



**TOSHKENT SHAHRIDAGI INHA UNIVERSITETI
INHA UNIVERSITY IN TASHKENT**

Database Application and Design (SOC-3060) – Spring 2019

Database System Report

for

The Uzbek Tragedy

Prepared by Anvarjon Yusupov (u1610026)
Jamoliddinkhuja Odilkhujaev (U1610092)
Mokhlaroyim Tuychibaeva* (U1610148)

Davinci Union

May 7, 2019

1. Introduction and scope

Proposed project name is called "**The Uzbek Tragedy**" dedicated to discovering mountains in Uzbekistan with its magnificent nature, animals and plants in form of the game. The Uzbek Tragedy is a sub-type of game genre Visual Novel.

Visual novel - is interactive game genre oriented on story-telling with the narrative style of literature and interacting with the player via sprites, choices and etc. Visual novels are often produced for video game consoles, and rarely ported to other systems

The product will entertain users by an astonishing story with 4 main characters. Where the fate of characters will depend on the players choice. Moreover, the product will show to users Uzbek nature in its pure serenity. The animals and plants represented in the project are animals that truly inhabit in Uzbek territory. Plants and animals can be dangerous or useful, for example: if player will see wolf, definitely this is dangerous animal, while if it squirrel, animal cannot harm player, that applies to plants too, some plant can be poisonous or helpful, and it depends on players choice how to use it, because sometimes even poison can be helpful.

The game will attract end users with its astonishing story, the whole story takes place around 4 medical university junior students, who got into trouble, in Uzbek mountains, while they were going to have several lectures about healing herbs. The main character's name is August, he will survive after the school bus crash, so his main goal is to survive and find way home. While he travels he can find his friends and survive together or he can try to save himself and sacrifice his friends. August will have an encyclopedia, a book from his studies with various information about herbs, so if he will see unknown herb, he can reveal if it can help him or harm. Every choice of August will lead to certain consequences, so if he wants to survive, he can't be hero of justice, he has to choose who will live and who will die.

Interface part of the project will be by Unity. Unity 3D is a cross-platform game engine and integrated development environment (IDE) for creating interactive media. It enabled the game app development for different platforms and distinct devices in a user-friendly development environment. It used to build games and the technology that executes the graphics, audio, interactions, and networking[1]. The relevance of using unity is that Unity is multi-platform and has all-in-one editor, relatively building project to any platform much easier.

2. Product perspective

2.1 Comparison

The Uzbek Tragedy is one of the Visual Novel game genres, therefore there is many competitors in the world

Most popular example is Fate Stay night[2], Katawa Shoujo[3], and Everlasting Summer[4]

Here is Competitors table, which compares several similar products with the Uzbek Tragedy

	Uzbek Tragedy	Fate	Katawa shoji	Everlasting summer
Framework	Unity	-	RenPy	RenPy
Interface	User-friendly	User-friendly	User-friendly	User-friendly
Database	Yes	No	No	No
Multiplatform	Windows, Android, Mac, Linux	No	Windows, Android	Windows, Android

2.2 Product perspective

The Uzbek tragedy is a unique product because only this project is using Database for his needs. Moreover, it is multi-platform. Because, The Uzbek Tragedy, has his own unique design, based on mountains of Uzbekistan and containing animals from Uzbekistan. This project uses SQLite database management system instead of MySQL for database needs because for project usage Mysql would be too heavy. SQLite self-contained, file-based database with faster speed, the game doesn't need a big amount of data, so light-weighted DBMS the best option here.

As it is mentioned above, interface part of the project will be by Unity. Unity 3D is a cross-platform game engine and integrated development environment (IDE) for creating interactive media. It enabled the game app development for different platforms and distinct devices in a user-friendly development environment. It used to build games and the technology that executes the graphics, audio, interactions, and networking[1]. The relevance of using unity is that Unity is multi-platform and has all-in-one editor, relatively building project to any platform much easier.

3. Specific requirements

3.1 Logical database requirements

No	Entity	Explanation	Identify	Attributes	Note	Sample Values	Relationship with
1	Characters	Storing basic data about characters in game	Strong	ID	INTEGER, PRIMARY, NOT NULL	1,2,3	
				Name	VARCHAR(20)	"James", "Tiger"	
2	NPC (Non-Playable Character)	Stores specific data about non-playable characters	Weak	NPC_ID	INTEGER, PRIMARY, FOREIGN, NOT NULL	1,2,3	Characters
				Info	VARCHAR(2000)	"Tigers are"	
				isAlive	INTEGER	0/1	
				Properties	VARCHAR(20)	"DANGEROUS"/ "NOT DANGEROUS"	
3	Images	Stores Images that represent NPC (It was multivalued attribute on conceptual level)	Weak	State_Img	VARCHAR(20), PRIMARY, NOT NULL	"Tiger.png", "Zod_Angry.png"	
				NPC_ID	INTEGER, PRIMARY, FOREIGN, NOT NULL	1,2,3	NPC
4	Player	Keeps specific information about player	Weak	P_ID	INTEGER, PRIMARY, FOREIGN, NOT NULL	1,2,3	Characters
				Name	VARCHAR(20), NOT NULL	"Anvar", "Jamil", "Mokhlar"...	
				Score	INTEGER, NOT NULL	0, 7, 15, 25	
				A_ID	INTEGER, NOT NULL	1,2,3	Achievements
5	Flags	Stores data about Players decisions	Strong	F_ID	INTEGER, PRIMARY, NOT NULL	1,2,3	
				Name	VARCHAR(20), NOT NULL	"Healing", "Wolf"...	
				Count	INTEGER	1,2,3,4,5..	
6	GameResult	Stores results of game. Thus are activated flags and ending of game. Aggregated from Flags, Endings and Result	Weak	F_ID	INTEGER, PRIMARY, FOREIGN, NOT NULL	1,2,3	Flags
				E_ID	INTEGER, PRIMARY, FOREIGN, NOT NULL	1,2,3	Endings
				Flag_Sum	INTEGER, DERIVED	15,20,...	

7	Endings	Stores all endings and boolean that says which one is happened	Strong	E_ID	INTEGER, PRIMARY, NOT NULL	1,2,3	
				Happened	INTEGER,NOT NULL	0/1	
8	Achievements	Stores all Achievements of player	Strong	A_ID	INTEGER, PRIMARY, NOT NULL	1,2,3	
				AchieveName	VARCHAR(20)	"HEALER", "BRAVE"	
				Cost	INTEGER	3,5,7	
				Info	VARCHAR(255)	"You defeated wolf" ..	
				Achieved	INTEGER	1/0	
				F_ID	INTEGER, FOREIGN	1,2,3	Flags, GameResult
				E_ID	INTEGER, FOREIGN	1,2,3	GameResult
9	Inventory	Stores data about items in Inventory	Weak	I_ID	INTEGER, PRIMARY, FOREIGN	1,2,3	Plants, Objects
				Charge	INTEGER	1,2,3	
10	Plants	Stores specific data about plants	Weak	ID	INTEGER, PRIMARY, FOREIGN	1,2,3	Items
				Properties	VARCHAR(20)	"DANGEROUS"/ "NOT DANGEROUS"	
11	Objects	Stores specific information about objects	Weak	ID	INTEGER, PRIMARY, FOREIGN	1,2,3	Items
				Properties	VARCHAR(20)	"DANGEROUS"/ "NOT DANGEROUS"	
				TL	INTEGER		
12	Items	Stores basic data about items(Object and Plant is included)	Strong	ID	INTEGER, PRIMARY, NOT NULL	1,2,3	
				Name	VARCHAR(20), NOT NULL	"Ferula Stinky", "Astragalus" ..	
				Info	VARCHAR(1500)	"Ferula stinky is .."	
				ItemType	INTEGER	1/2	
				Image	VARCHAR(255)	"Ferula.png"	
13	Obtains	Stores ID of player and his achievements	Weak	P_ID	INTEGER, PRIMARY, FOREIGN	1,2,3	Player
				A_ID	INTEGER, PRIMARY, FOREIGN	1,2,3	Achievements

No	Relationship	Design Characteristics	Design Value	Note	Sample Values
1	Reaches	Meaning	Player reaches Flags		
		Type	Non-identifying		
		Parent Entity Set and Participation (min, max)	Player(O,M)		
		Parent Entity Set and Participation (min, max)	Flags(0,N)		
		Descriptive Attributes			
2	Has	Meaning	Player has Inventory		
		Type	Non-identifying		
		Parent Entity Set and Participation (min, max)	Player(1,1)		
		Parent Entity Set and Participation (min, max)	Inventory(1,1)		
		Descriptive Attributes			

3	Contain_Plants	Meaning	Inventory Contains Plants		
		Type	Identifying		
		Parent Entity Set and Participation (min, max)	Inventory(0,M)		
		Parent Entity Set and Participation (min, max)	Plants(0,1)		
		Descriptive Attributes			
4	Contain_Objects	Meaning	Inventory Contains Objects		
		Type	Identifying		
		Parent Entity Set and Participation (min, max)	Inventory(0,M)		
		Parent Entity Set and Participation (min, max)	Objects(0,1)		
		Descriptive Attributes			
5	Obtains	Meaning	Player obtains Achievements		
		Type	Non-identifying		
		Parent Entity Set and Participation (min, max)	Player(0,M)		
		Parent Entity Set and Participation (min, max)	Achievements(1,N)		
		Descriptive Attributes			
6	Achieves	Meaning	After gaining the GameResult, Player Achieves Achievement		
		Type	Non-Identifying		
		Parent Entity Set and Participation (min, max)	GameResults(1,1)		
		Parent Entity Set and Participation (min, max)	Achievements(0,1)		
		Descriptive Attributes			
7	Leads	Meaning	Flags leads to Achievements		
		Type	Non-identifying		
		Parent Entity Set and Participation (min, max)	Flags(0,1)		
		Parent Entity Set and Participation (min, max)	Achievements(1,1)		
		Descriptive Attributes			
8	Results	Meaning	Flags results Ending		
		Type	Non-identifying		
		Parent Entity Set and Participation (min, max)	Flags(1,M)		
		Parent Entity Set and Participation (min, max)	Endings(1,N)		
		Descriptive Attributes			

3.2 Functional database requirements

- 1) The system shall have characters with id and name attributes, also characters table should have player and NPC childs.
- 2) The system shall store information about players: score.
- 3) The system shall store information about nonplayable characters:
- 4) The system should know, if the nonplayable character is alive or not, history of the character and the path to his images.
- 5) The player should have inventory(encyclopedia).
- 6) There exists items which has plants and objects. Items has info about ID, name, images, item type, info,.
- 7) Plants have their own properties. Plant can heal, hurt, be eaten or do nothing.
- 8) Objects also have their properties.
- 9) In encyclopedia there can be info about plants and objects which player have seen
- 10) Player can reach some flags
- 11) Flags have their own ID, Name and Count. Count is the count of flag, how much does player reached or used this flag.
- 12) After getting needed amount of flag count, player gets to the ending.
- 13) Ending have unique attribute happened. Which indicates does, this ending happened or not.
- 14) System shall store result from flags and endings in flagsum derived attribute
- 15) System shall have aggregated table game results where result from flags and endings are stored.
- 16) Happening of ending will lead to the game result, which will show what achievements will Player get.
- 17) Achievements has attributes, id, name ,achieved, info, and cost. Cost will show the score of the player in the end.
- 18) Also some flags can bring to achievements to. For example: Getting death flag will bring to achievement “Death”
- 19) Player can obtain achievements.
- 20) Inventory should be weak and depend on plants and objects.

4. Design

4.1 Conceptual database design

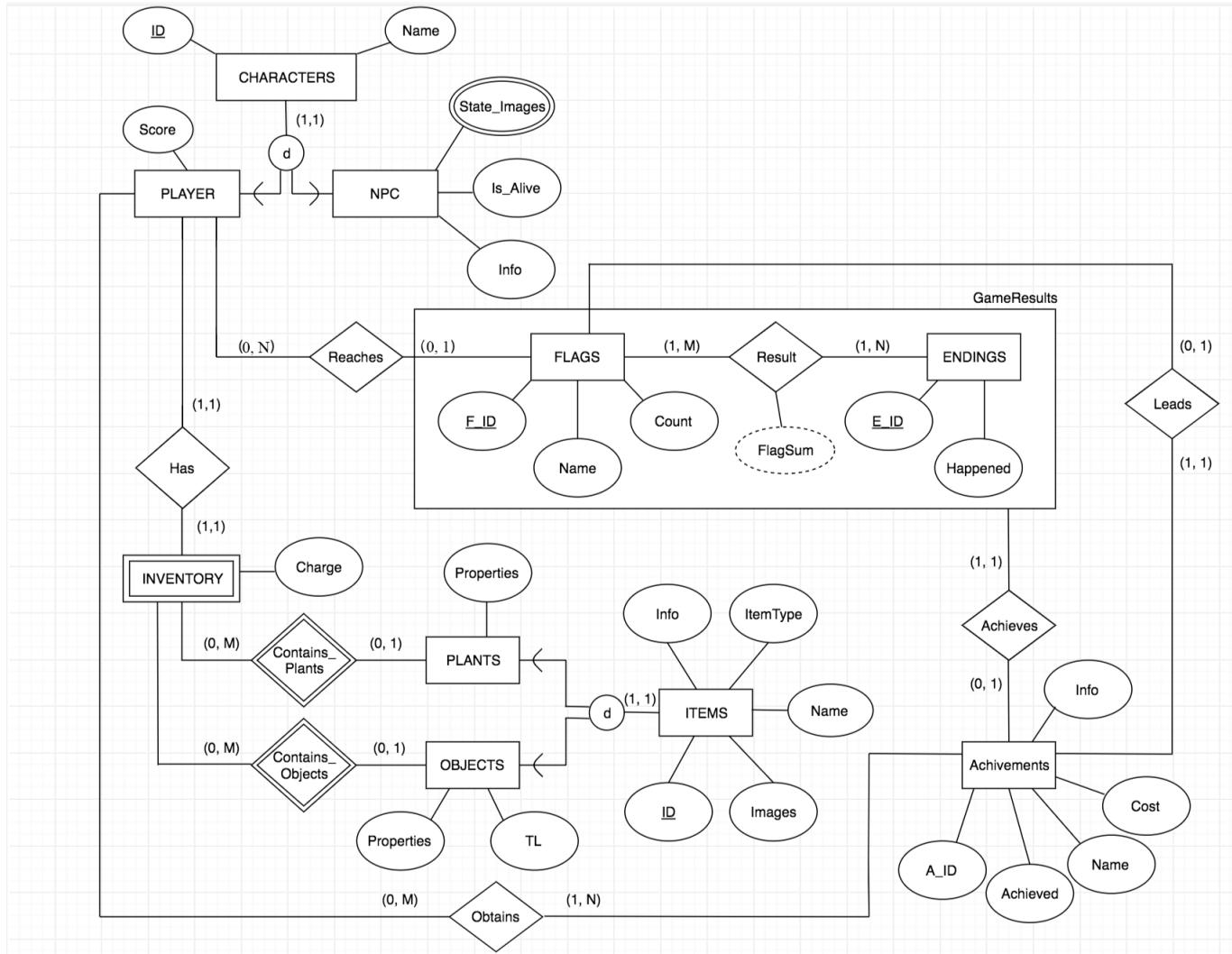
Number of Entity Sets: 13

Number of Relationship Sets: 8

Number of Weak Entity Sets: 8

Number of Superclasses/Subclasses: 2/4

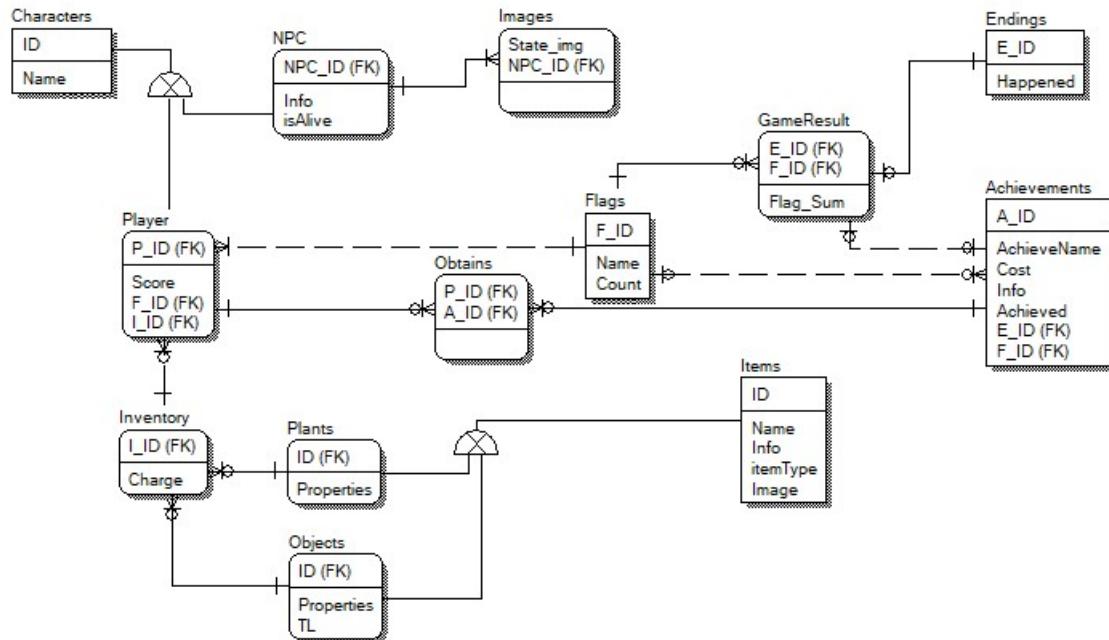
Number of Aggregate Entity Sets: 1



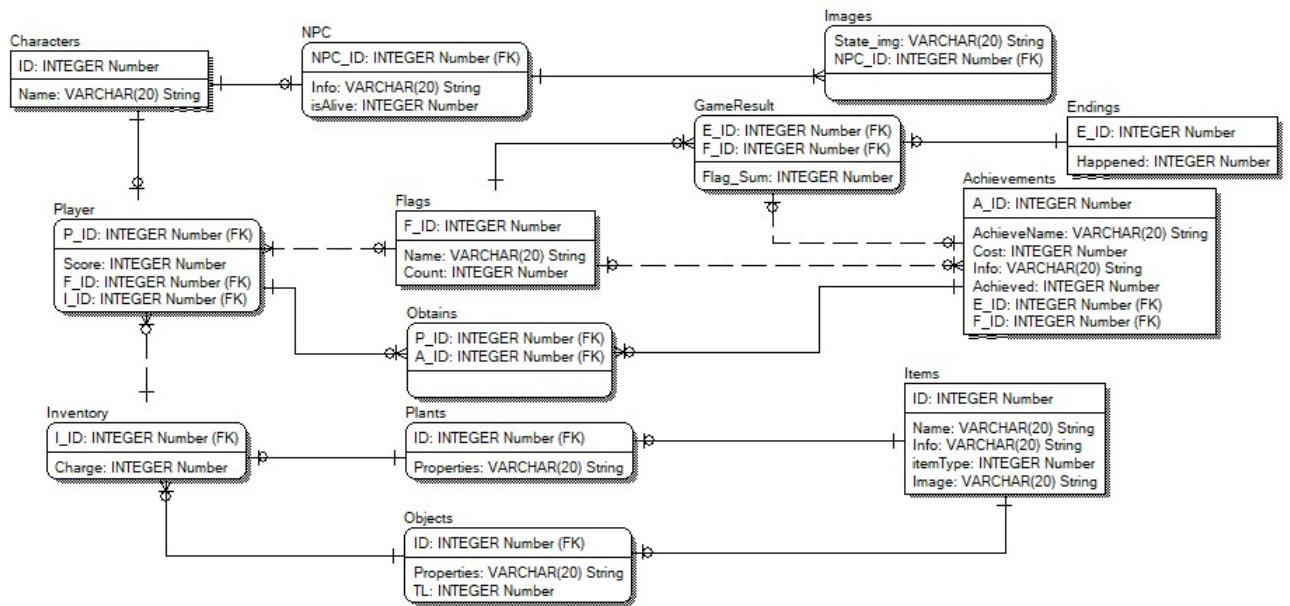
4.2 Logical database design by ER-Win

Next figures are used to define: Sets/Table - []; Relationships - (); Aggregation - ()()

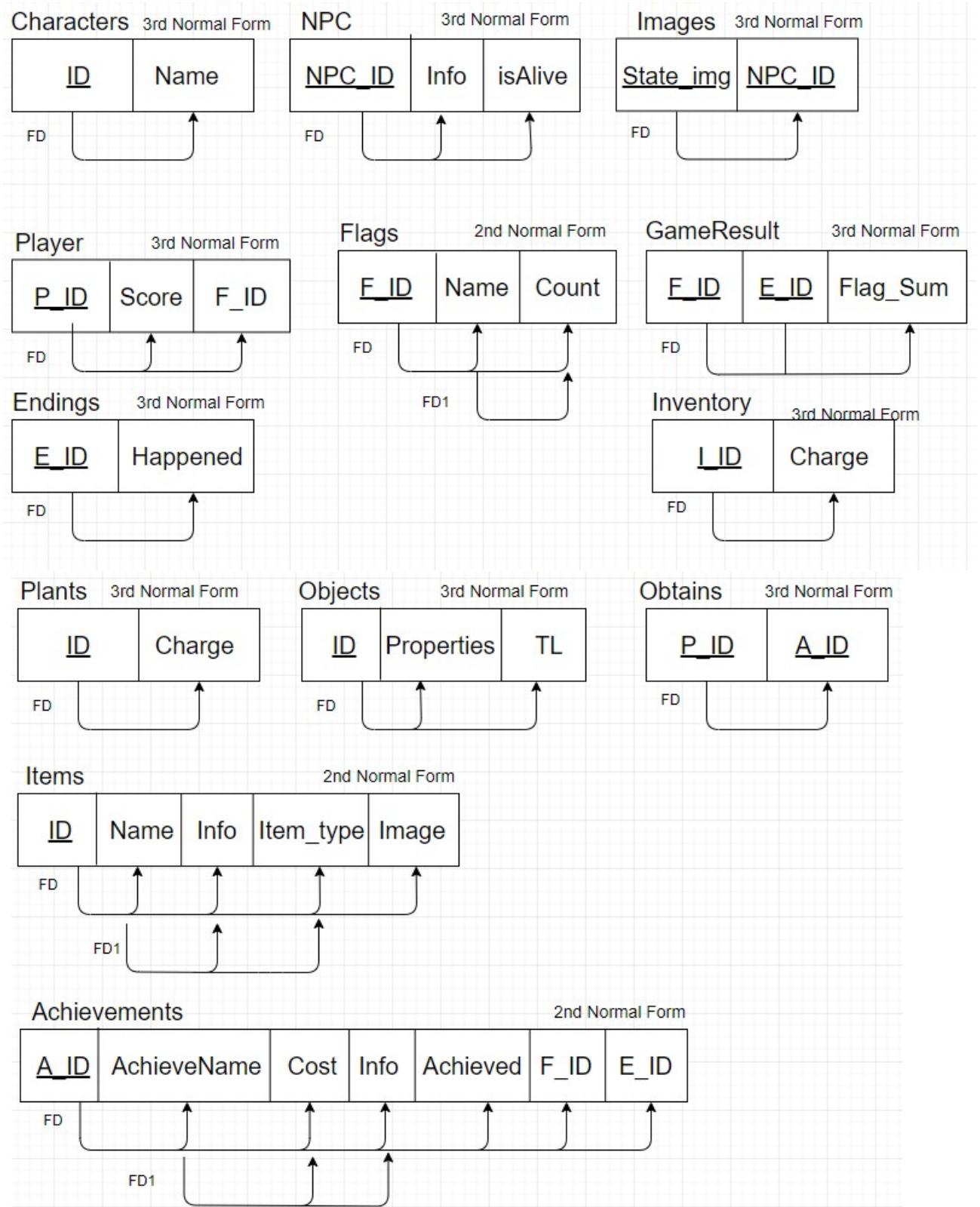
- 1) Multivalue attribute state_image in Entity [NPC] in EER has become independent table in ER-Win
- 2) Relationship (Reaches) has been optimised as foreign key to table [Player] from table [Flags]
- 3) In EER, [Flags] and [Endigs] Entities have relationship (Result) and their aggregation ((GameResult)) has relationship (Achieved) with Entity [Achievements]. In ER-Win, aggregation ((GameResult)) has become independent table and relationship (Result) is optimized into [GameResult] table.
- 4) Relationship (Leads) has been optimised as foreign key to table [Achievements] from table [Flags]
- 5) Relationship (Achievies) has been optimised as foreign key to table [Achievements] from aggregation ((GameResult))
- 6) Relationship (ContainsPlants) which is identifying relationship for [Inventory], has been optimised as foreign key to table [Inventory] from table [Plants]
- 7) Relationship (ContainObjects) which is identifying relationship for [Inventory], has been optimised as foreign key to table [Inventory] from table [Objects]



4.3 Physical database design by ER-Win



5. Dependency design



Achievements

2nd Normal Form

A_ID	AchieveName	Cost	Info	Achieved	F_ID	E_ID
FD						

FD
FD1

Achievements

```

A_ID: INTEGER Number
AchieveName: VARCHAR(20) String
Cost: INTEGER Number
Info: VARCHAR(20) String
Achieved: INTEGER Number
E_ID: INTEGER Number (FK)
F_ID: INTEGER Number (FK)

```

CREATE TABLE Achievements

```

(
    A_ID          INTEGER NOT NULL,
    AchieveName   VARCHAR(20) NULL,
    Cost          INTEGER NULL,
    Info          VARCHAR(20) NULL,
    Achieved      INTEGER NULL,
    E_ID          INTEGER NULL,
    F_ID          INTEGER NULL,
    PRIMARY KEY (A_ID),
    FOREIGN KEY (E_ID, F_ID) REFERENCES GameResult (E_ID, F_ID),
    FOREIGN KEY (F_ID) REFERENCES Flags (F_ID)
);

```

Items

2nd Normal Form

ID	Name	Info	Item_type	Image
FD				

FD
FD1

Items

```

ID: INTEGER Number
Name: VARCHAR(20) String
Info: VARCHAR(20) String
itemType: INTEGER Number
Image: VARCHAR(20) String

```

CREATE TABLE Items

```

(
    ID           INTEGER NOT NULL,
    Name         VARCHAR(20) NULL,
    Info         VARCHAR(20) NULL,
    itemType    INTEGER NULL,
    Image        VARCHAR(20) NULL,
    PRIMARY KEY (ID)
);

```

Flags 2nd Normal Form

<u>F_ID</u>	Name	Count
FD		FD1

Flags

```

F_ID: INTEGER Number
Name: VARCHAR(20) String
Count: INTEGER Number

```

CREATE TABLE Flags

```

(
    F_ID           INTEGER NOT NULL,
    Name          VARCHAR(20) NULL,
    Count         INTEGER NULL,
    PRIMARY KEY (F_ID)
);
```

GameResult 3rd Normal Form

<u>F_ID</u>	<u>E_ID</u>	Flag_Sum
FD		

GameResult

```

E_ID: INTEGER Number (FK)
F_ID: INTEGER Number (FK)
Flag_Sum: INTEGER Number

```

CREATE TABLE GameResult

```

(
    E_ID           INTEGER NOT NULL,
    F_ID           INTEGER NOT NULL,
    Flag_Sum       INTEGER NULL,
    PRIMARY KEY (E_ID,F_ID),
    FOREIGN KEY (E_ID) REFERENCES Endings (E_ID),
    FOREIGN KEY (F_ID) REFERENCES Flags (F_ID)
);
```

Endings 3rd Normal Form

<u>E_ID</u>	Happened
FD	

Endings

```

E_ID: INTEGER Number
Happened: INTEGER Number

```

CREATE TABLE Endings

```

(
    E_ID           INTEGER NOT NULL,
    Happened      INTEGER NULL,
    PRIMARY KEY (E_ID)
);
```

6. Interface description

Main Menu contains five buttons: START, OPTIONS, ENCYCLOPEDIA, ACHIEVEMENTS, EXIT.

START - starts the game itself

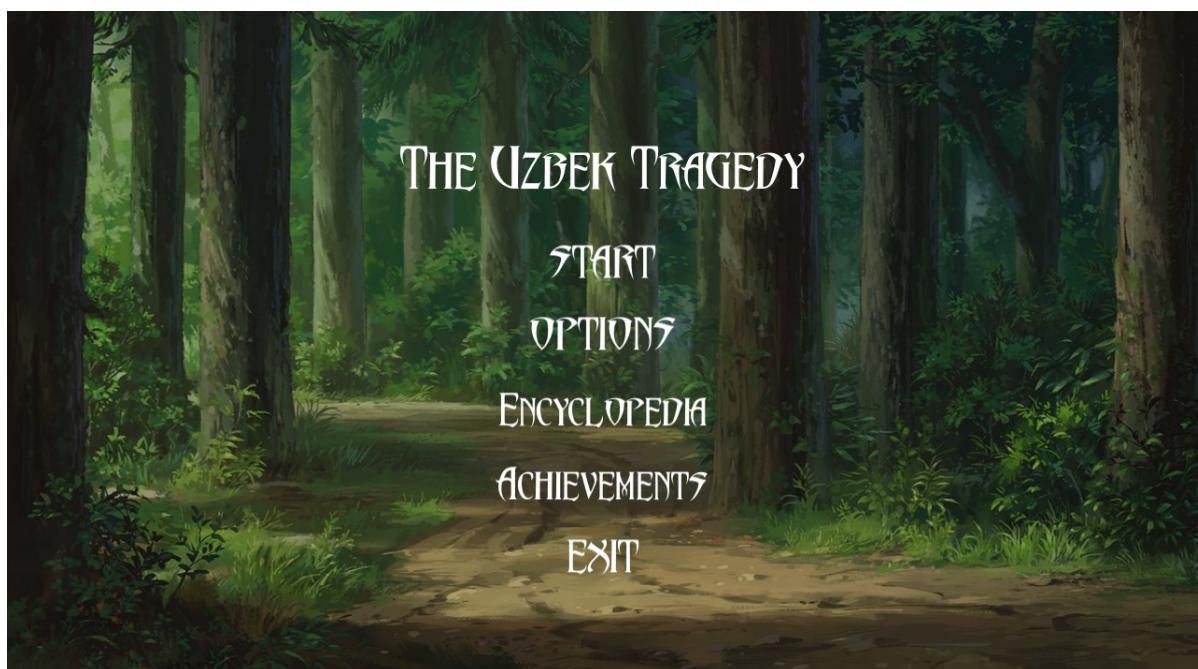
OPTIONS - adjust volume of sound

ENCYCLOPEDIA - contains information about NPC, plants, animals, items, objects.

Selection from NPC, Characters and Items table

ACHIEVEMENTS - shows game results like name, cost (how much "achieved" costs), info (what you have done), achieved (boolean). *Selection* from Achievements table

EXIT - exit/quit game

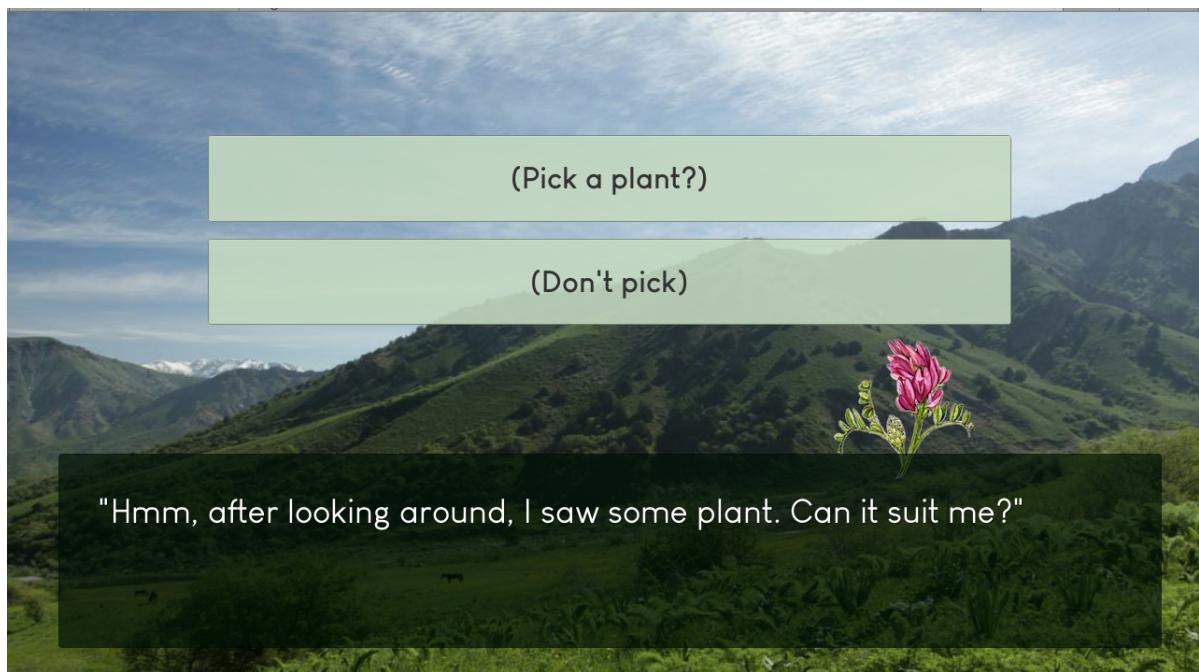


MAIN MENU

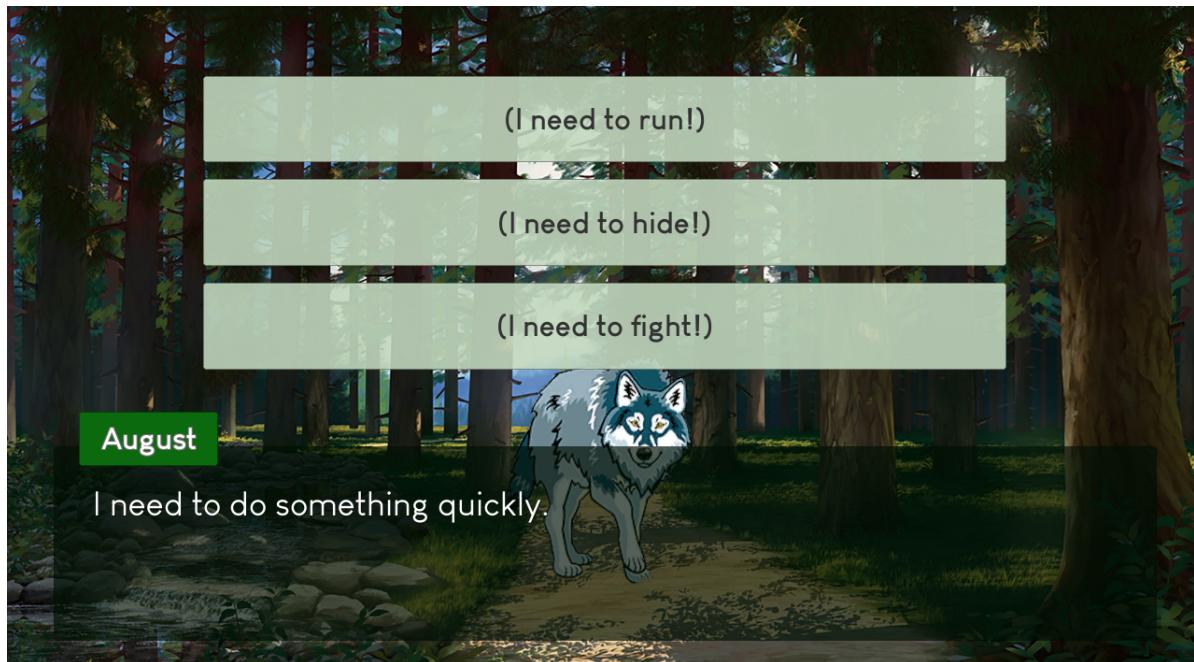
#	Name	Info
---	------	------

- | | | |
|----|------------------|---|
| #1 | Ferula
stinky | In Kyzyl Kum, in places it forms dense thickets up to one and a half meters high. It is hard to believe that the first images grow only basal leaves. Then, in 5-7 weeks, a spherical umbrella of yellow flowers and fruits unfolds. The smell of ferula disgusting. She was only once in 10 years. Another feature is the presence of a thick milky sap that occurs after a stalk is cut at the root. Juice stiffens in the air in the form of gum - resin. It used to be worth its weight in gold. Ibn Sina recommended it as a fortifying, appetizing and diuretic. It is used in scientific medicine in |
|----|------------------|---|

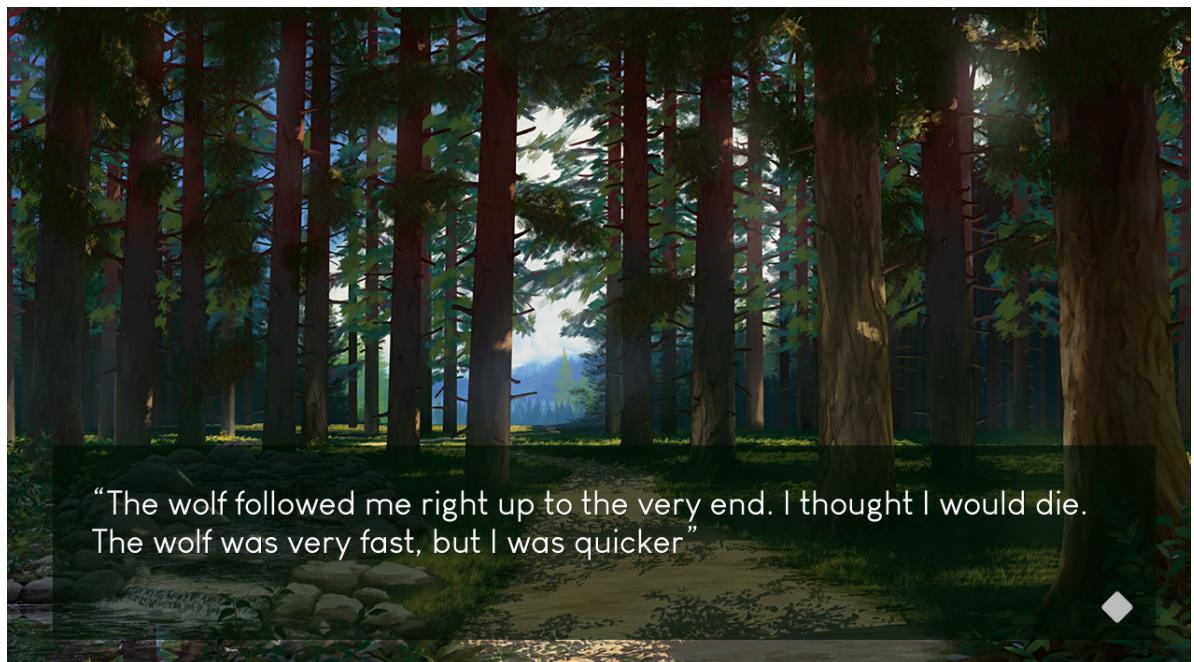
As an example, we consider only one story line. After bus accident, a player gets hurt and finds out a plant that can be helpful to heal his wound. The player is supposed to do a choice that change attribute count in the table Flags. Suppose, the player chooses to "pick a plant". *Update of Flags table.* Every choice is updating flags table and will lead to any of endings



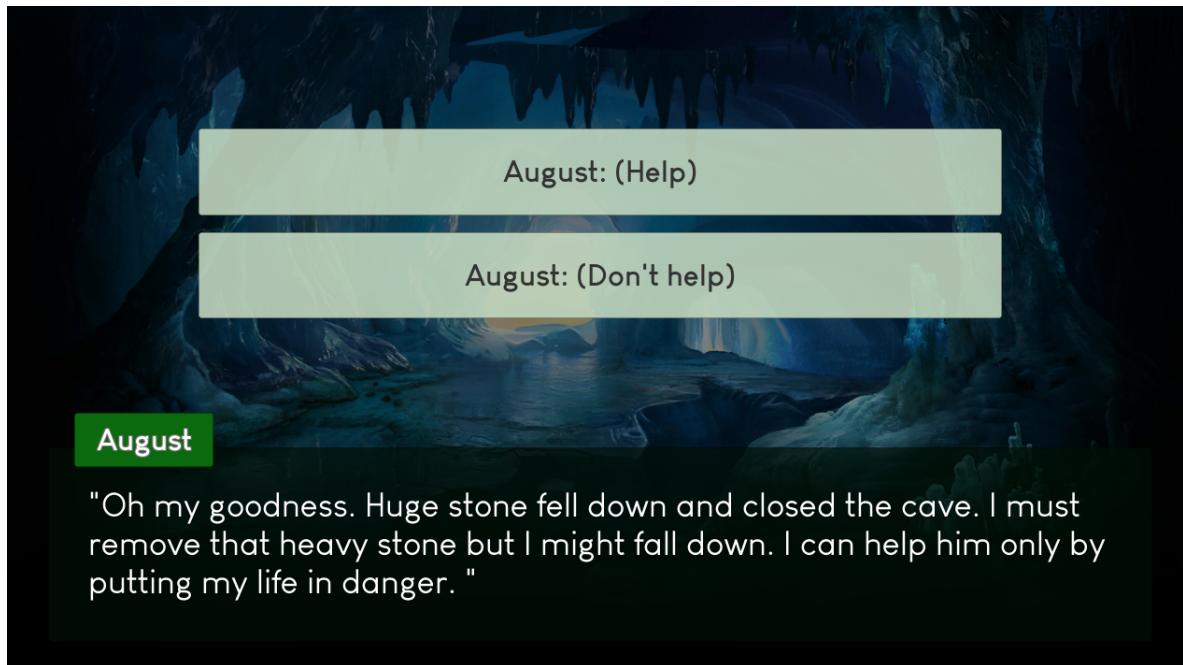
After some time, the player again needs to choose, and let it be "I need to run". If player would choose I need to fight, he would achieve Brave One Achievement.



The player is lucky that can avoid the wolf. But it is possible that the player would die, if he initially didn't choose "pick a plant".



And the player encounters with one of his friends (Zod) whom the player's help is needed. Let the player choose "Help".



August

"Oh my goodness. Huge stone fell down and closed the cave. I must remove that heavy stone but I might fall down. I can help him only by putting my life in danger. "

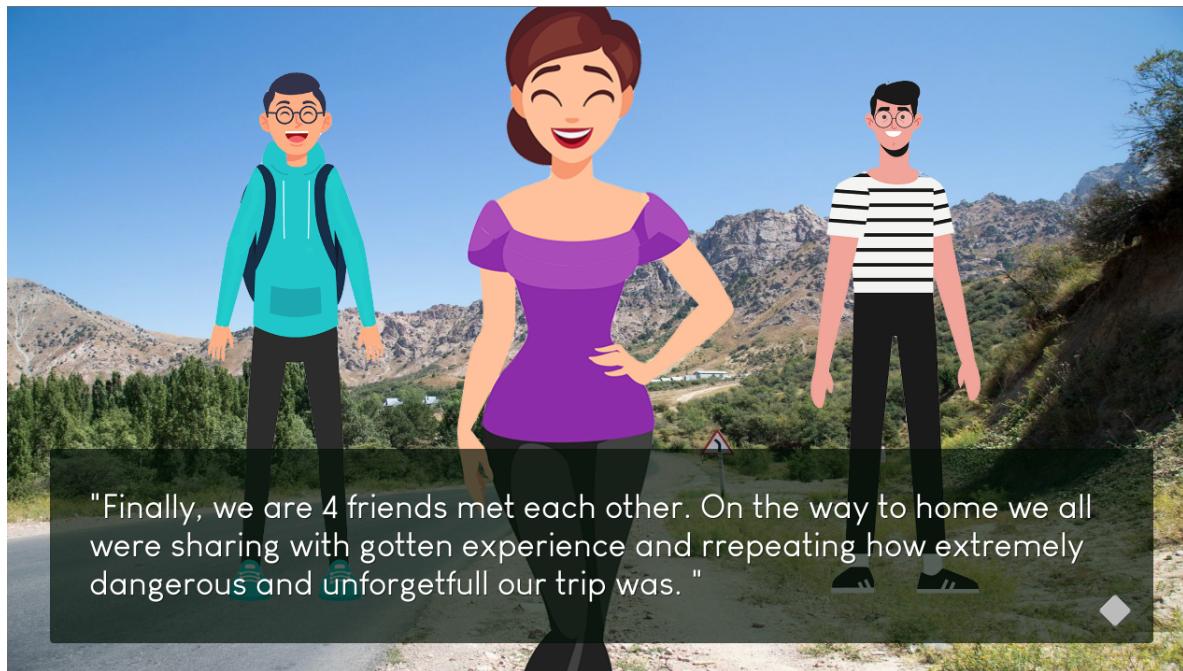
Now, the player needs to descent from the mountain with rope or without.



August

"Mmm..."

The choice "Climb down with rope" leads to successful ending. It is the end of game. After the game end, Player will get Game Result inserting result into GameResult table, with the Sum of the flags count and ending id showing which ending the player got.



Here is Achievements of the player illustrating the name, cost, info and achieved attributes.

MAINMENU		ACHIEVEMENTS	
NAME	COST	INFO	ACHIEVED
ZOD'S	5	YOU HAVE SAVED ZOD	1

7. Testing of database

Plain English

6. Update players score to 1.
2. Set achieved in Achievements table to 1, meaning that it is true, and player got some achievements.
3. Insert into Obtains table players id and achievements id.
4. Make Gameresult to the sum of flags count.

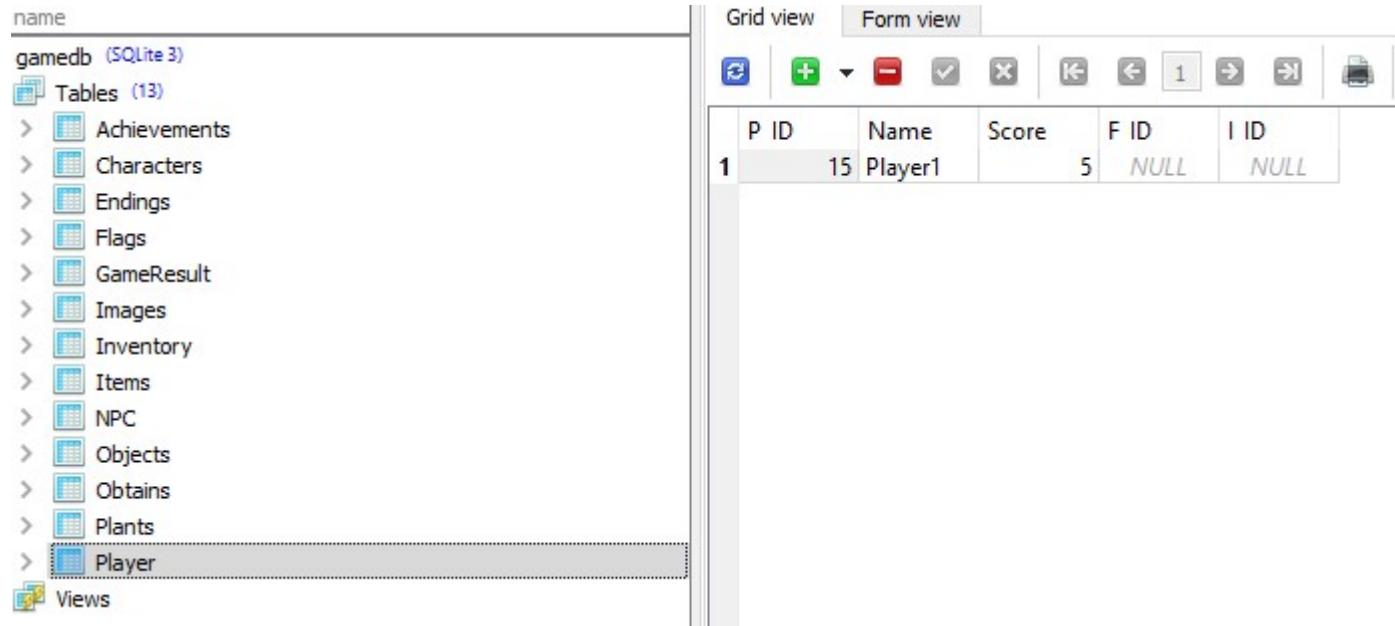
5. When game is starting set all endings to 0.
6. Characters is a superclass of NPC, so Select everything from characters and npc where characters id is equal to npc id.

Standard SQL

1. UPDATE Player SET score = score + 1 WHERE P_ID = (SELECT P_ID FROM PLAYER)
2. UPDATE achievements SET achieved = 1, cost = 1 WHERE A_ID = 1
3. INSERT INTO Obtains values ((SELECT P_ID FROM PLAYER) , 1))
4. INSERT OR REPLACE INTO gameresult VALUES (1, 1, (SELECT SUM(count) as flag_sum from flags))
5. UPDATE Endings SET Happened = REPLACE(0, E_ID, 0)
6. SELECT * FROM characters INNER JOIN NPC ON id = npc_id

DBMS Screenshots

1.



The screenshot shows a database interface for a SQLite database named 'gamedb'. On the left, there is a tree view of tables: Achievements, Characters, Endings, Flags, GameResult, Images, Inventory, Items, NPC, Objects, Obtains, Plants, and Player. The 'Player' table is currently selected. On the right, the 'Grid view' is displayed, showing one row of data:

	P_ID	Name	Score	F_ID	I_ID
1	15	Player1	5	NULL	NULL

2.

Filter by name

Achievements

A_ID	AchieveName	Cost
1	BRAVE ONE	1
2	DEATH WALKER	1
3	ZOD's GUARDIAN ANGEL	0
4	RENE's GUARDIAN ANGEL	4
5	MORPHIUS'es GUARDIAN ANGEL	0

3.

atabases

Filter by name

Structure Data Constraints Indexes

Grid view Form view

P_ID	A_ID
15	1

4.

By name

gamedb (SQLite 3)			
Tables (13)			
>	Achievements		
>	Characters		
>	Endings		
>	Flags		
>	GameResult		
>	Images		
>	Inventory		
>	Items		
>	NPC		
>	Objects		

Grid view Form view

F_ID	E_ID	Flag_Sum	
1	7	1	5
2	1	0	5
3	6	2	2
4	2	0	3
5	4	0	4
6	9	0	3

5.

By name

gamedb (SQLite 3)			
Tables (13)			
>	Achievements		
>	Characters		
>	Endings		
>	Flags		
>	GameResult		
>	Images		
>	Inventory		
>	Items		
>	NPC		
>	Obtains		
>	Plants		

Grid view Form view

E_ID	Happened	
1	0	0
2	1	1
3	2	0
4	3	0
5	4	0
6	5	0
7	6	0
8	7	0
9	8	0

6.

Total rows loaded: 14

ID	Name	NPC ID	Info
1	Brown bear		1 The Brown bear (<i>Ursus arctos</i>) can be found on coats of arms as well as on flags. A
2	Tiger		2 Tigers are the largest living cats. They reach a length up to 280 cm and a weight of
3	Caracal		3 The caracal (<i>Caracal caracal</i>) is most famous for his skill in the catching of birds, s
4	Jungle cat		4 The population of Jungle cats (<i>Felis chaus</i>) is significant in India (and adjacent na
5	Marbled Polecats		5 The small mammal, the Marbled Polecat (<i>Vormela peregusna</i>), belongs to the mon
6	Little ground squirrel		6 The little ground squirrel has a stout, low-slung body, short legs and a well-furred
7	Turkestan red pika		7 The Turkestan red pika (<i>Ochotona rutila</i>) is a species of mammal in the family Och
8	Wolf		8 The wolf (<i>Canis lupus</i>), also known as the gray/grey wolf, timber wolf, or tundra w
9	Corsac fox		9 The Corsac Fox (<i>Vulpes corsac</i>) lives in the steppes and semi-desert of Central Asia
10	Striped hyena		10 They have big heads, big eyes, and ears that are also big, yet this is not a giant ani
11	Giant Noctule		11 The Giant Noctule (<i>Nyctalus lasiopterus</i>) is a rare, flying mammal also known as the
12	Morphius		12 Morphius is third year student of INPA University. He friends with August since the
13	Rene		13 Rene is known as Miss Active University. From the very beginning of her studies, sh
14	Zod		14 Zod is one of the the cleverest guys of our University, he always gets first place in

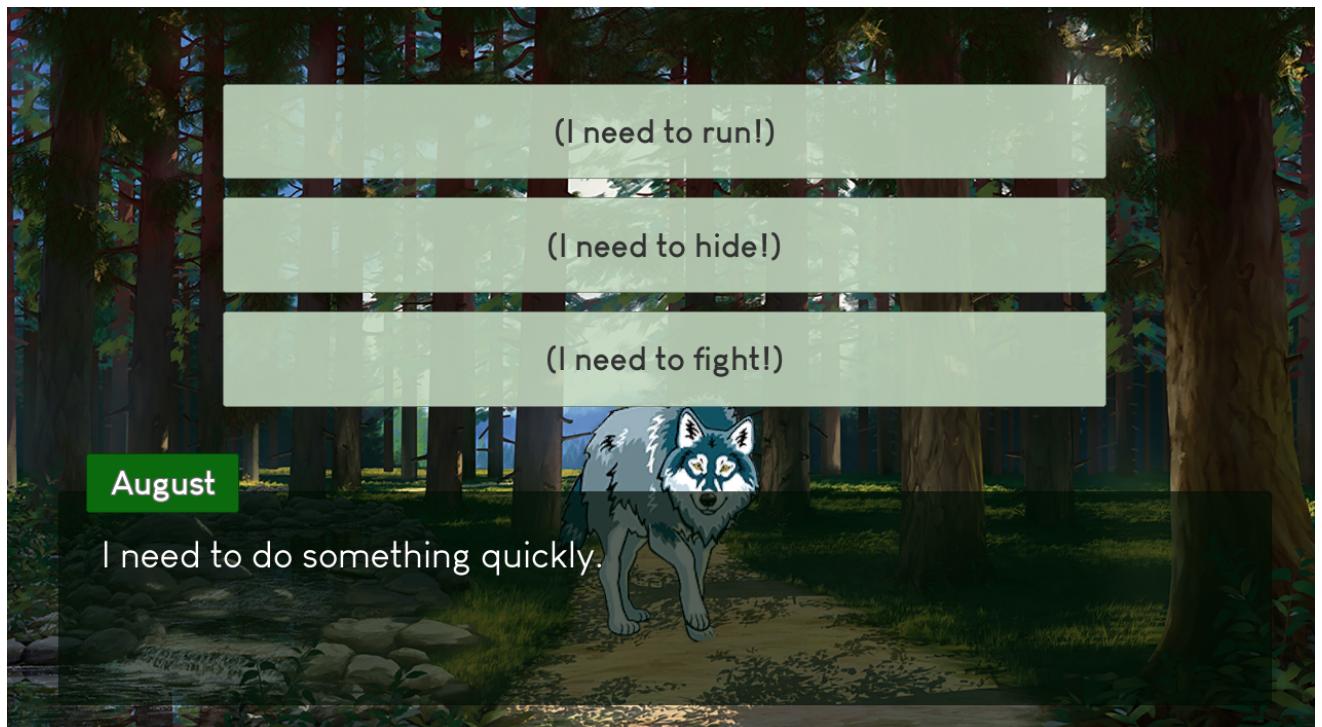
Interface Screenshots

1-3.

```
[nCommand("UpdateAchievements")]
references
lic void UpdateAchievement(params string[] parameters)

var par = CleanParams( parameters );
var id = par[0];
var cost = par[1];

using (dbconn = new SqliteConnection(conn))
{
    dbconn.Open(); //Open connection to the database.
    dbcmd = dbconn.CreateCommand();
    sqlQuery = string.Format("UPDATE achievements SET achieved = 1, cost = \"{1}\" WHERE A_ID = \"{0}\",id, cost");// table name
    dbcmd.CommandText = sqlQuery;
    dbcmd.ExecuteScalar();
    sqlQuery = string.Format("UPDATE Player SET score = score + \"{0}\" WHERE P_ID = (SELECT P_ID FROM PLAYER)",cost );
    dbcmd.CommandText = sqlQuery;
    dbcmd.ExecuteScalar();
    sqlQuery = string.Format("INSERT INTO Obtains values ( (SELECT P_ID FROM PLAYER) ,\"{0}\",id );// table name
    dbcmd.CommandText = sqlQuery;
    dbcmd.ExecuteScalar();
    dbconn.Close();
}
```

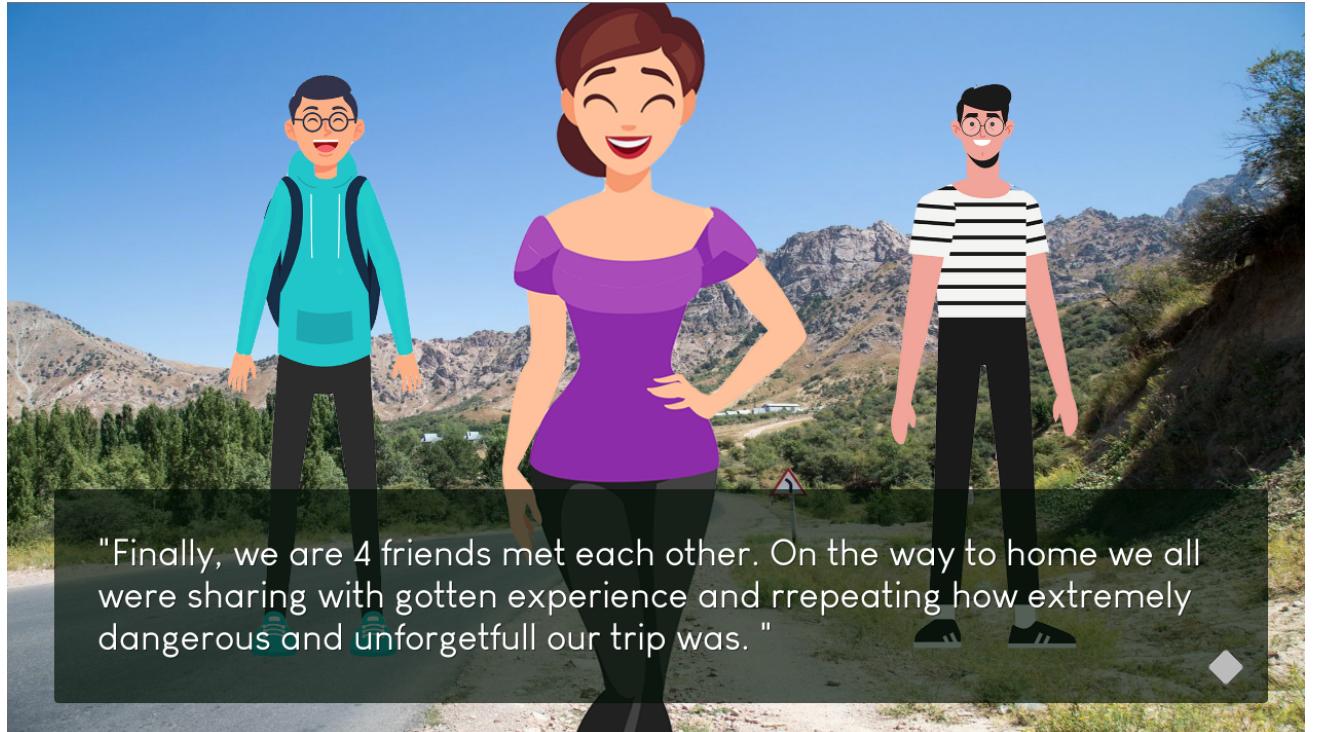


4-5.

```
onnection(conn)

onnection to the database.
command();
t("insert or replace into gameresult values (\\"{0}\", \\"{1}\", (SELECT SUM(count) as flag_sum from flags))", f_id, e_id);//
Query;

t("update Endings set happened = 1 where e_id = \\"{0}\\"", e_id); // table name
Query;
```



6.

```
ng(IDbConnection dbConnection = new SqliteConnection(conn))

    dbConnection.Open();
    using(IDbCommand dbCmd = dbConnection.CreateCommand())
    {
        string sqlQuery = "SELECT * FROM characters INNER JOIN NPC ON id = npc_id";
        dbCmd.CommandText = sqlQuery;
        using(IDataReader reader = dbCmd.ExecuteReader())
        {
            while(reader.Read())
            {

```

MAINMENU		ACHIEVEMENTS	
NAME	COST	INFO	ACHIEVED
BRAVE ONE	5	YOU DECIDED TO FIGHT WITH WOLF	1

In this SQL queries 1 means number taken from the system.

References

- [1] Unity platform,
<https://unity.com/>
- [2] Fate Stay night,
<http://www.typemoon.com/products/fate/index.html>
- [3] Katawa shoujo,
<https://www.katawa-shoujo.com/>
- [4] Everlasting Summer
<http://www.typemoon.com/products/fate/index.html>