Федеральное казенное профессиональное образовательное учреждение «Новокузнецкий государственный гуманитарно-технический колледж-интернат»

Министерства Труда и социальной защиты Российской Федерации

Специальность: 09.02.04 «Информационные системы (по отраслям)»

КУРСОВОЙ ПРОЕКТ

по дисциплине: Средства диагностики и управления баз данных

на тему: Разработка веб-приложения «Арканоид» с использованием СУБД MS SQL.

РАСЧЁТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Студент Михайлов В. В.

дата личная подпись расшифровка подписи

Руководитель проекта Грисман С.С.

дата личная подпись расшифровка подписи

Нормоконтроль Михайлов Н.Н,

дата личная подпись расшифровка подписи

Оценка

Новокузнецк, 2019 г.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc26781202)

[1. ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА РАЗРАБОТКУ ПРОГРАММНОГО ПРОДУКТА 5](#_Toc26781203)

[2. ОБЩАЯ ЧАСТЬ 7](#_Toc26781204)

[2.1 Выбор и обоснование средств реализации БД 7](#_Toc26781205)

[2.2. Выбор и обоснование средств разработки программного продукта 9](#_Toc26781206)

[2.3 Анализ прототипов, существующих ИС 9](#_Toc26781207)

[3 СПЕЦИАЛЬНАЯ ЧАСТЬ 11](#_Toc26781208)

[3.1 Разработка алгоритма и проектирование структуры базы данных 11](#_Toc26781209)

[3.2 Разработка программного продукта 13](#_Toc26781210)

[3.3 Руководство пользователя 13](#_Toc26781211)

[ЗАКЛЮЧЕНИЕ 30](#_Toc26781212)

[СПИСОК ЛИТЕРАТУРЫ 31](#_Toc26781213)

# **ВВЕДЕНИЕ**

Игры – это неотъемлемая часть нашего существования. В наш, насыщенный информацией век игры двигают вперед развитие компьютерной сферы.

Основой любой игры является интерес к происходящему, личное участие и самоутверждение по завершении игры.

Для хранения информации о пользователе применяется база данных. База данных – это информационная структура, хранимая и предоставляющая информацию к игре.

Целью данного курсового проекта, является разработка игры «Арканоид» с использованием СУБД MSSQL-server.

Актуальность темы обуславливается необходимостью развития координации и скорости реакции.

Объектом в данной области является игра «Арканоид»

Предметом является возможность оптимизации процессов, связанных с обработкой информации и её обменом с пользователями, регистрация незарегистрированных, и авторизация в системе уже зарегистрированных клиентов.

Для достижения поставленной цели необходимо выполнить следующие задачи:

* Разработка технического задания;
* Выбор и обоснование средств реализации БД
* Проанализировать прототипы;
* Произвести анализ и проектирование структуры базы данных;
* Выбрать и обосновать программные средства реализации;
* Разработать программный продукт;
* Провести тестирование и отладить программный продукт.
* Разработать руководство пользователя

Разрабатываемое приложение сможет применяться как средство для развлечения пользователей, развития внимания и реакции.

Создание данного приложения поможет освоить технологии разработки на языке программирования C#, базы данных в MS SQL, проработать связи программы с базой данных.

# **ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА РАЗРАБОТКУ ПРОГРАММНОГО ПРОДУКТА**

**Разработка программного продукта "Игра АРКАНОИД" с применением СУБД**

Прототипом данной системы можно считать интернет [ресурс](http://www.bettergames.ru/game.php?id=3773) (http://www.bettergames.ru/game.php?id=3773), имеющий функциональную часть работы данного приложения.

**Требования к функциональным характеристикам следующие:**

1. Необходимо предусмотреть два режима работы – режим администратора и режим пользователя.
2. В режиме пользователя должна быть возможность зарегистрироваться, заполнить свою анкету (дата рождения, фото). Реализовывать возможность смены пароля и редактирования данных.
3. В режиме пользователя можно начать игру, а также посмотреть свой уровень игры (счет и пр.).
4. В режиме администратора должна быть возможность просматривать и редактировать пользователей, редактировать счет игрока.
5. Игрок характеризуется именем, счетом, фотографией.

В режиме администратора должна быть реализована возможность заполнения игровых данных об игроке.

1. Также у администратора и пользователя имеется возможность формировать рейтинг игроков (по счету, набранным в игре или дате игры и по ткущему пользователю)

**Требования к информационному обеспечению следующие:**

К приложению должна быть предоставлена документация, содержащая техническое описание продукта и руководство пользователя.

**Требования к лингвистическому обеспечению следующие:**

Приложение должно иметь интерфейс и документацию на русском языке.

**Требования к программному обеспечению следующие:**

Разработка программного продукта предусматривается под OS Windows не ниже версии 7.

**Требования к техническому обеспечению следующие:**

Для работы приложения требуется MS SQL Server, Visual Studio С#. Также, необходимо не менее 512Мб оперативной памяти и 100Мб свободного места на жестком диске.

# **ОБЩАЯ ЧАСТЬ**

## **2.1 Выбор и обоснование средств реализации БД**

Основным требованием работы является наличие интерфейса приложения, разрабатываемого в СУБД MS SQL.

**Microsoft SQL Server** — система управления реляционными базами данных (СУБД), разработанная корпорацией Microsoft. Основной используемый язык запросов — Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями. Используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия; конкурирует с другими СУБД в этом сегменте рынка.

**База данных MS SQL** – это система, предназначенная для хранения и обработки информации. Комплекс таблиц, взаимосвязанных между собой, для доступа к которым применяется система управления базами данных (СУБД) MS SQL. MS SQL – это специальная программа с открытым кодом, которая используется на сервере SQL. Данная программа не способна обрабатывать большое количество информации, однако она идеальна для небольших и крупных веб-ресурсов.

В настоящее время, СУБД MS SQL является одной из самых известных, надежных и быстрых из всего семейства существующих СУБД.

Одной из уникальных особенностей версии 2016 года является temporal data support (временная поддержка данных), которая позволяет отслеживать изменения данных с течением времени. Последняя версия Microsoft SQL-сервер поддерживает dynamic data masking (динамическую маскировку данных), которая гарантирует, что только авторизованные пользователи будут видеть конфиденциальные данные.

Достоинства:

1. очень простой графический интерфейс, который позволяет не только создавать собственную базу данных, но и разрабатывать приложения, используя встроенные средства,
2. хранит все данные в одном файле, хотя и распределяет их по разным таблицам, как и положено реляционной СУБД. К этим данным относится не только информация в таблицах, но и другие объекты базы данных.
3. предлагает большое количество Мастеров, которые выполняют основную работу за пользователя при работе с данными и разработке приложений, помогают избежать рутинных действий и облегчают работу неискушенному в программировании пользователю.
4. распространенность, которая обусловлена тем, что Access является продуктом компании Microsoft,
5. постоянно обновляется производителем, поддерживает множество языков,
6. полностью совместим с операционной системой Windows,
7. ориентированность на пользователя с разной профессиональной подготовкой, что выражается в наличии большого количества Мастеров, развитую систему справки и понятный интерфейс.
8. Широкие возможности по импорту/экспорту данных в различные форматы, от таблиц Excel и текстовых файлов, до практически любой серверной СУБД через механизм ODBC,
9. Наличие развитых встроенных средств разработки приложений. Большинство приложений, распространяемых среди пользователей, содержит тот или иной объем кода VBA (Visual Basic for Applications),
10. Наличие встроенного языка макрокоманд.

СУБД MS SQL нельзя причислить в полной мере к проектам Open Source (Открытый исходный код), так как при определенных условиях покупка лицензии все же требуется. Однако условия лицензирования не очень строгие.

Снижение экономических барьеров позволяет получить доступ к базам данных большему количеству людей и организаций, чем, когда бы то ни было.

Мощные реляционные СУБД могут теперь использовать как организации, так и индивидуальные пользователи для решения самых различных задач. И MS SQL – одна из таких систем.

## **2.2. Выбор и обоснование средств разработки программного продукта**

При написании приложения был выбран язык программирования C# (Си шарп).

Язык C#, разработанный компанией Майкрософт, один из самых популярных современных языков программирования. Он востребован на рынке разработки в различных странах, C# применяют при работе с программами для ПК, создании сложных веб-сервисов или мобильных приложений.

Приложение имеет графический интерфейс для построения приложения и программную оболочку для работы непосредственно с кодом. Работать в программе довольно удобно и просто.

Важной и отличительной особенностью C# является его простота. К примеру, всегда ли вы помните, когда пишите на С++, где нужно использовать "->", где "::", а где "."? Даже если нет, то компилятор всегда поправляет вас в случае ошибки. Это говорит лишь о том, что в действительности можно обойтись только одним оператором, а компилятор сам будет распознавать его значение. Так в C#, оператор"->" используется очень ограничено, оператор "::" вообще не существует. Практически всегда вы используете только оператор "." и вам больше не нужно стоять перед выбором.

## **2.3 Анализ прототипов, существующих ИС**

При реализации программного продукта, разработчик анализирует и систематизирует данные, просматривая аналогичные приложения. На данном этапе происходит весь сбор информации, выявление достоинств и недостатков, которые имеются у прототипов. После проведённой работы, он продумывает план создания приложения, куда будут включены плюсы, а также исключены ошибки прототипов.

Схема анализа прототипов включает в себя:

* дизайн приложения и общее впечатление от него;
* доступность информации для пользователей;
* структура программного продукта и удобство использования;
* навигация в приложении;
* объем и актуальность представленной информации в данном программном продукте;
* удобство использования панели администратора данного приложения.

При проектировании приложения «АРКАНОИД» прототипами стали Web-приложения «3DNOID». Перейти на сайт можно по адресу http://www.bettergames.ru/arcanoid.php. Представлено на рисунке ниже.

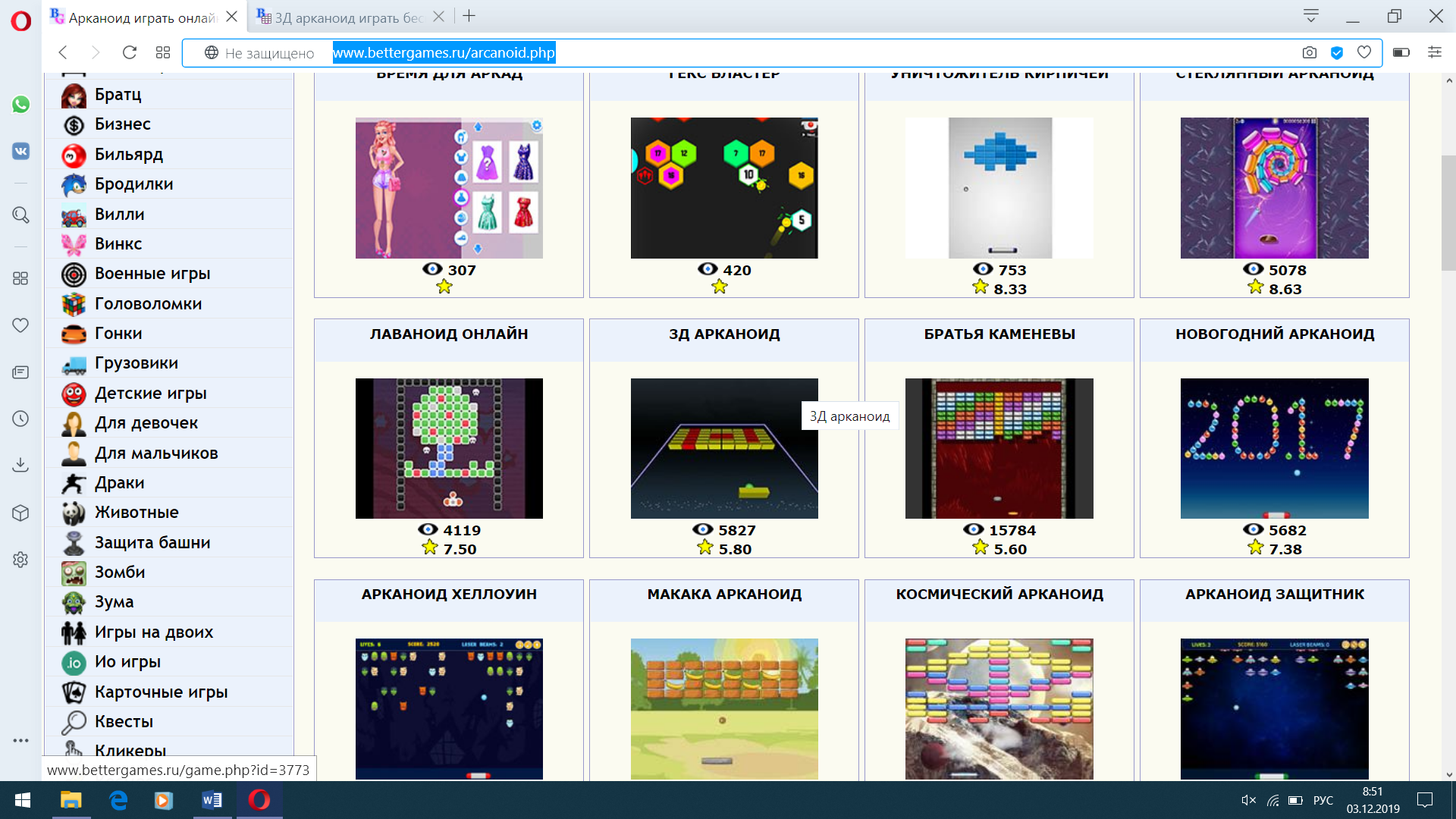


Рисунок 1 - «Сайт игр. Раздел Арканоид»

На сайте присутствует множество игр подобного жанра, остальных направлений.

Игра имеет псевдо 3D дизайн, в ней работает подсчёт баллов. Но не реализована авторизация и регистрация пользователей, результаты не сохраняются. При этом весь необходимый для создания программного продукта механизм работает отлично.

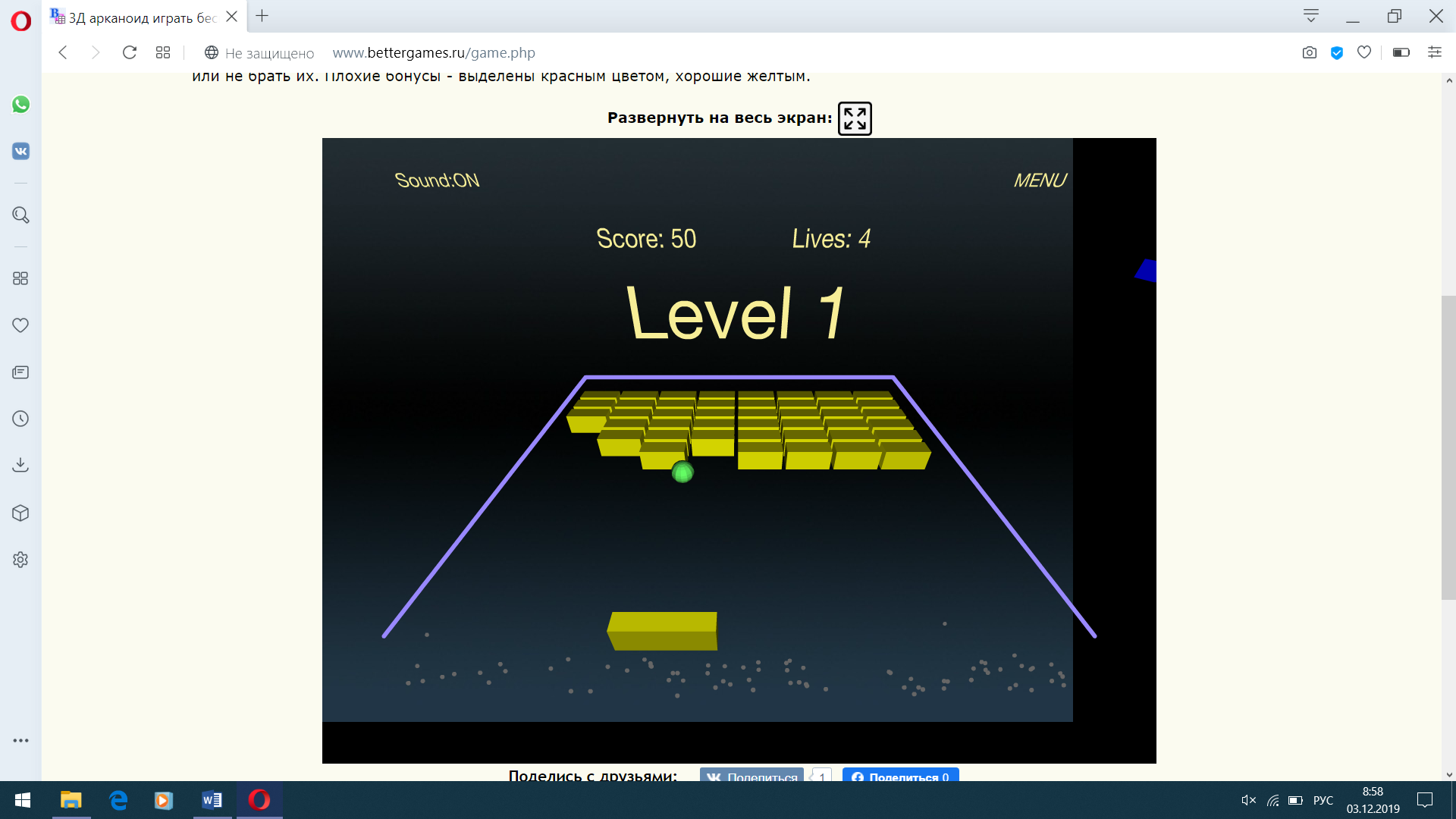


Рисунок 2 - «Игра 3DNOID»

# **3 СПЕЦИАЛЬНАЯ ЧАСТЬ**

## **3.1 Разработка алгоритма и проектирование структуры базы данных**

База данных «ARCANOID» была создана на основании задания, представленного в приложения А.

Модель базы данных «ARCANOID» создана в дизайнере Microsoft SQL Server Management Studio 18.

При создании таблиц базы данных использовался режим конструктора MS SQL, где были установлены свойства полей всех таблиц данной базы данных.

База данных «ARCANOID» состоит из двух таблиц:

Для создания таблиц использовался режим конструктора, позволяющий установить свойства полей таблицы.

Модель базы данных «ARCANOID» состоит из двух таблиц, каждая из которых имеет свою структуру, необходимую для установления связи между всеми таблицами:

* «USERS» (Пользователи);
* «PLAYERS» (Игроки);

Таблица «USERS» предназначена для входа в систему пользователей, при вводе логина и пароля. Ключевым полем является id\_user (Код пользователя), тип данных int. Также имеются поля login\_user (Логин пользователя), тип данных nvarchar(50) и password\_user (Пароль пользователя), тип данных nvarchar(50).

Таблица «PLAYERS» имеет связь с таблицей «USERS». Ключевым полем является поле id (Код) тип данных int. Также имеются поля id\_players (Код игрока), по которому проходит соединение с таблицей «USERS».

Целостность данных *–* это значит, что данные будут надёжно хранится в базе данных. СУБД автоматически поддерживает целостность данных с помощью установленных связей между таблицами. Таблицы базы данных связаны связями «один ко многим»; для каждой из связей установлена поддержка целостности данных, каскадные обновление и удаление. Схема связей таблиц представлена в соответствии с рисунком 3.

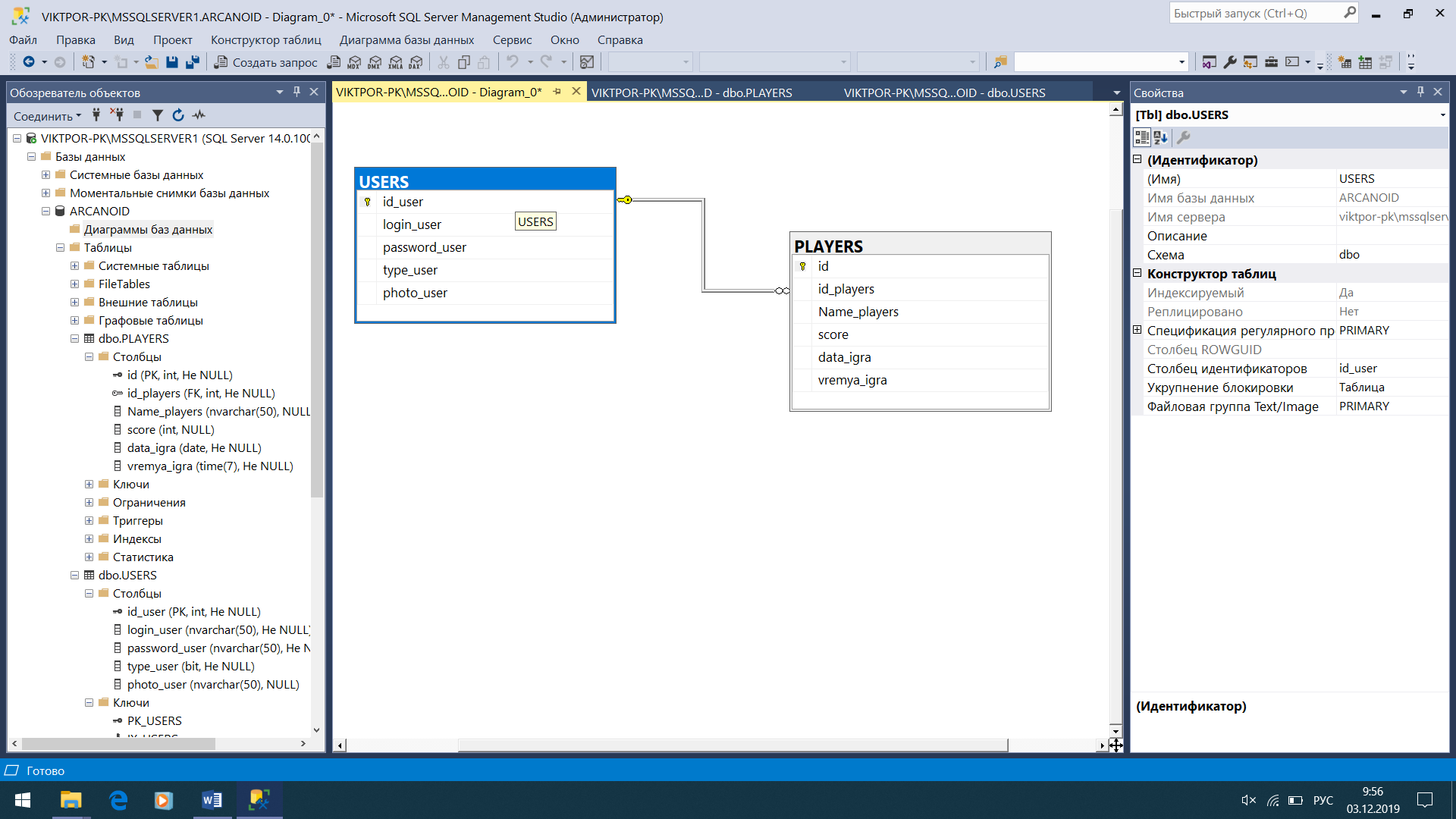


Рисунок 3 - «Схема связей таблиц»

## **3.2 Разработка программного продукта**

При разработке данного программного продукта было реализовано несколько подсистем. Первая из них – система авторизации. В ней реализован механизм, где работа происходит с зарегистрированными пользователями. Алгоритм работы выглядит так, что при загрузке приложения, происходит проверка на регистрацию пользователя в базе программы. Если она имеется, то авторизация не предлагается, а в противном случае, необходимо её пройти. Данный механизм реализован посредством простой проверкой существования записи в таблице. Проверка идет SQL запросом в присоединенную к программе базу данных. См. Листинг 1.

Листинг 1. Поиск юзера

//строка н поиск юзера

string s = "select \* from users where login\_user = '" + login\_user + "' and password\_user = '" + password\_user + "'";

Стоит отметить, что в данном действии, как и во всей системе происходит работа с базой данных. После ввода логина и пароля происходит отправка запроса на выборку имени пользователя (login\_user) и пароля (password\_user) после чего возвращает результат выполнения. В программе предусмотрен вывод предупреждения в случае ошибочных данных. См. Листинг 2.

Листинг 2. Вывод ошибки пользователя

else

{

MessageBox.Show("Пользователь не найден");

}

Вторая из подсистем – регистрация. При заполнении необходимых полей происходит добавление нового пользователя в базу данных. Листинг 3.

Листинг 3. Запрос на регистрацию

SqlConnection connection = new SqlConnection(Program.st\_connect);

try

{

connection.Open();

SqlCommand command = new SqlCommand($"INSERT INTO[dbo].[USERS]" +

$"([login\_user],[password\_user],[type\_user],[photo\_user])" +

$"VALUES" +

$"('{LoginBox.Text}','{PasswordBox.Text}',0,'{pathImage}')");

command.Connection = connection;

int rows = command.ExecuteNonQuery();

if(rows > 0)

{

MessageBox.Show($"Добавлено {rows} пользователей");

}

После выполнения команды в базу данных попадает запись о новом пользователе. После чего выводится сообщение о добавлении нового пользователя.

Далее открывается форма просмотра и изменения статистики. Переход в нее осуществляется через проверку типа пользователя. См. Листинг 4. Если авторизуемый тип пользователя «Администратор», то открывается форма администратора, если «Пользователь», соответственно открывается форма для пользователя.

Листинг 4. Проверка типа пользователя

if (Program.type\_user == false)

{

USERS\_FORM f = new USERS\_FORM();

f.Show();

}

else

{

Form3 f = new Form3();

f.Show();

}

Из этих форм можно попасть в формы «Изменение пользователя», «Статистика пользователя» или запустить саму игру.

Форма изменения пользователя реализована так же, как и регистрация, так же там добавлена возможность смены типа пользователя. См. Листинг 5.

Листинг 5. Тип пользователя

Запрос на обновление данных в таблице пользователей

int k = dataGridView1.CurrentRow.Index;

Program.id\_user = Convert.ToInt32(dataGridView1[0, k].Value);//запоминаем id выбранного юзера

SqlConnection connect = new SqlConnection(Program.st\_connect);

connect.Open();

string s = "UPDATE USERS set " +

"login\_user='" + LoginBox.Text.Trim() + "', " +

"password\_user='" + PasswordBox.Text.Trim() + "'," +

"type\_user='" + comboBox1.SelectedIndex + "'," +

"photo\_user='" + label3.Text + "' " +

"WHERE id\_user = " + Program.id\_user.ToString();

SqlCommand comm = new SqlCommand(s, connect);

comm.ExecuteScalar();

connect.Close();

Запрос на обновление видимых данных в таблице после обновления данных в базе

SqlConnection conn1 = new SqlConnection(Program.st\_connect);

conn1.Open();

string s1 = "SELECT \* FROM USERS";

//Делаем запрос к БД через адаптер, что бы потом поместить даные в DATASET

SqlDataAdapter adap = new SqlDataAdapter(s1, conn1);

//создаем пустой датасет

DataSet ds1 = new DataSet();

//соединяем датасет с нашим набором данных

adap.Fill(ds1);

//соединяем датасет с визуальным компонентом

dataGridView1.DataSource = ds1.Tables[0];

if (ds1.Tables[0].Rows.Count >= 10)

{

button2.Enabled = true;

}

conn1.Close();

dataGridView1.Columns[0].Visible = false;

LoginBox.Text = "";

PasswordBox.Text = "";

comboBox1.Text = "";

Вывод сообщения об изменении

MessageBox.Show("Изменено");

Далее идет форма просмотра и изменения статистики. Существует форма изменения статистики, открываемая только для администратора, и упрощенная форма администратора, открываемая для пользователя, с меньшим функционалом.

В данной форме показывается таблица всех игроков, которую можно перелистывать. В ней так же можно удалять игроков (Листинг 6), менять его игровые данные (Листинг 7, 8, 9), формировать отчет в Excell (Листинг 10).

Листинг 6. Удаление

int k = dataGridView1.CurrentRow.Index;

Program.id\_user = Convert.ToInt32(dataGridView1[0, k].Value);

SqlConnection conn = new SqlConnection(Program.st\_connect);

conn.Open();

string s = "DELETE FROM PLAYERS WHERE id=" + Program.id\_user.ToString();

SqlCommand comm = new SqlCommand(s, conn);

comm.ExecuteScalar();

conn.Close();

MessageBox.Show("Чел удален");

При нажатии кнопки изменения, происходит изменение размеров формы, открываются дополнительные поля

Листинг 7. Изменение размеров формы

this.Width = 900;

this.Height = 465;

В текстбоксы переносятся данные выделенной в таблице строки, которые можно изменять.

Листинг 8. Данные в текстбоксах

textBox2.SelectionStart = 0;

textBox2.SelectionLength = textBox2.Text.Length;

textBox2.Focus();

textBox1.Text = dataGridView1.CurrentRow.Cells[1].Value.ToString();

textBox2.Text = dataGridView1.CurrentRow.Cells[2].Value.ToString();

Листинг 9. Изменение данных

int k = dataGridView1.CurrentRow.Index;

Program.Id = Convert.ToInt32(dataGridView1[0, k].Value);//запоминаем id выбранного юзера

SqlConnection conn = new SqlConnection(Program.st\_connect);

conn.Open();

string s = "UPDATE PLAYERS " +

" SET " + "Name\_players= " + "'" + textBox1.Text.Trim() + "'," +

"score = " + "'" + textBox2.Text.Trim() + "'," +

"data\_igra = " + "'" + dateTimePicker1.Value.ToShortDateString() + "'," +

"vremya\_igra = " + "'" + dateTimePicker2.Value.ToShortTimeString() + "' WHERE Id=" + Program.Id.ToString();

SqlCommand comm = new SqlCommand(s, conn);

comm.ExecuteScalar();

conn.Close();

Измененные данные попадают в таблицу и их сразу можно увидеть.

Листинг 10. Формирование отчета в Excel

//создаем виртуальный объект Excel

Microsoft.Office.Interop.Excel.Application myExcel = new Microsoft.Office.Interop.Excel.Application();

//создать книгу в объекте Excel

myExcel.Application.Workbooks.Add(Type.Missing);

//название страницы

DateTime dt = DateTime.Now;

string curDate = dt.ToShortDateString();

myExcel.ActiveSheet.Name = "Отчет за " + curDate;

//Настраиваем ячейки

myExcel.Columns.ColumnWidth = 20;

//Пишем заголовки ячеек

for (int i = 0; i < dataGridView1.ColumnCount; i++)

{

myExcel.Cells[1, i + 1] = dataGridView1.Columns[i].HeaderText;

// заголовки программно

}

for (int i = 0; i < dataGridView1.ColumnCount; i++)

{

for (int j = 0; j < dataGridView1.RowCount - 1; j++)

{

myExcel.Cells[j + 2, i + 1] = dataGridView1[i, j].Value.ToString();

}

}

//скрываем столбец А

Range range = (Range)myExcel.Columns[1, Type.Missing];

range.EntireColumn.Hidden = true;

myExcel.Visible = true;

}

В отчете показывается имя игрока, его счет, дата и время игры. Первый столбец с айди игрока скрывается программно. Готовый отчет можно сохранить.

Так же в этой форме можно формировать списки по дате (листинг 11), по счету (листинг 12), и по имени текущего пользователя (листинг 13).

Формирование списков проходит через запросы к базе данных. После отправки запроса можно увидеть измененную таблицу. Списки показываются выбором определенного радиобаттона.

Листинг 11. Формирование списка по дате игры

else if (radioButton2.Checked)

{

string s = "SELECT Id, Name\_players as 'Имя игрока', score as 'Счёт', data\_igra as 'Дата', vremya\_igra as 'Время' FROM PLAYERS ORDER BY data\_igra DESC OFFSET " + n.ToString() + "ROWS FETCH NEXT 10 ROWS ONLY ";

SqlDataAdapter adap = new SqlDataAdapter(s, conn);

DataSet ds = new DataSet();

adap.Fill(ds);

dataGridView1.DataSource = ds.Tables[0];

if (ds.Tables[0].Rows.Count >= 10)

{

button1.Enabled = true;

button2.Enabled = false;

}

else

{

button1.Enabled = false;

button2.Enabled = true;

}

Листинг 12. Формирование списка по счету

else if (radioButton1.Checked)

{

string s = "SELECT Id, Name\_players as 'Имя игрока', score as 'Счёт', data\_igra as 'Дата', vremya\_igra as 'Время' FROM PLAYERS ORDER BY id OFFSET " + n.ToString() + " ROWS FETCH NEXT 10 ROWS ONLY";

SqlDataAdapter adap = new SqlDataAdapter(s, conn);

DataSet ds = new DataSet();

adap.Fill(ds);

dataGridView1.DataSource = ds.Tables[0];

if (ds.Tables[0].Rows.Count >= 10)

{

button1.Enabled = true;

button2.Enabled = false;

}

else

{

button1.Enabled = false;

button2.Enabled = true;

}

}

Листинг 13. Демонстрация счета авторизированного игрока

SqlConnection connect = new SqlConnection(Program.st\_connect);

string s = "SELECT Id\_user, Name\_players as 'Имя игрока', score as 'Счёт', data\_igra as 'Дата', vremya\_igra as 'Время' FROM PLAYERS INNER JOIN USERS on PLAYERS.id\_players=USERS.id\_user WHERE Id\_user= " + Program.id\_user.ToString();

SqlDataAdapter adap = new SqlDataAdapter(s, conn);

DataSet ds = new DataSet();

adap.Fill(ds);

dataGridView1.DataSource = ds.Tables[0];

Изменения игроков и формированием списков завершено, перейдем к самой игре.

Игра представляет собой механизм разбивания блоков, подсчета очков, так же необходимо не упустить шарик, орудие игры. Внизу экрана игроком передвигается платформа, которой нужно отбивать шарик. Блоки для разбивания строятся циклом, цвет блоков подбирается случайно. Перед инициализацией формы необходимо указать название рандома, на который ссылается цикл Foreach.

Листинг 14. Расстановка блоков

private Random rnd = new Random();

public Form1()

{

InitializeComponent();

foreach (Control x in this.Controls)

{

if (x is PictureBox && x.Tag == "block")

{ Color RandomColor = Color.FromArgb(rnd.Next(256), rnd.Next(256),

x.BackColor = RandomColor;

}

}

}

В листинге 15 можно увидеть механизм игры. ball – это шарик, ballx ballly – его координаты по осям (необходимы для столкновений с блоками или границами игры), player – платформа, отбивающая шарик, goleft и gorigt – направления движения платформы, право и лево. Speed – скорость движения платформы.

Листинг 15.

private void timer1\_Tick(object sender, EventArgs e)

{

ball.Left += ballx;

ball.Top += bally;

label1.Text = "Очки " + score;

if (goLeft)

{

player.Left -= speed;

}

if (goRight)

{

player.Left += speed;

}

if (player.Left < 1)

{

goLeft = false;

}

else if (player.Left + player.Width > 670)

{

goRight = false;

}

if (ball.Left + ball.Width > ClientSize.Width || ball.Left < 0)

{

ballx = -ballx;

}

if (ball.Top < 0 || ball.Bounds.IntersectsWith(player.Bounds))

{

bally = -bally;

}

if (ball.Top +ball.Height > ClientSize.Height)

{

GameOver();

MessageBox.Show("Вы проиграли");

timer1.Stop();

}

foreach(Control x in this.Controls)

{

if (x is PictureBox && x.Tag == "block")

{

if (ball.Bounds.IntersectsWith(x.Bounds))

{

this.Controls.Remove(x);

bally = -bally;

score++;

}

}

}

if (score > 34)

{

GameOver();

MessageBox.Show("Вы победили");

timer1.Stop();

}

}

this.Text = "Имя " + player.Text.ToString() + ", Счет " + score;

По завершении игры (если мячик упущен или разбиты все блоки) останавливается таймер далее происходит запись в базу данных. Записывается id игрока, его имя, счет, дата и время игры. Окно игры закрывается и выводится сообщение о проигрыше или выигрыше, открывается форма, из которой моно посмотреть свой счет или начать игру заново. См. листинг 16.

Листинг 16. Конец игры

void GameOver()

{

timer1.Stop();

SqlConnection conn = new SqlConnection(Program.st\_connect);

conn.Open();

DateTime dateTimeVariable = DateTime.Now;

string date1 = dateTimeVariable.ToString("yyyy-MM-dd");

string date2 = dateTimeVariable.ToString("H:mm:ss");

string s = "INSERT INTO PLAYERS " +

"(id\_players, Name\_players, score, data\_igra, vremya\_igra) VALUES" +

"('" + Program.id\_user + "'," +

"'" + player.Text.ToString() + "'," +

"'" + score + "', " +

"'" + date1 + "', " +

"'" + date2 + "')";

SqlCommand command = new SqlCommand(s, conn);

SqlDataReader reader = command.ExecuteReader();

conn.Close();

this.Close();

}

## **3.3 Руководство пользователя**

Запуск программы. Программа запускается с ярлыка приложения. Сразу после открывается форма авторизации.

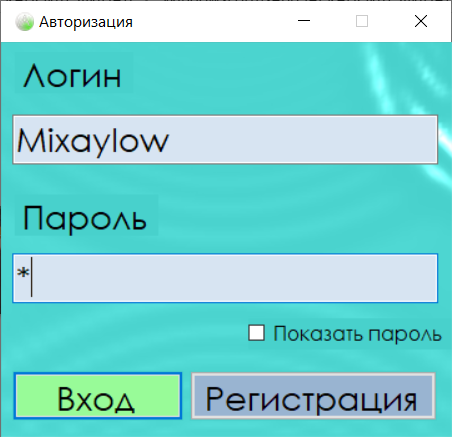


Рисунок 4 - «Окно авторизации»

Для входа необходимо ввести уже зарегистрированные учетные данные, либо пройти регистрацию, нажав по кнопке «Регистрация»

Кнопка «Регистрация» открывает окно регистрации нового пользователя. В ней необходимо внести логин, пароль и предоставить свое фото. Данные записываются в базу данных игры и хранятся там. Сразу после регистрации можно авторизоваться под новыми учетными данными.

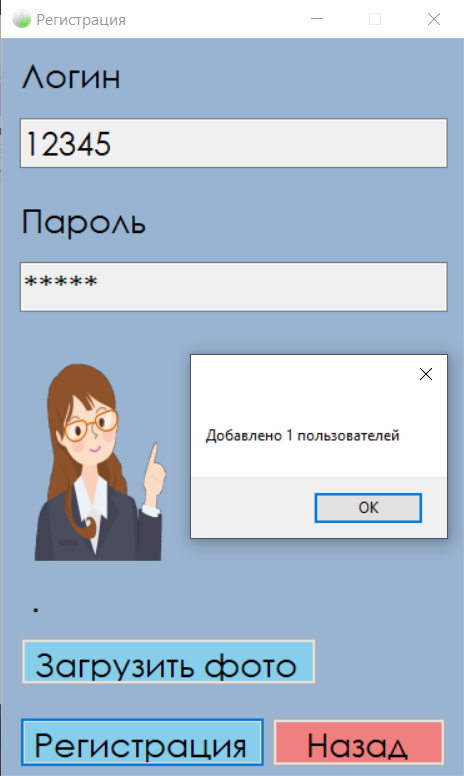


Рисунок 5 - «Регистрация нового пользователя»

После успешной авторизации открывается окно пользователя (или администратора), в котором можно выбрать игру, посмотреть статистику и т.д.



Рисунок 6 - «Форма администратора»

В форме администратора реализована возможность изменения данных пользователя, просмотр и изменения игровых итогов игроков, запустить игру и покинуть форму.

Кнопка «Изменить юзера» осуществляет переход на форму изменения пользователей. В ней можно менять логин, пароль, тип юзера (админ, не админ) и фотографию (аватарку). См. рис. 7.

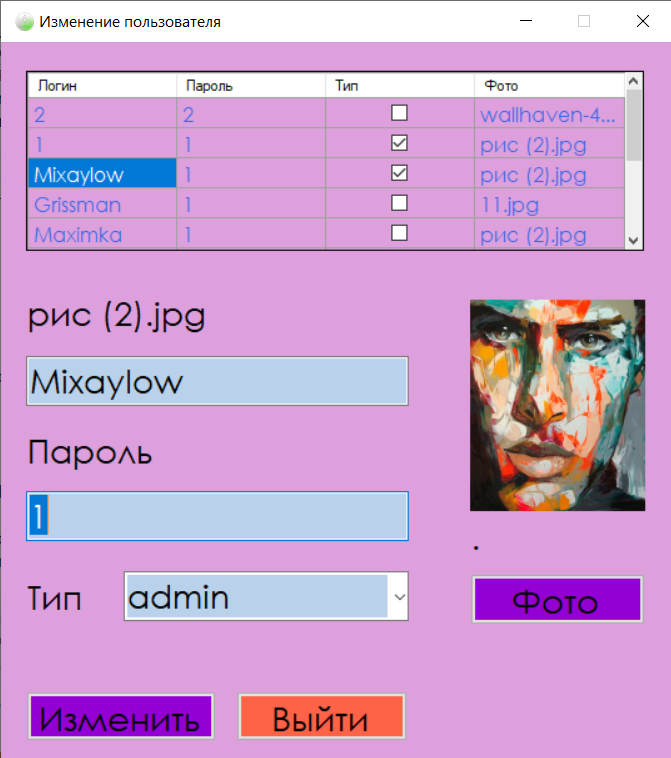


Рисунок 7 - «Форма изменения пользователей»

Кнопка «Изменить игрока»

В этом окне (рис. 8) формируется таблица с данными игроков, учувствовавших в игре. Здесь можно увидеть имя игрока, его счет, дату и время игры. Здесь осуществляется листание игроков (кнопки << и >>), игрока можно удалить (кнопка «Удалить»), изменить его данные (кнопка «Изменить»). Список можно сформировать по счету, по дате (Радиобаттоны). Данные можно вывести в Excel документ (кнопка «Отчет в Excel»).

При нажатии кнопки «Изменить» окно изменяется и открывается дополнительное меню (рис 9). В нем можно менять данные игрока. Измененные данные сохраняются в базе.

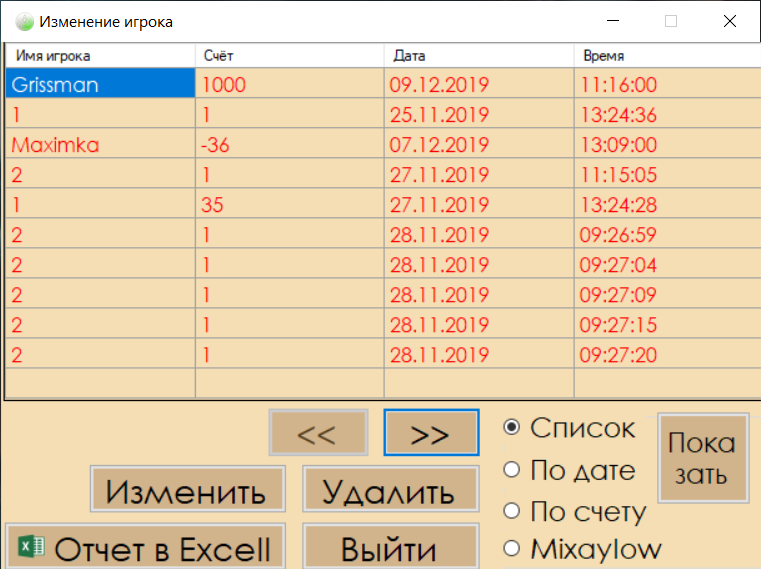


Рисунок 8 - «Форма изменения игроков»

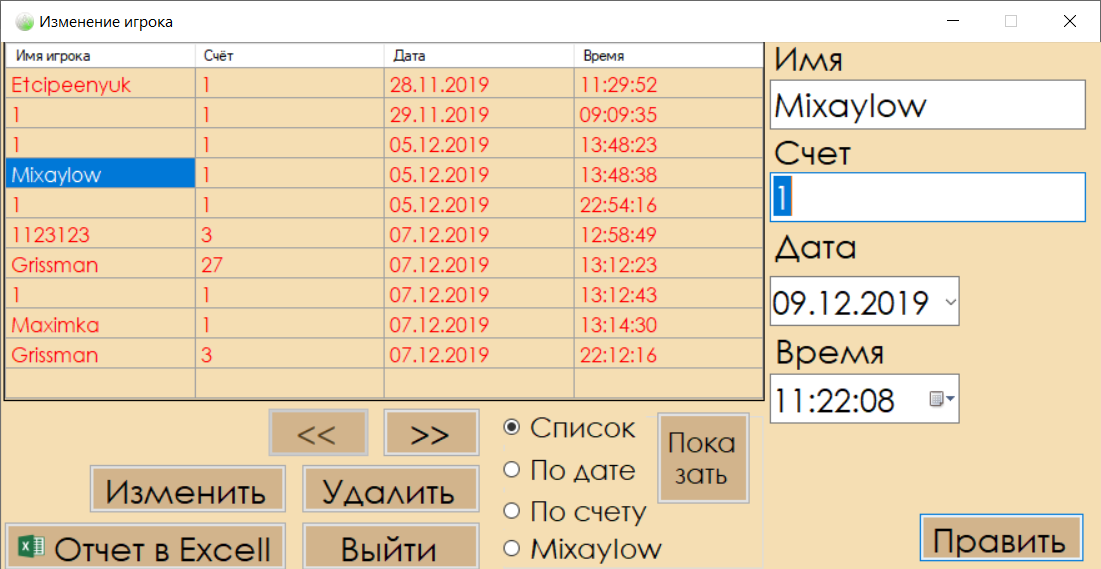


Рисунок 9 - «Расширенная форма изменения игроков»

Кнопка «Игра» начинает саму игру. Игра представляет собой несложный процесс разбивания блоков (с ведением счета) и отбивание шарика плавающей внизу экрана платформой. На платформе видно имя игрока, так же оно написано в заголовке формы. Там же можно увидеть и свой текущий счет. Данные так же видны и на самой форме, в верхнем левом углу.



Рисунок 10 - «Игра»

После завершения игры (разбиты все блоки, шарик упущен или игра закрыта) в базу поступают данные игрока. Имя, счет, дата и время игры, которые потом можно просмотреть или изменить (если вы админ) в статистике.

Кнопка «Выйти» закрывает форму, после открывается форма авторизации.

Форма игрока гораздо проще. В ней по сравнению с формой админа функционал ограничен. Здесь можно посмотреть статистику, не изменяя её и запустить игру.



Рисунок 11 - «Форма пользователя»

Кнопка «Статистика» открывает упрощенную форму изменения игроков. В ней скрыты кнопки, которые были доступны в режиме админа. Оставлена только возможно листания таблицы, формирования рейтингов, и вывод статистики в Excel. См. рис. 12.

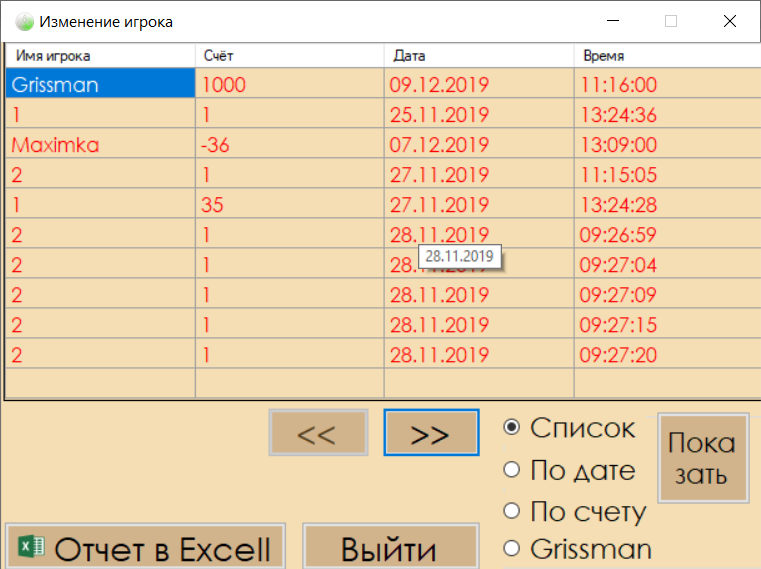


Рисунок 12 - «Упрощенная форма изменения игроков»

Остальные кнопки в форме игрока выполняют аналогичные роли, что и в форме администратора.

**3.4 О программе**

В игре имеется меню «О программе». Вызывается кнопкой знак вопроса в области кнопок управления окном.



Рисунок 13 - «Кнопка вызова меню «О программе»»

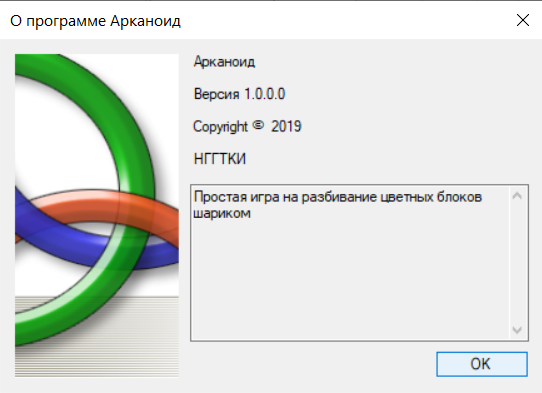


Рисунок 14 - «Кнопка вызова меню «О программе»»

В этой форме можно найти информацию о программе, и нехитрое описание игры.

# **ЗАКЛЮЧЕНИЕ**

В результате курсовой работы было проведено исследование предметной области, была разработана игра «Арканоид» с использованием СУБД MS SQL-server, изучена среда разработки, разработан интерфейс проекта с привлекательным дизайном, проведено тестирование и отладка программного обеспечения, произведен сбор данных об объектах.

Разработанный проект позволит пользователям системы играть в игру «Арканоид», сохранять и просматривать свои результаты.

В дальнейшем планируется провести более детальную проработку интерфейса приложения.

Цель достигнута, задачи выполнены.

# **СПИСОК ЛИТЕРАТУРЫ**

1. Васильев, Алексей C#. Объектно-ориентированное программирование / Алексей Васильев. - М.: Питер, 2012. - 320 c.
2. Подбельский, В. В. Язык С#. Базовый курс / В.В. Подбельский. - М.: Финансы и статистика, 2013. - 408 c.
3. Троелсен, Эндрю Язык программирования C# 5.0 и платформа .NET 4.5 / Эндрю Троелсен. - М.: Вильямс, 2015. - 633 c
4. Вишневский, Алексей Microsoft SQL Server. Эффективная работа / Алексей Вишневский. - М.: Питер, 2009. - 143 c.
5. Ицик, Бен-Ган Microsoft SQL Server 2012. Высокопроизводительный код T-SQL. Оконные функции / Бен-Ган Ицик. - М.: Русская Редакция, 2013. - 861 c.