to be in the own hero's range* or *the own number of cards in the hand is equal some number*.

Effects themselves can also have conditions. If these conditions are not fulfilled, only that effect will not activate, but the effect container as a whole will not be affected in case it has more than one effect. Effects also have a type and can have an amount, when it is necessary for the effect type. The effect type defines what it will do, for example deal damage, whereas the amount would then indicate how much damage is dealt. There are more than 30 different effect types included in the game. Examples are *deal damage* or *draw card*.

This architecture allows to have a certain effect applied on multiple targets or several effects on one target and any combination of these. More freedom and the possibility of many combinations are provided, which is in particular interesting for randomly generated cards.

4.4 Random Generation

4.4.1 Motivation

One part of this project is balancing the cards. Balancing can especially be interesting when the cards are generated randomly, as it is not known beforehand which combinations are exactly generated in the end. It is an interesting

challenge to come up with strategies as no one has thought over which card could match together to form a strong deck. For cards in this game names, images, attributes and effects are needed. Besides the images, these are generated randomly.

4.4.2 Random Name Generation

Names give a strong indication of what something is, can help the imagination and underline an image. They also illustrate parts of the world in which the game takes place in. Cards in collectible card games have specific names, which often consist of different words combined. For example in Magic: the Gathering there is a card called *Fanatic of Xenagos* which corresponds to a minion in this project. Another card is *Warstorm Surge*, which would be a sort of spell. The two cards can be seen in Figure 4.5. The name *Fanatic of Xenagos*, consists of *Fanatic* which exists in the English language, the binding word *of* and a fictional place in the Magic: the Gathering universe *Xenagos*. In *Warstorm Surge* we can see that two nouns were coupled together: *War* and *Storm*, followed by the other noun *Surge*. In this project, random names are generated according to these models. A name can consist of randomly generated fictional words, the binding word *of*, a few predefined existing nouns and can have an adjective in front, which is also chosen from a predefined pool. The fictional words are generated with a simple algorithm: they consist of four to eight letters and follow the rule to always consist of a vowel followed by one or two consonants and then again a vowel and so forth. When a fictional word is in the name, one can also think of it as a imaginary creature, place or something else which does not exist in the real world.

Figure 4.5: The Magic: the Gathering cards *Fanatic of Xenagos*[16] and *Warstorm Surge*[17]

4.4.3 Random Card Generation

Cards always have a random name, as described above and also get a random cost, which consist of one to three types of land types. The cost is more likely to consist of two different land types, with a chance of 60%, rather than having a single land type, where the chance is 30%. Three different land types are the least likely cost to occur with a chance of 10%. The number of lands in the cost is randomly chosen between one and seven when the card is generated. There is a distinction between minion and spell cards. Minion cards have randomly generated attributes, with a few restrictions. The health is a random number between one and nine. There is a big chance to have more or less the same value as the health for the attack, namely a 25% chance of having the same value as the health and a 55% chance to have an attack which differs from the health by a uniformly chosen random number between zero and two. The chance of choosing the attack to be a new uniformly chosen random number between zero and nine is 20 %. The range for the minion is a number between one and three, where the smaller values are more likely to appear, as a bigger range is considered to be exceptionally strong. The chance to have a range of one is 55%, the one to have a range of two is 30% and the chance to have a range of three is 15%. The minimum and maximum range are with a chance of 75% most likely the same, but it can also be the case that the maximum range is the minimum range plus one with a chance of 25%. A minion's movement is in 65% of the cases randomly chosen between one and two, but can also be randomly chosen between zero and four with a chance of 35%.

These numbers and chances are assumptions based on playing different collectible card games. With these restrictions it is much simpler for humans to calculate minions into strategies. This is especially the case for higher ranges and higher movement, as it is at some point very hard to tell what can happen next, if a minion is moved for example. It becomes hard to preserve a general view of what is happening on the board. This is not fun to play, but cumbersome to deal with. A similar reason holds for big differences between attack and health. These are interesting if they are not the rule. Minions with almost equal attributes in terms of attack and health are easier to handle and embed into strategies. For example it is not desirable to have many minions which have big attack but all low health, as they are an easy target for weak minions of the enemy or spell cards which deal only a small amount of damage. A deck is better off if it has only few such minions but more evened ones which can be used in various strategies and circumstances.

Each of the five possible triggers (see Chapter 2.2) has a chance of 25% to have one to two effects in one effect container. Like this, it is possible to have a few heavier minions with many effects, but also some which have no effect at all and thus are simpler to take into account for any strategy. Spell cards only need effects, which are generated in the same fashion, with the difference that

they can have one or two effect containers (equally likely to appear) with one or two effects each. We want at least one effect in each spell, as they are useless otherwise.

Chapter 5

Evaluation

5.1 Card Balancing

In this project, a set of 225 different cards was generated, consisting of 150 minion cards and 75 spell cards. A total of 7500 games were simulated by the AI. A game was aborted and declared as a draw game, if the sum of turns of both players exceeded 80. This maximum turn number was chosen, because it is certain that after 80 turns at the latest both player's decks would be empty and thus the game would not make much progress. After every 100 simulated games, the cards were adapted according to the ratings. Thus, the set of cards was modified 75 times.

The average number of rated cards per modification was 198.01 from the total of 225 cards. Each card was rated 5.33 times on average (including the cards which were rated zero times).

On average, the AI considered 3690.87 different action sequences per turn from the allowed maximum of 5000. The average length of the longest of these sequences in each turn was 28.64 actions.

In Figure 5.1 it can be seen that it does not matter for the result of a game which player started. Many games were aborted because they reached the maximum number of turns allowed. Also there were almost no draw games other than due to reaching the maximum number of turns allowed.

The average number of turns taken over all games is 40.39 without the games which reached the maximum number of turns and 54.38 with these games. This distribution can be seen in Figure 5.2. An interesting feature is the large amount of games which only took five to ten turns. This could be explained by certain cards, which make a game much shorter and can turn out good or bad. An example of such a card is a cheap minion which reduces the health of both heroes to five when it is played. The high number of aborted games could be interpreted as stalemates due to even games.

Figure 5.3 shows the average, minimum and maximum ratings of the cards when the modifications happened. The average rating is around zero and does

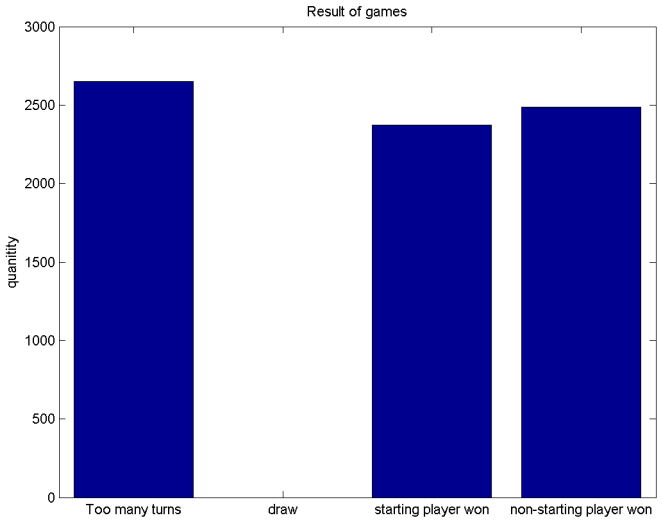

Figure 5.1: Result of the games.

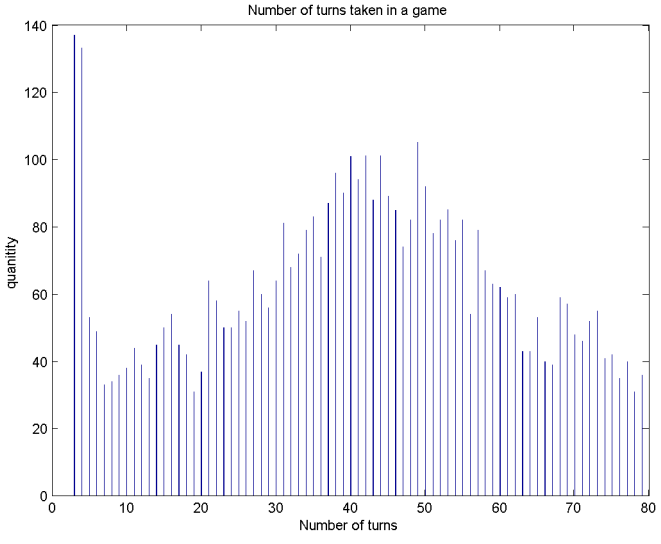

Figure 5.2: The number of turns taken, measured over all the games without the games which were aborted due to having reached the limit of 80 turns.

not change much, which is what could be expected as this can be seen as the final rating for all balanced cards. The most negative ratings are further away from the average than the most positive ones. In exchange, there are more cards with a positive rating (not necessarily outliers), as the average would not be around zero otherwise. One reason for this could be the random generation of the cards, as there may be a lot of average cards (which can also win a game) and only few bad cards with which a player loses often.

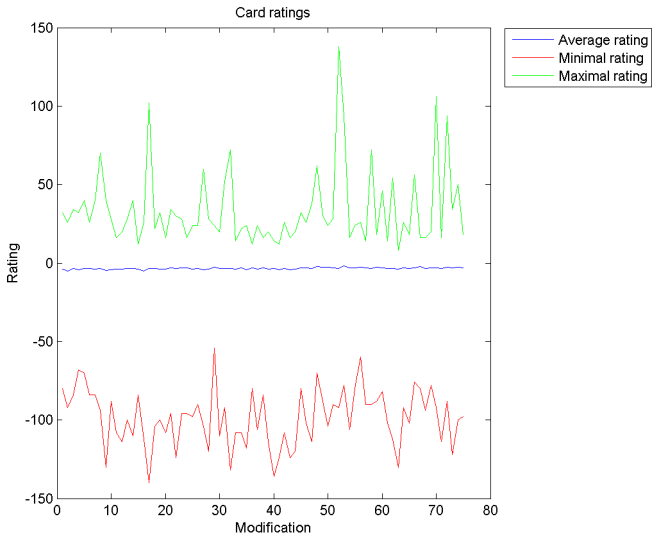

Figure 5.3: The average, minimum and maximum ratings of the cards at the time of the modifications

513 cards were adjusted in the 75 rounds of modification. 240 were modified to be cheaper and 273 to be more expensive. There were no unplayable cards detected in the randomly generated set, so no card had to be generated anew.

The number of modified cards per modification oscillates relatively severely, but a slight trend towards zero can be seen in Figure 5.4, a few adaptions less were made when comparing the last modifications to the first ones. However this trend is not very strong. This could be mainly due to the selection of which outliers are modified. They are selected based only on the ratings of the other cards and thus there can possibly never be a state where the ratings would be so close to each other such that there were no outliers anymore. This problem could for example be addressed by a (small) fixed minimum rating difference from the average in whose bounds the cards would not get modified. This could be desirable as the modifications could then be run until no further modifications happen for a certain amount of rounds. Then the cards could be considered as balanced.

The standard deviations in Figure 5.5 illustrate a similar picture. The trend

goes down, but slowly and with many oscillations.

Another issue could be the way the modifications were made. It could be that an additional cost which is more distant from the position where a card can be played makes a card harder to play than an additional cost which is nearer. This could be included by having several types of outliers. Cards with ratings which are very far away from the average could be adapted with an additional cost which is further away from the position where the card can be played, whereas the outliers which are nearer to the average could be adapted with an additional cost close to the center. The cost reduction could be done in a similar way: the cost which is removed can be near or further away from the place where the card can be played.

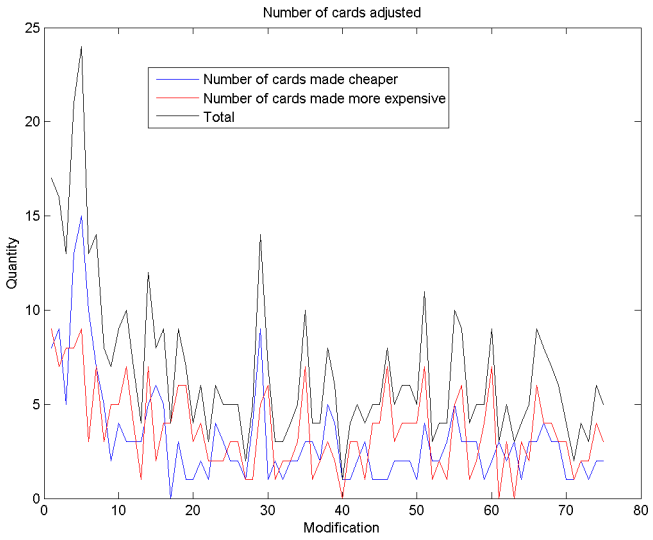

Figure 5.4: The number of modified cards for each modification.

In Figure 5.6 it can be seen that the number of cards rated, as well as the average number of ratings, decreases with the number of games played. The reason for this effect is that the number of games which were declared as a draw due to reaching the maximum number of allowed turns increased for the later modifications. This suggests that the game became more even and therefore often ended in a draw. The reason for this could be that the cards were more balanced and the decks equally strong in more games, which is exactly what the goal of the modifications was.

Overall it can be said that the modifications were a partial success, at least from the AI's point of view. The games tended to be more even and less cards were modified the more games the AI simulated. However there was no point in these 75 modifications where it could be said that there were no more mod-

ifications to be made (see Figure 5.4 and 5.5). Improvements in the balancing could probably be made by adapting the term of outliers when looking at the ratings. It also has to be said at this point that the AI was kept simple and the results could differ when improving the AI or granting it more time per turn for calculation.

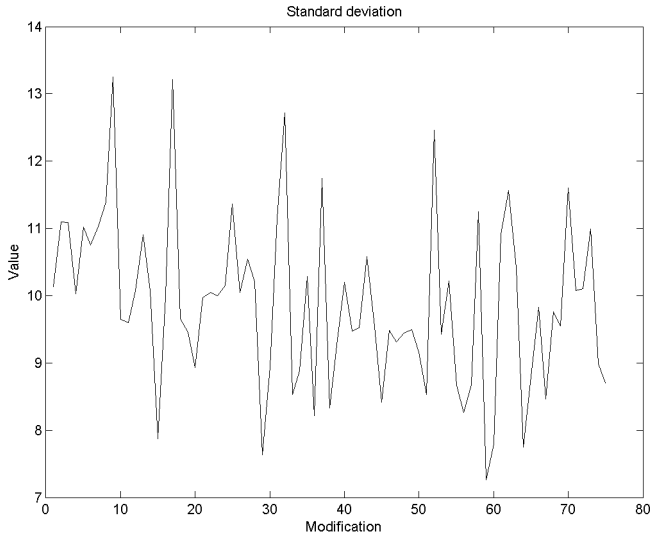

Figure 5.5: The standard deviations of the ratings of the cards measured when the modifications happened.

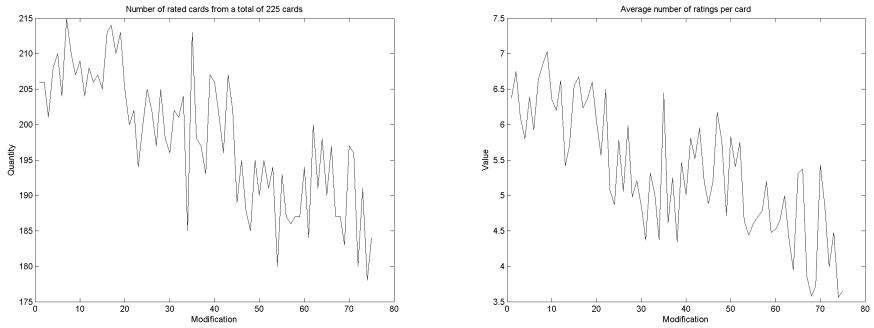

Figure 5.6: The number of average cards rated and the average number of ratings per card, both per modification.

5.2 User Feedback and Analysis

A playtesting session with eight participants was organized. After and during they tested the game the participants gave feedback.

The overall response was that the game concept is interesting, but the game itself needs refinement in terms of card balancing, more support for the user and fixes of problems which arose during the session. It was harder for the users to learn the basic mechanics of the game than expected.

This section covers mostly the feedback for the card balancing. Other suggestions of improvement will be treated in the future work chapter

It was often criticized that it takes a long time (several turns) until the first cards could be played. After the balancing with help of the AI, many cards became expensive and harder to play. On the other hand users also mentioned that some cards where too strong for their cost. One example was the card *observing amohjar-sea*. But when the final *observing amohjar-sea* card after balancing is compared to the initially generated card, the cards cost was raised from only two to five. So the card was detected to be too strong three times. The modification thus goes into the right direction, but the card would have needed a higher cost compared to other cards in the end according to the feedback.

Another reported issue was the random generation. It was suggested by the users that some combinations of attributes and effects in the random generation should be prevented. One example was that some effects have too many or too complicated conditions to be playable. This was especially the case with very specific conditions, such as that the own hero needs to have exactly 30 health or that the number of cards in the players hand needs to be five. A possible improvement could be that these conditions get excluded from the game.

Users found cards in the game which they estimated to be unplayable. One example was the spell *wasp-xifmed* which had a cost of two and would set the health of five random friendly characters to four. This card has a high chance of setting the own hero's health to four, which is very bad. This was not detected by the AI, in fact it was never modified since the initial generation of the card. A reason could be that the card did not affect many games outcomes by just being in the hand of players, which then could win even though one card in their hand was unplayable. Together with the many draw games due to too many turns played, this could lead to an average rating which is near zero and thus will not be modified.

Another card which was rated as unplayable by the playtesters was the spell *star-mawoxef*. This card has a cost of five, targets five random friendly minions and deals three damage to each of them. It is a prime example of a card which is unplayable, yet it was not removed from the set. When analyzing the change of the cost of *star-mawoxef* it can be seen that it changed from six to five. As for

observing amohjar-sea, the modification goes into the right direction and thus it was detected as a card which is too expensive, but this was not perceived often enough in order to mark this card as unplayable.

As a conclusion it can be said that the AI which was programmed during the project did not modify the cards enough for them to be balanced from a human point of view. Many cards lack a fair cost for their services: there are cards which are too expensive as well as cards which are too cheap and even unplayable cards in the final set. It could be that with more modifications the costs would be more accurate, but there is the possibility that the method has to be adapted in general.

Chapter 6

Future Work

6.1 Gameplay Elements

There are a lot of features which can be added, but had to be left out for this project due to the lack of time. Some of them are inspired by feedback received during the playtesting session. The following features can be added in a future version of the game:

- More triggers for effects of minions, such as when a minion dies or whenever you spawn another minion or play a spell.
- More effects, for example the possibility to spawn many minions at once with spells or with minion effects. New effects could be added in some kind of a new set of cards much like with other card games, which again can be randomly generated (including the new effects and possibly excluding some old ones) and balanced by the AI.
- A characterization of the land types, in terms of specific effects which only, or mostly occur for cards of a certain land type, as Hearthstone (in form of different classes)[11] or Magic: the Gathering (in form of colors) have these feature [18]. For example the ability to draw cards could be a common effect for water cards, but almost never be on cards of the type fire. In return, fire could have many effects which deal a lot of damage to characters. For this separation to be interesting, there would have to be enough different effects, such that they can be distributed fairly among the diffferent land types.
- To bring more variety into the different games, there could be added different boards, for example one with a big hole in the middle, or one which is double the size but with a lot of holes and one single bigger spot in the middle to be able to play the more expensive cards. This could change the gameplay a lot, for example with the last mentioned board, it would be more like a race which player gets to the center first and is then able to

defend that spot. With different boards, there may also be different cards which are advantageous over others.

- The game could be extended to allow more than only two players playing in one game. For example it could be interesting to have a four player all versus all match, where the last player surviving would win. Also team matches are possible, much like in Magic: the Gathering (Two-Headed Giant mode, which can be sometimes seen at tournaments)[19].
- Names which are randomly generated could be refined, such that there are two different generators, one for spells and one for minions. The reason is, that minions would rather always be creatures and spells not, which could be reflected by the names. Another improvement would be if the names were generated according to the land types needed for them to be played. For example only cards which need water type land would have words like *Water* or adjectives like *blue* in them.
- A timer could be added to limit the maximum duration of a turn, similar to Hearthstone and some variants of chess.

6.2 User Interface

As players want to be aware of what happens on the board and what the other player does, there could be more features which help the user to see what happens:

- Some kind of animation, when a character attacks another character. Examples of how this can be done are shown very well by Hearthstone[1], where attacking minions ram into the defending ones.
- More visual effects which are displayed on targets of spells or effects, such that the player does not have to extract all the information out of the information box while playing.
- Different sound effects could be added for different types of effects, for example when a minion dies or the attack of a minion gets boosted.
- When a card is selected a grid could be shown which indicates the cards cost and moves with the users mouse in order to better grasp where a card can be placed and where not.
- While a player is changing the type of a land, the details of the previously selected card could still be visible, such that it is not necessary to remember where the land should be changed in order to play this card.

6.3 Artificial Intelligence

The AI can deal with all of the effects, but it is also very simple. Improvements on the AI would highly improve the balancing of the cards.

Parts of the currently present decision-tree could possibly be cut off. For example for spells, there could be an improved distinction between the different effects and the targets we want to look at. When considering targets for a simple spell which deals damage to a minion, we most likely do not want to pursue what happens when this spell is played on a friendly minion. Distinctions like this are more involved, especially if there are many effects for a target, but can make the computation much faster, as the AI throws away more actions which will never result in any good state.

Nevertheless, improving or extending the AI can also be hard, due to the time limit we want to give it for taking a turn. Another possibility would be to allow more computation time for one turn, if enough time is available for balancing. An example of this is shown by Hearthsim[20], an AI for Hearthstone. Hearthstone has much more effects and also another type of cards than there are currently in this project, but on the other hand, there is no real board where minions can move around. Hearthsim simulates for 90 seconds, which is the time limit per turn for players of Hearthstone. This is much more computation time than the one second that the AI of this project currently uses per turn.

In its present state, the AI does never look at what the enemy could do next. An improvement would be to look, where the enemy could attack the next turn and do not walk there with minions which can get easily attacked and killed from enemy minions nearby, especially when they are not even able to defend themselves, for example due to range reasons.

It could be a strong extension for the AI to be able to guess what the enemy could do in the next turn out of the board, the land types in the enemy's deck or the number of cards in his hand. The AI could then counter certain cards or prevent the enemy from doing something. For example a minion could be moved to a place where the enemy will most likely spawn a minion next turn, such that the place is occupied and he cannot play the minion card at all the next turn.

Chapter 7

Conclusion

The goal of this bachelor thesis was to design and implement a game that is a mix between a CCG and a board game. Additionally an AI which can play the game was developed. The cards for the game were randomly generated and balanced based on the results of the AI playing many games. The modifications were then analyzed and playtested in order to reason about the quality of the balancing.

The modifications were not good enough to provide a balanced set of cards. The analysis of the modifications showed that the balancing could be incomplete which was then confirmed by the feedback from the playtesting. The method could possibly be improved by increasing the number of simulated games, but it could also be that it has to be adapted in general for the balancing to be accurate enough.

The concept of the game itself was received very positive and with the suggested improvements implemented, there are good chances the game could be more fun to play one day.

Developing games is a very involved process, as it combines many fields like for example visuals and networking. I never worked on an AI before this project and it took me some time until it did something useful. The game can still be polished, mainly in terms of visual feedback for users and also extended with a better AI and more cards and effects. It was very interesting to peek into making a game and the possible problems the companies which publish card and board games have to face. I am planning to look more into AI and different attempts, like for example artificial neural networks, as this first insights made me curious of what else could be possible.

Bibliography

[1] Blizzard: Hearthstone Main Page. http://us.battle.net/hearthstone/en/ Last checked 11.03.2015.

[2] StoneBlade Entertainment: Solforge. http://solforgegame.com/ Last checked 11.03.2015.

[3] Mojang: Scrolls. https://scrolls.com/ Last checked 11.03.2015.

[4] Wizards of the Coast: Magic: The Gathering Main Page. http://magic.wizards.com/en/magic-gathering Last checked 11.03.2015.

[5] http://icv2.com/: Magic: The Gathering popularity. http://icv2.com/articles/games/view/21471/magic-doubled-since-2008 Last checked 11.03.2015.

[6] Bit Mass LLC: Cards and Castles. http://www.cardsandcastles.com Last checked 11.03.2015.

[7] Abrakam: Faeria. http://www.faeria.net/ Last checked 12.03.2015.

[8] Tim Achee: Collectible Card Games. http://libguides.slu.edu/boardgames Last checked 13.03.2015.

[9] Wizards of the Coast: Magic: the Gathering Rulebook. http://media.wizards.com/images/magic/resources/rules/EN_MTGM14_PrintedRulebook_LR.pdf Last checked 11.03.2015.

[10] Wizards of the Coast: Magic: the Gathering history. http://archive.wizards.com/Magic/magazine/article.aspx?x=mtg/daily/feature/41a Last checked 11.03.2015.

[11] Blizzard: Heroes in Hearthstone. http://eu.battle.net/hearthstone/en/game-guide/heroes Last checked 11.03.2015.

[12] Laurent Gomila: SFML. http://www.sfml-dev.org/ Last checked 11.03.2015.

[13] TGUI: Texus' Graphical User Interface. https://tgui.eu/ Last checked 11.03.2015.

[14] GIMP: GNU Image Manipulation Program. http://www.gimp.org/ Last checked 12.03.2015.

[15] Mathworks: Matlab. http://ch.mathworks.com/products/matlab/ Last checked 12.03.2015.

[16] Wizards of the Coast: Magic: The Gathering Fanatic of Xenagos. http://gatherer.wizards.com/Pages/Card/Details.aspx?multiverseid=378519 Last checked 11.03.2015.

[17] Wizards of the Coast: Magic: The Gathering Warstorm Surge. http://gatherer.wizards.com/Pages/Card/Details.aspx?multiverseid=376581 Last checked 11.03.2015.

[18] Wizards of the Coast: The five colors in Magic: The Gathering. http://archive.wizards.com/Magic/multiverse/colors.aspx Last checked 11.03.2015.

[19] Wizards of the Coast: Magic: The Gathering Two Headed Giant. http://mtg.wikia.com/wiki/Two-Headed_Giant Last checked 11.03.2015.

[20] BuddyPanda: Hearthsim. http://buddypanda.com/hearthsim Last checked 12.03.2015.