

GWENT

BY NICK DEPAUL



TABLE OF CONTENTS

Gwent

Executive Summary

3

Entity Relationship Diagram

4

Tables

5

Reports

28

Views

39

Stored Procedures

44

Triggers

46

Security

48

Implementation Notes

50

Known Problems / Future Enhancements

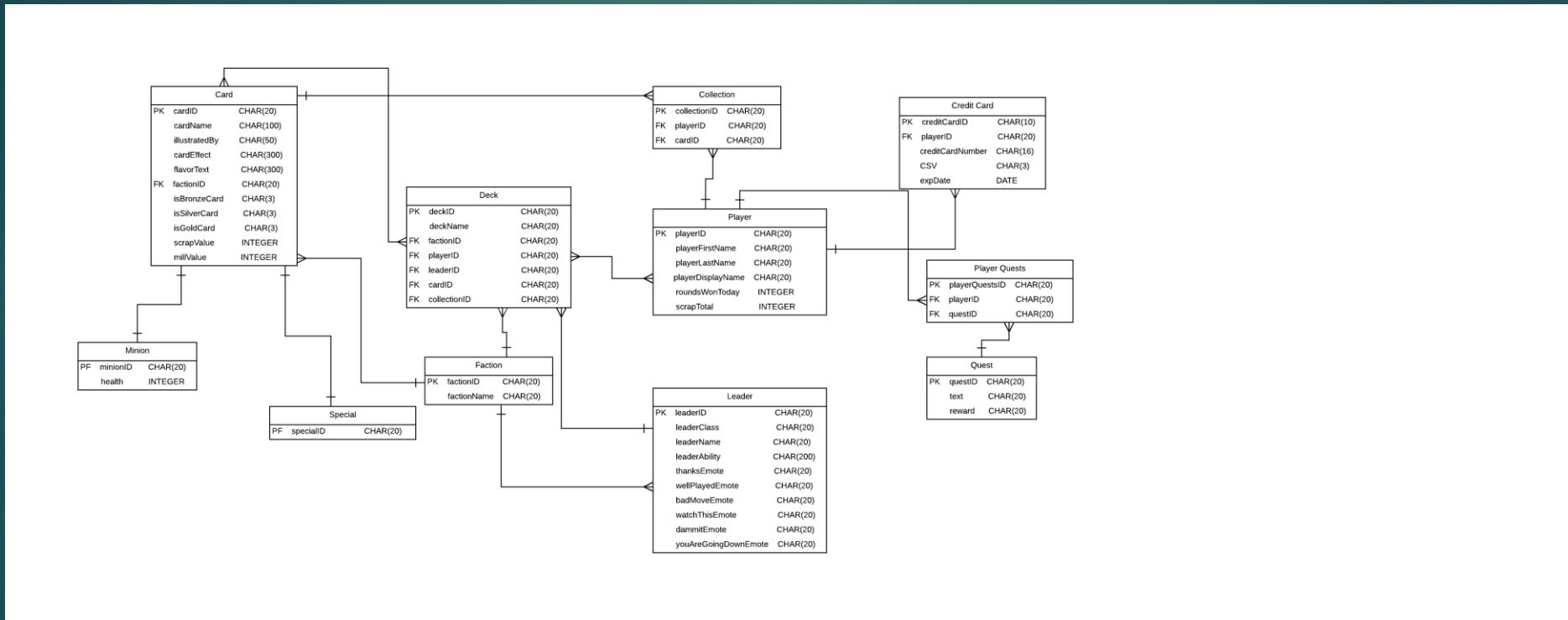
55



EXECUTIVE SUMMARY

Gwent is an online card game that utilizes the world of The Witcher, created by CD Projekt Red. In Gwent, each player has a collection of cards that they collect through in game currencies, like orens, or scrap. Players compete against each other in a best of three format, using decks created from cards in their collections. Every day Gwent will keep track on how many rounds the player has won, at the end of the day it will reset. Based on how many rounds you win, you will receive a reward of scraps that you can use to either craft more cards or open a card pack. Gwent maintains its long-term appeal by containing a variety of cards with different effects; these cards are further separated by factions, each of which has leaders with special abilities that allows each faction to have a different play style. As such, cards can be restricted as being in a faction, or available to all the factions. The database needs to record the effects of individual cards, including; health, effects, card artists, and flavor text. This Database is made to serve as a portrayal of the CD Projekt Red game, Gwent. It will provide an approximation of the fields that are required for the basic mechanics of the game, including the use of minions and special cards. The database will need to contain different cards, decks, collections possessed by players, leader, factions, credit card information, player information, and daily quests.

Entity Relationship Diagram





TABLES

Gwent

THE WITCHER CARD GAME

CARD

```
CREATE TABLE Card (
    cardID      CHAR(20),
    cardName     CHAR(100),
    illustratedBy CHAR(50),
    cardEffect   CHAR(300),
    flavorText   CHAR(300),
    factionID   CHAR(20),
    isBronzeCard CHAR(3),
    isSilverCard CHAR(3),
    isGoldCard   CHAR(3),
    scrapValue   INTEGER,
    millValue    INTEGER,
    PRIMARY KEY(cardID)
);
```

cardid character (20)	cardname character (100)	illustratedby character (50)	cardeffect character (300)	flavortext character (300)	factionid character (20)	isbronzeard character (3)	issilvercard character (3)	isgoldcard character (3)	scrapvalue integer	millvalue integer
1	Nekker	Katarzyna Red...	Whenever an Al...	These little guy...	1	Yes	[null]	[null]	80	20
2	Wild Hunt Navig...	Diego De Alme...	Deploy: Choose...	For hundreds of...	1	Yes	[null]	[null]	80	20
3	Foglet	Bartlomiej Ga...	Whenever you ...	Fog creeps on li...	1	Yes	[null]	[null]	30	10
4	Ghoul	Adrian Smith	Deploy: Consu...	If ghouls are pa...	1	Yes	[null]	[null]	30	10
5	Wild Hunt Hound	Marek Madej	Deploy: Play a ...	Cry Havoc!, and...	1	Yes	[null]	[null]	80	20
6	Ice Giant	Bartlomiej Ga...	If a Frost Hazar...	Fled one time in...	1	Yes	[null]	[null]	80	20
7	Wyvern	Alejandro Mira...	Deploy: Damag...	Imagine a cross...	1	Yes	[null]	[null]	30	10
8	Wild Hunt Warri...	Marta Dettlaff	Deploy: Damag...	The White Frost...	1	Yes	[null]	[null]	30	10
9	Ekimmara	Bartlomiej Ga...	Deploy: Gain Re...	Who would thin...	1	Yes	[null]	[null]	80	20
10	Drowner	Adrian Smith	Deploy: Move a ...	Though the wit...	1	Yes	[null]	[null]	30	10
11	Wild Hunt Long...	Marek Madej	While on the Bo...	Waves pummel ...	1	Yes	[null]	[null]	80	20
12	Archgriffin	Anna Podedw...	Deploy: Clear H...	It is a griffin, ju...	1	Yes	[null]	[null]	30	10
13	Wild Hunt Rider	Marek Madej	Increase by 1 t...	First the buffalo...	1	Yes	[null]	[null]	30	10
14	Crone: Whispess	Bartlomiej Ga...	Deploy: Play all ...	I would be your...	1	[null]	Yes	[null]	200	50
15	Crone: Weavess	Bartlomiej Ga...	Deploy: Play all ...	I sense your pa...	1	[null]	Yes	[null]	200	50
16	Crone: Brewess	Bartlomiej Ga...	Deploy: Play all ...	We will cut you ...	1	[null]	Yes	[null]	200	50
17	Imlerith	Marek Madej	Deploy: Damag...	Ladd nahw! Kill ...	1	[null]	[null]	Yes	800	200
18	Caranthir	Marek Madej	Deploy: Move a ...	A favorite son ...	1	[null]	[null]	Yes	800	200
19	Draug	Nemanja Stan...	Deploy: Spawn ...	Some men cann...	1	[null]	[null]	Yes	800	200
20	Caretaker	Marek Madej	Deploy: Resurr...	There are more...	1	[null]	[null]	Yes	800	200
21	Alan Labouseur	Nick DePaul	Deploy: Gains 2...	It only has to b...	2	[null]	[null]	Yes	800	200
22	Herbalist	Lorenzo Mastr...	Deploy: Play a r...	We do not burn...	2	Yes	[null]	[null]	80	20
23	Heymaey Battle...	Grafit Studio	Deploy: Choose...	Skellige women...	2	Yes	[null]	[null]	80	20
24	Clan Drummond	Grafit Studio	Deploy: Damag...	They will chatte...	2	Yes	[null]	[null]	80	20

MINION

```
CREATE TABLE Minion (  
    minionID      CHAR(20),  
    health        INTEGER,  
    PRIMARY KEY(minionID)  
);
```

	minionid character (20)	health integer
1	1	3
2	2	3
3	3	4
4	4	4
5	5	4
6	6	6
7	7	6
8	8	6
9	9	6
10	10	7
11	11	7
12	12	8
13	13	9
14	14	6
15	15	6
16	16	8
17	17	9
18	18	8

SPECIAL

```
CREATE TABLE Special (
    specialID          CHAR(20),
    PRIMARY KEY(specialID)
);
```

specialid
character (20)
41
42
43
44
45
46
47
48
49
50

DECK

```
CREATE TABLE Deck (  
    deckID      CHAR(20),  
    deckName    CHAR(20),  
    factionID   CHAR(20),  
    playerID    CHAR(20),  
    leaderID    CHAR(20),  
    cardID      CHAR(20),  
    collectionID CHAR(20),  
    PRIMARY KEY(deckID)  
);
```

deckid character (20)	deckname character (20)	factionid character (20)	playerid character (20)	leaderid character (20)	cardid character (20)	collectionid character (20)
1	Monster	1	1	1	1	1
2	Monster	1	1	1	2	2
3	Monster	1	1	1	3	3
4	Monster	1	1	1	4	4
5	Monster	1	1	1	5	5
6	Monster	1	1	1	6	6
7	Monster	1	1	1	7	7
8	Monster	1	1	1	8	8
9	Monster	1	1	1	9	9
10	Monster	1	1	1	10	10
11	Monster	1	1	1	11	11
12	Monster	1	1	1	12	12
13	Monster	1	1	1	13	13
14	Monster	1	1	1	14	14
15	Monster	1	1	1	15	15
16	Monster	1	1	1	16	16
17	Monster	1	1	1	17	17
18	Monster	1	1	1	18	18
19	Monster	1	1	1	19	19
20	Monster	1	1	1	20	20
21	Monster	1	1	1	41	21

FACTION

```
CREATE TABLE Faction (  
    factionID      CHAR(20),  
    factionName    CHAR(20),  
    PRIMARY KEY(factionID)  
);
```

factionid character (20)	factionname character (20)
1	Monster
2	Skellige

COLLECTION

```
CREATE TABLE Collection (  
    collectionID      CHAR(20),  
    playerID          CHAR(20),  
    cardID            CHAR(20),  
    PRIMARY KEY(collectionID)  
)
```

collectionid character (20)	playerid character (20)	cardid character (20)
1	1	1
2	1	2
3	1	3
4	1	4
5	1	5
6	1	6
7	1	7
8	1	8
9	1	9
10	1	10
11	1	11
12	1	12
13	1	13
14	1	14
15	1	15
16	1	16
17	1	17
18	1	18

PLAYER

```
CREATE TABLE Player (  
    playerID      CHAR(20),  
    playerFirstName   CHAR(20),  
    playerLastName    CHAR(20),  
    playerDisplayName  CHAR(20),  
    roundsWonToday    INTEGER,  
    scrapTotal       INTEGER,  
    PRIMARY KEY(playerID)  
)
```

playerid character (20)	playerfirstname character (20)	playerlastname character (20)	playerdisplayname character (20)	roundswontoday integer	scraptotal integer
1	Nick	DePaul	GwentMaster	20	1024
2	Chris	Algozzine	ISIstheBest	35	12943

LEADER

```
CREATE TABLE Leader (
    leaderID          CHAR(20),
    leaderClass       CHAR(20),
    leaderName        CHAR(20),
    leaderAbility     CHAR(200),
    thanksEmote       CHAR(20),
    wellPlayedEmote   CHAR(20),
    badMoveEmote      CHAR(20),
    watchThisEmote    CHAR(20),
    dammitEmote       CHAR(20),
    youAreGoingDownEmote CHAR(20),
    PRIMARY KEY(leaderID)
);
```

leaderid character (20)	leaderclass character (20)	leadername character (20)	leaderability character (200)	thanksemote character (20)	wellplayedemote character (20)	badmoveemote character (20)	watchthisemote character (20)	dammitemote character (20)	youaregoingdownemote character (20)
1	Monster	Eredin	Doomed: Spaw...	Thanks	Well Played	Bad Move	Watch This	Dammit	You Are Going Down
2	Skellige	King Bran	Deploy: Discard...	Thanks	Well Played	Bad Move	Watch This	Dammit	You Are Going Down

CREDIT CARD

```
CREATE TABLE CreditCard (
    creditCardID      CHAR(10),
    playerID          CHAR(20),
    creditCardNumber   CHAR(16),
    CSV                CHAR(3),
    expDate           DATE,
    PRIMARY KEY(creditCardID)
);
```

	creditcardid character (10)	playerid character (20)	creditcardnumber character (16)	csv character (3)	expdate date
1	1	1	80027476001283...	300	2018-0...
2	2	2	47720377712034...	120	2020-1...

PLAYER QUESTS

```
CREATE TABLE PlayerQuests (
    playerQuestsID      CHAR(20),
    playerID            CHAR(20),
    questID             CHAR(20),
    PRIMARY KEY(playerQuestsID)
);
```

playerquestsid character (20)	playerid character (20)	questid character (20)
1	1	1
2	1	2
3	2	1
4	2	2

QUEST

```
CREATE TABLE Quest (  
    questID      CHAR(20),  
    text         CHAR(20),  
    reward       CHAR(20),  
    PRIMARY KEY(questID)  
)
```

questid character (20)	text character (20)	reward character (20)
1	Win 2 Rounds	15 scrap
2	Win 4 Rounds	15 scrap

REPORTS

This will show all the factions that the minions belong to and it displays their health.

```
SELECT      Card.cardName, Faction.factionName, Minion.health  
FROM        Card  
RIGHT JOIN Faction ON Card.factionID = Faction.factionID  
RIGHT JOIN Minion ON Card.cardID      = Minion.minionID  
ORDER BY    Faction.factionName;
```

	cardname character (100)	factionname character (20)	health integer
1	Nekker	Monster	3
2	Wild Hunt Navig...	Monster	3
3	Foglet	Monster	4
4	Ghoul	Monster	4
5	Wild Hunt Hound	Monster	4
6	Ice Giant	Monster	6
7	Wyvern	Monster	6
8	Wild Hunt Warri...	Monster	6
9	Ekimmara	Monster	6
10	Drowner	Monster	7
11	Wild Hunt Long...	Monster	7
12	Archgriffin	Monster	8
13	Wild Hunt Rider	Monster	9
14	Crone: Whispess	Monster	6
15	Crone: Weavess	Monster	6
16	Crone: Brewess	Monster	8
17	Imlerith	Monster	9
18	Caranthur	Monster	8
19	Draug	Monster	8
20	Caretaker	Monster	4
21	Alan Labouseur	Skellige	8
22	Herbalist	Skellige	2

This displays all of the cards information in the deck

```
SELECT      Deck.deckName, Faction.factionName,  
Player.playerLastName, Leader.leaderName,  
          Card.cardName  
FROM        Deck  
RIGHT JOIN  Faction ON Deck.factionID = Faction.factionID  
RIGHT JOIN  Player  ON Deck.playerID   = Player.playerID  
RIGHT JOIN  Leader   ON Deck.leaderID  = Leader.leaderID  
RIGHT JOIN  Card    ON Deck.cardID    = Card.cardID  
ORDER BY    Deck.deckName;
```

deckname character (20)	factionname character (20)	playerlastname character (20)	leadername character (20)	cardname character (100)
Monster	Monster	DePaul	Eredin	Nekker
Monster	Monster	DePaul	Eredin	Wild Hunt Navig...
Monster	Monster	DePaul	Eredin	Foglet
Monster	Monster	DePaul	Eredin	Ghoul
Monster	Monster	DePaul	Eredin	Wild Hunt Hound
Monster	Monster	DePaul	Eredin	Ice Giant
Monster	Monster	DePaul	Eredin	Wyvern
Monster	Monster	DePaul	Eredin	Wild Hunt Warri...
Monster	Monster	DePaul	Eredin	Ekimmara
Monster	Monster	DePaul	Eredin	Drowner
Monster	Monster	DePaul	Eredin	Wild Hunt Long...
Monster	Monster	DePaul	Eredin	Archgriffin
Monster	Monster	DePaul	Eredin	Wild Hunt Rider
Monster	Monster	DePaul	Eredin	Crone: Whispess
Monster	Monster	DePaul	Eredin	Crone: Weavess
Monster	Monster	DePaul	Eredin	Crone: Brewess
Monster	Monster	DePaul	Eredin	Imlerith
Monster	Monster	DePaul	Eredin	Caranthir
Monster	Monster	DePaul	Eredin	Draug
Monster	Monster	DePaul	Eredin	Caretaker
Monster	Monster	DePaul	Eredin	Impenetrable F...
Monster	Monster	DePaul	Eredin	Biting Frost

This shows what the player has in their collection

```
SELECT      Player.playerLastName, Card.cardName  
FROM        Collection  
RIGHT JOIN Player ON Collection.playerID = Player.playerID  
RIGHT JOIN Card   ON Collection.cardID   = Card.cardID  
ORDER BY    Player.playerLastName;
```

playerlastname character (20)	cardname character (100)
Algozzine	Skjall
Algozzine	Morkvarg
Algozzine	Jutta An Dimun
Algozzine	Udalryk
Algozzine	Birna Bran
Algozzine	Ermion
Algozzine	Hjalmar
Algozzine	Lacerate
Algozzine	Mahakam Ale
Algozzine	First Light
Algozzine	Swallow Potion
DePaul	Nekker
DePaul	Wild Hunt Navig...
DePaul	Foglet
DePaul	Ghoul
DePaul	Wild Hunt Hound
DePaul	Ice Giant
DePaul	Wyvern
DePaul	Wild Hunt Warri...
DePaul	Ekimmara
DePaul	Drowner

This shows the player's credit card information

- ▶ `SELECT Player.playerLastName, CreditCard.creditCardNumber, CreditCard.CSV, CreditCard.expDate`
- ▶ `FROM CreditCard`
- ▶ `RIGHT JOIN Player ON CreditCard.playerID = Player.playerID`
- ▶ `ORDER BY Player.playerLastName;`

playerlastname character (20)	creditcardnumber character (16)	csv character (3)	expdate date
Algozzine	4772037771203486	120	2020-12-15
DePaul	8002747600128364	300	2018-01-20

This shows the daily quests that the player has completed

```
SELECT      Player.playerLastName, quest.text, quest.reward  
FROM        PlayerQuests  
RIGHT JOIN  Player ON PlayerQuests.playerID = Player.playerID  
RIGHT JOIN  Quest  ON PlayerQuests.questID  = Quest.questID  
ORDER BY    Player.playerLastName;
```

playerlastname character (20)	text character (20)	reward character (20)
Algozzine	Win 2 Rounds	15 scrap
Algozzine	Win 4 Rounds	15 scrap
DePaul	Win 2 Rounds	15 scrap
DePaul	Win 4 Rounds	15 scrap



Gwent

THE WITCHER CARD GAME

```
CREATE VIEW CurrentPlayers  
AS  
SELECT      playerDisplayName  
FROM        Player;  
SELECT      *  
FROM        CurrentPlayers  
ORDER BY    playerDisplayName ASC;
```

SHOWS THE CURRENT PLAYERS

playerid character (20)	playerdisplayname character (20)
1	GwentMaster
2	ISIstheBest

```
CREATE VIEW LeaderClass  
AS  
SELECT      leaderClass, leaderName, leaderAbility  
FROM        Leader;  
SELECT      *  
FROM        LeaderClass  
ORDER BY    leaderClass ASC;
```

SHOWS THE LEADERS CLASS, NAME, AND ABILITY

leaderclass character (20)	leadername character (20)	leaderability character (200)
Monster	Eredin	Doomed: Spawn a Bronze Wild Hunt Unit.
Skellige	King Bran	Deploy: Discard up to 3 cards from your Deck and Strengthen all Units among them by 1. Then shuffle your Deck.



STORED PROCEDURES

ADD EVERY ROUND A PLAYER HAS WON WHEN THEY WIN A ROUND

```
CREATE OR REPLACE FUNCTION playerRoundsWon()
RETURNS TRIGGER AS $$

DECLARE

BEGIN

IF roundsWonToday > 0 THEN

SET roundsWonToday = 1;

END IF;

RETURN new;

END;

$$ language plpgsql;
```



TRIGGERS

This is a continuation based off of my Store Procedure

```
CREATE TRIGGER playerRoundsWon  
BEFORE UPDATE  
ON Player  
FOR EACH ROW  
EXECUTE PROCEDURE playerRoundsWon();
```

A dramatic, high-energy scene from The Witcher card game. In the center, Geralt of Rivia stands in a dynamic pose, his right hand held forward as if casting a spell. A massive, jagged, blue energy shield surrounds him, crackling with power. Behind him, a large, dark, ethereal creature with glowing blue eyes and a skeletal face looms, partially obscured by the energy field. The background is filled with swirling, colorful energy particles in shades of blue, purple, and white, suggesting a powerful magical confrontation. The overall atmosphere is one of intense magic and combat.

GWENT

SECURITY

THE WITCHER CARD GAME

ONLY GRANTS THE PLAYER ACCESS TO SELECT ON ALL OF THE TABLES. THE ADMIN CAN SELECT, INSERT AND UPDATE ON ALL OF THE TABLES. OF COURSE THE ADMIN WILL NEED A PASSWORD. THE PASSWORD IS ALWAYS ALPACA.

```
CREATE ROLE Player;  
GRANT SELECT ON ALL Tables IN SCHEMA PUBLIC TO Player;
```

```
CREATE ROLE Admin WITH Login Password 'alpaca';  
GRANT SELECT, INSERT, UPDATE ON ALL Tables IN SCHEMA PUBLIC TO Admin;
```

Implementation Notes

The set-up of the database demonstrates the importance of the card and deck concepts within this system, and uses the player table to gathering personal information and related data to them in-game performance and collection.

The card idea is at the heart of Gwent, and this is made clear on how over half the tables in the system are tied to the card table directly. In this case, the card table holds game relevant information (stats and effects) but also displays information, such as rarity and funny card descriptions. These details can be used to encourage in-game purchases, to obtain cards ranging from the common bronze card to the difficult to get gold cards. It also will help the player grow an attachment to their collection as they have an ever-growing number of cards reflecting the time and effort they put in.

Implementation Notes Continued

The way the card set is used in Gwent is that it's designed to tell the player what each card's functionality and role is. Each card can be broken into two different categories, minions and special cards. Minions are creatures or people that are on the board and serve to provide the scores that determine the winners of each round. Special cards such as weather effects, can drastically change the course of the round, which are used to interact with the board without placing a minion. While these special cards have no unique values, creating a separate category offers potential to expand the idea; such as effects that will only work under certain requirements. Additionally, adding tables specifically for special cards and minions helps prevent every card from having multiple null values as type-specific values are prohibited. It will also decrease errors from filling in the wrong values, for example, filling in a health cost for a card that does not have that value, such as a weather effect card.

Implementation Notes Continued

Each card can either be used by all factions or are limited to a specific faction. Each faction has unique cards, abilities, and effects, which promotes different strategies. This encourages players to pick a faction that best suits their style of play. This creates a new level of depth in the product without adding anything that is too hard to understand. The faction table creates a place to refer to in order to separate the cards that fall under their distinct set.

Each faction is represented by multiple leaders, that have unique personalities which is characterized by emotes that can be spoken in game, and are further defined by specific abilities, even for leaders within the same faction. For example, an aggressive leader may use a nasty greeting, while a more relaxed one would be more pleasant. Also, the more aggressive leader could have an ability that will destroy an opponent's card, while the relaxed one could pick a random card from their deck and play it.

Implementation Notes Continued

Each of these factors come together in the concept of a deck. The implementation of the deck concept was one that went through several versions. The final execution is a combination of related tables giving the sum-total of a deck's value of cards, faction and leader selection. The deck is the main method by which players interact with the game, and is the core of Gwent.

An additional feature of the game lies in quests, which can be used to acquire in-game resources to get cards without spending real-world money. The PlayerQuest table ties quests to the players that have them. This allows a free-to-play style, and many players will be encouraged to pay real-world money to obtain cards faster and allow them to complete their collection of cards a lot quicker.

Implementation Notes Continued

As of right now, the player field is only being used to store information and a way to bring together assets, such as decks, card collection, and quests and tie them to a single player. However, this could store the in-game statistics for each player faction; such as rounds that were won, overall win rates, and storing records of games to allow players to study the way they play. While this is really limited as of right now only storing the number of games won daily, there is a lot more that can be done to enhance the game with this table. Providing tools that will offer the player a means by which to invest themselves in the game, and is another way to encourage free-to-play players to start using real-world money to speed the rate by which they earn their card collection.

Known Problems / Future Enhancements

One enhancement that they should make to this database would be adding a set of tables for in game effects such as rounds, and players. This change would record information and allow an analysis of gameplay to give the developers and players information in order to change tactics and balance cards more effectively. The current system does not give us this data, weakening decision making which will have users going to third party applications to improve their performance.

The player entity is not being used enough for storing information based on player behavior. If additional information is stored, including play times and activity, this will help in going after users who may be losing interest in the game by offering encouragement to continue playing; such as giving a bonus after a certain amount of playtime.

Known Problems / Future Enhancements Continued

Adding additional information within cards, including the card set and rate of activities performed, such as scrapping could also give information. This could be valuable information for the developer. If a certain amount of players are constantly scrapping the same card then the developer may want to change the effects on that card. Or if a card is winning too often then they may want to weaken it.

Gwent could take cues from their rival Hearthstone in player deck customization. To enhance customization, they could allow the player to create decks they have created. Right now, they only call your deck based on whatever faction it belongs to. This could help prevent players losing interest in the game because it will allow them to grow an attachment to their decks.