СОДЕРЖАНИЕ

[Введение 5](#_bookmark0)

1. [Технический проект 6](#_bookmark1)
   1. [Описание предметной области 6](#_bookmark2)
   2. [Технология обработки информации в системе 7](#_bookmark3)
   3. [Основные алгоритмы 8](#_bookmark4)
   4. [Цель и назначение 8](#_bookmark5)
   5. [Обзор аналогов 9](#_bookmark6)
   6. [Требования к техническому и программному обеспечению 10](#_bookmark6)
2. [Рабочий проект 11](#_bookmark7)
   1. [Общие сведения о системе 11](#_bookmark8)
   2. [Функциональное назначение 15](#_TOC_250002)
   3. [Инсталляция программного продукта 15](#_TOC_250001)

2.5 Руководство пользователя 15

* 1. [Сообщения системы 15](#_TOC_250000)
  2. [Обработка исключений 16](#_bookmark9)

1. [Методика испытания и тестирования 17](#_bookmark7)

[Заключение 21](#_bookmark10)

[Список используемой литературы 22](#_bookmark11)

[Приложение 1. Блок-схема алгоритма 23](#_bookmark11)

[Приложение 2. Оптический носитель информации 25](#_bookmark11)

ВВЕДЕНИЕ

Интеллектуальная игра — это вид игры, основывающийся на применении игроками своего интеллекта и/или эрудиции. Как правило, в таких играх от участников требуется отвечать на вопросы из различных сфер жизни. Интеллектуальные игры распространены в массмедиа, в первую очередь, на телевидении, где победитель награждается какими-либо призами.

“Своя игра” – игра-викторина, в которой три человека должны отвечать на тематические вопросы из различных областей знаний. Каждый вопрос имеет свою стоимость, прямо пропорциональную его сложности. За правильный ответ игрок получает очки, равные стоимости вопроса, за неправильный - теряет. Игрок, набравший наибольшее количество очков в конце игры, объявляется победителем.

В настоящее время телевизионный формат “Своей игры” по популярности стоит наравне с такими проектами, как “Умники и умницы”, “Что? Где? Когда?” и “Кто хочет стать миллионером?” Она существует и в виде спортивной интеллектуальной игры, по которой проводится множество соревнований не только в нашей стране, но и в некоторых приближенных странах СНГ, при этом являясь одной из трех основных дисциплин (“Что? Где? Когда?”, “Брейн ринг” и “Своя игра”)

В данной курсовой работе будет рассматриваться одиночная версия “Своей игры” компьютер-человек. Задача человека – набрать максимально возможное количество очков, ответив на все вопросы пакета. Победная стратегия – закончить игру, имея положительное количество очков.

1 ТЕХНИЧЕСКИЙ ПРОЕКТ

* 1. Описание предметной области

Предметная область данной работы – интеллектуальная игра “Своя игра”. Реализация программы будет осуществляться за счет интерактивных полей ввода, кнопок и программных сообщений.

Концепция игры:

Вводимыми данными пользователя будут – его имя, аватар и файл пакета вопросов.

После ввода стартовой информации и нажатия кнопки “старт”, перед пользователем будет появляться следующая форма с таблицей вопросов, вместо самих вопросов в таблице будут выведена стоимость каждого вопроса.

Пакет вопросов – это xml файл таблицы, его компиляция будет производиться с помощью дополнительной программы “quester”.

Вопросы могут быть текстовыми, с картинкой, с видео файлом и с аудио файлом.

После выбора игроком вопроса, таблица сменяется текстом вопроса. Если к вопросу прилагается медиа файл, то он будет заменять текстовый вопрос через 10 секунд, чтобы у игрока было время для прочтения вопроса.

После вывода медиа файла, у игрока будет 15 секунд на ответ. В случае если за это время игрок ответит на вопрос правильно, его счет увеличится на количество очков, равное стоимости вопроса. В случае если игрок ответит неправильно, или не успеет ответить за 15 секунд, он потеряет количество очков, равное стоимости вопроса.

Победой считается положительное количество очков у игрока после ответа на все вопросы пакета. Помимо этого, в таблице рекордов будет вестись статистика рейтинга по количеству очков, набранному определенным игроком в определённом пакете вопросов.

Таблица рекордов будет реализована с помощью базы данных на основе SQL сервера.

* 1. Технология обработки информации в системе

Диаграмма Use Case - это схема вариантов использования, которая обобщает, кто именно использует приложение или систему и что именно он может с ними делать.

На схеме вариантов использования не отображаются подробности о вариантах использования: на ней представлены только некоторые отношения

между вариантами использования, субъектами и системами. В частности, схема показывает порядок выполнения шагов для достижения цели каждого из вариантов использования. Эти сведения можно описать и в других схемах и

документах, которые можно связать с каждым вариантом использования, но Use Case одна из самых удобных.

В разрабатываемой программе предусматривается один Пользователь.

Объектно-ориентированный анализ предметной области выявил основные функции Пользователя программы.

Диаграмма вариантов использования (Use Case Diagram) выполнена в нотации универсального языка моделирования UML и представлена на рис. 1.1.

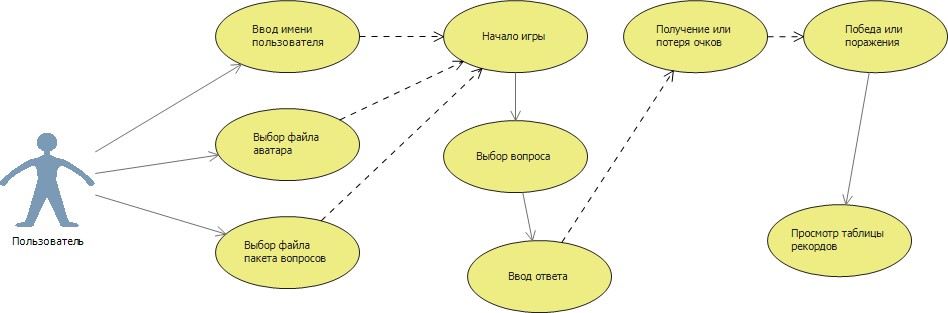


Рисунок 1.1 – Диаграмма вариантов использования (Use Case Diagram)

Пользователь, запустив программу, может вводить персональные данные в любом порядке, но перейти на форму самой игры не сможет, пока всё не введет.

Нажав на кнопку “Начать игру!”, пользователь переходит в форму игры, где может выбрать любой вопрос из списка. После выбора вопроса, у него будет 15- 25 секунд на ответ. Если за это время игрок сможет правильно ввести ответ, ему начисляться очки, в обратном – снимут.

После того, как пользователь сыграет все вопросы, с помощью элемента “MessageBox”, программа оповестит его о конечном результате, после нажатия кнопки согласия на этом сообщении, данные о результате игрока запишутся в таблицу рекордов, а игрока отправят на начальную форму, где он сможет перейти на форму просмотра таблицы рекордов.

* 1. Основные алгоритмы

Основнымы алгоритмами программы являются алгоритмы анализа типа вопросов и сортировка используемых внешних данных по соответствующим элементам формы WPF, проигрывающим медиа файлы. Так же присутствуют алгоритм зачисления очков, алгоритм подсчета финального результата и алгоритм записи или изменения данных об игровой сессии в таблице рекордов. Подробный алгоритм в виде блок-схемы можно увидеть в Приложении 1.

* 1. Цель и назначение

Целью данной курсовой работы является создание интерактивной игры

“Своя игра”, позволяющей пользователю отвечать на вопросы из разных областей истории, науки и культуры, используя свой интеллект и эрудицию.

Назначение программы – развлечение Пользователя, путем проверки его ума и смекалки.

Для решения поставленной задачи потребуется написать программный код интерактивной игры “Своя игра”. Для этого будет использоваться клиентское приложение Windows Presentation Foundation сокращенно WPF.

Windows Presentation Foundation (WPF) – система для построения

клиентских приложений Windows с визуально привлекательными возможностями взаимодействия с пользователем.

В основе WPF лежит векторная система визуализации, не зависящая от разрешения устройства ввода и созданная с учётом возможностей современного графического оборудования. WPF предоставляет средства для создания визуального интерфейса, включая язык XAML (eXtensible Application Markup Language), элементы управления, привязку данных, макеты, двухмерную и трёхмерную графику, анимацию, стили, шаблоны, документы, текст, мультимедиа и оформление.

Графической технологией, лежащей в основе WPF, является DirectX, в отличие от Windows Forms, где используется GDI/GDI+. Производительность WPF выше, чем у GDI+ за счёт использования аппаратного ускорения графики через DirectX.

* 1. Обзор аналогов

Самый популярный в интернете аналог – “SIGame: Компьютерная игра”, который можно установить на сайте https://vladimirkhil.com/si/game.

Преимущества аналога:

* + - Сетевая игра с другими людьми.
    - Возможность одиночной игры против компьютера в виде ботов.
    - Качественная графическая и логическая реализация.
    - Локальная игра.
    - Наличие дополнительного ПО в виде программы, позволяющей создавать свои пакеты вопросов.
    - Возможность играть, не устанавливая приложения на сайте - [http://vladimirkhil.com/si/online.](http://vladimirkhil.com/si/online)

Недостатки аналога:

* + - Слабый сервер, периодически подвергаемый атакам.
    - Плохая оптимизация работы с файлами тяжелее десяти мегабайт. Скриншот игры представлен на рис. 1.2.



Рисунок 1.2 – Одиночный режим игры SIGame

* 1. Требования к техническому и программному обеспечению

*Требования к техническому обеспечению:*

* + - Процессор не ниже 1,6 ГГц;
    - ОЗУ не меньше 1 Гб;
    - Монитор
    - Клавиатура;
    - Мышь;
    - Средство воспроизведения звука (Звуковые колонки, наушники…);

*Требования к программному обеспечению:*

* + - Операционная система Windows XP и выше;
    - Платформа .NET Framework 4.
    - SQL Server 2010.

1. РАБОЧИЙ ПРОЕКТ
   1. Общие сведения о системе

Программа разработана на языке программирования C# в среде разработки Microsoft Visual Studio 2010 в приложении WPF. У данной программы два функциональных программных модуля. Собственно, сама игра, и квестер, позволяющий составлять пакеты вопросов. В своей работе программа использует процедуры и функции стандартных библиотек языка C# 4.0.

Программа состоит из трех форм.

Первая форма MainWindow.xaml содержит в себе элемент типа Label “Имя игрока”, 4 кнопки “Выбрать пакет вопросов”, “Аватар…”, “Начать игру!” и “Таблица рекордов”, а так же два элемента типа Textbox - “namebox” и “pucklink”.

Вторая форма - Game.xaml наполнена объектами, осуществляющими вывод информации из пакета вопросов пользователю. Например, для вывода аудио и видео файлов используется медиаэлемент “audioquest”, а для таблицы вопросов – объект типа Listbox “siq”. Так же сюда транслируются аватар и имя пользователя. Элемент типа Label “scorelabel” отображает текущий счет игрока, а элемент типа Label “cost” – стоимость текущего вопроса. Для визуального удобства объект Progressbar “PG” представляет собой индикатор оставшегося на ответ времени. Ответ вводится в Textbox “AnswerBox”.

Третья форма – таблица рекордов, реализованная с помощью базы данных

SQL.

Общая длина кода игры (LOC) равна 410 строкам.

Комплексный показатель качества кода игры (*Maintainability Index*) разработанного программного обеспечения равен 87.

Общая длина кода квестера (LOC) равна 54 строкам.

Комплексный показатель качества кода приложения (*Maintainability Index*) разработанного программного обеспечения равен 71.

* 1. Функциональное назначение

Результатом выполнения курсового проекта является готовый программный продукт, позволяющий пользователю стать победителем интерактивной игры “Своя игра”.

Данная программа обрабатывает и выводит пакет вопросов в виде таблицы стоимостей вопросов. Её задача – позволить игроку отвечать на вопросы пакета и обрабатывать вводимые игроком данные ответа с указанием их правильности.

К её возможностям можно причислить вывод вопросов с медиа фалами разных форматов, таких как mp3, mp4, jpg, png и т.д.

К достоинствам программы можно отнести интуитивно понятный интерфейс, а так же разнообразные элементы WPF, влияющие на облегчение визуального восприятия игроком информации, такие как “ProgressBar”, отражающий течение времени таймеров, всплывающие окна, оповещающие игрока о результатах его взаимодействия с программой. Действие этого элемента начинается через 10 после вывода текстовой части вопроса в случае если к вопросу прилагается медиа файл, в обратном случае, его действие таймера, ограничивающего время на ответ, начинается сразу после появления вопроса. До начала действия таймера элемент “Listbox”, предназначенный для ввода ответа недоступен для изменения. Так же, достоинствами программы является возможность создания своих собственных пакетов вопросов с помощью дополнительного приложения “quester” и ведение учета рекордов каждого игрока в формате таблицы, с указанием максимального количества очков, набранного игроком в этом пакете вопросов.

Так же в программе присутствует база данных построенная с помощью SQL сервера, предназначение которой – хранение статистических рекордных данных о результатах игроков.

Главным недостатком программы является невозможность открытия таблицы рекордов до того, как в ней появится хотя бы одна запись.

* 1. Инсталляция программного продукта

Инсталляция программного продукта происходит при помощи приложения setup.exe.

Состав файла инсталляции:

* Проприетарные файлы образа диска, содержащие файл базы данных Database1.mdf и Database1\_log.mdf.
* Сторонние файлы, используемые программой:
  + Файл картинки скрипичного ключа, используемый в аудио вопросах ef.png.
  + Файл аудио трека, используемый в аудио вопросе тестового пакета вопросов moon.mp3.
  + Файл картинки, используемой в вопросе с картинкой opyatdvoika.jpg.
  + Файл картинки аватара пользователя padoru.jpg.
  + Файл пакета вопросов test1.xml.
* Файл программного приложения WpfApplication1.exe.
* Файл базы данных программы WpfApplication.pdb.
* Файл WpfApplication1.vhost.exe, нужный для ускорения и упрощения отладки.

Скриншот состава файла инсталляции представлен на рис. 2.1.

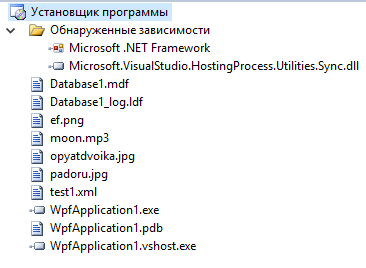


Рисунок 2.1 – Состав файла инсталляции

* 1. Руководство пользователя

При запуске программы пользователю представляется форма ввода начальных персональных данных, которая представлена на рис. 2.2.

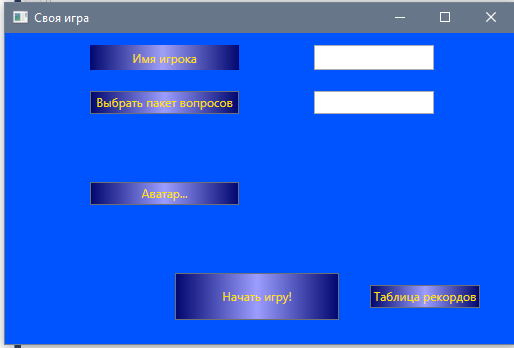


Рисунок 2.2 - Главное диалоговое окно

На этой форме игрок вводит имя с помощью элемента “TextBox”, аватар и пакет вопросов с помощью соответствующих кнопок, а с помощью кнопки “Таблица рекордов” может открыть таблицу рекордов.

После нажатия на кнопку “Начать игру”, открывается форма игра, которая представлена на рис. 2.3.

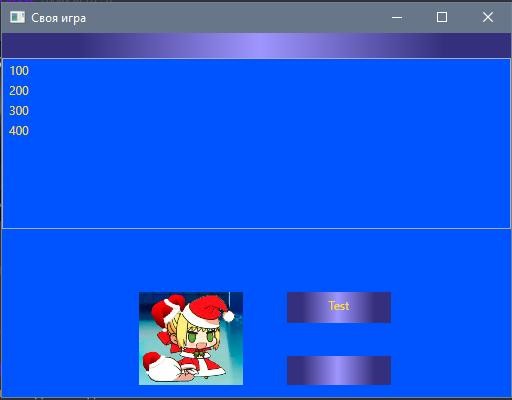


Рисунок 2.3 - Интерфейс основной формы

Форма содержит 3 элемента типа “Image”, которые содержат аватар, картинку с дополнительным файлом вопроса, в случае если тип вопроса – “С картинкой” и картинку со скрипичным ключом, которая будет выводиться на время проигрывания аудио файла в случае, если тип вопроса – “Аудио”. Так же в форме есть 3 элемента типа “Label” для имени игрока, его счета и для стоимости текущего вопроса. Два элемента типа “Textbox” для текста вопроса и текста вводимого ответа. “MediaElement” для медиа файлов, “ListBox” для таблицы вопросов и “ProgressBar”.

На рис. 2.4 можно увидеть форму с выбранным вопросом типа “Аудио”.

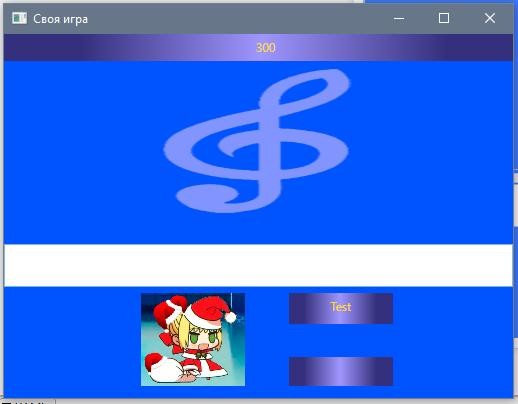


Рисунок 2.4 – Интерфейс с выбранным вопросом

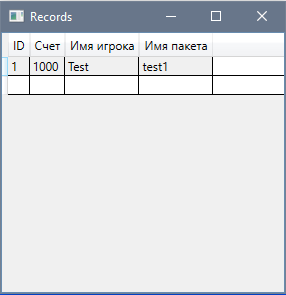
На рисунке 2.5 представлена таблица рекордов, о которой говорилось выше.

Рисунок 2.5 – Таблица рекордов

* 1. Сообщения системы

В табл. 2.1 перечислены сообщения системы, причины их возникновения и действия пользователя.

Таблица 2.1 - Сообщения системы

|  |  |
| --- | --- |
| **Сообщение** | **Причины и действия пользователя** |
| Имя пользователя не корректно. | Не указано имя пользователя. Следует указать имя пользователя. |
| Не выбран пакет. | Не выбран пакет вопросов. Необходимо выбрать пакет вопросов. |
| Файл пакета вопросов не корректен. | Выбран файл неподходящего формата. Необходимо выбрать другой файл. |
| В таблице рекордов нет записей. | В таблице рекордов еще нет записей, чтобы добавить их, необходимо закончить хотя бы одну партию игры. |
| Ответ правильный + ‘стоимость’ | Игрок правильно ответил на вопрос, к его счету прибавляется количество очков, равные стоимости вопроса. |
| Ответ неправильный – ‘стоимость’ | Игрок неправильно ответил на вопрос, из его счета вычитается количество очков, равные стоимости вопроса. |

* 1. Обработка исключений

В программе для обработки исключений используется оператор Try – Catch.

Использование оператора обработки исключений позволяет избежать аварийного завершения работы программы, вместо чего выводится системное сообщение о типе возникшей ошибки и завершение выполнения процедуры, вызвавшей ошибку.

В данной работе ошибка выполнения может возникнуть вследствие неверного ввода пользователем данных (пример: вместо xml файла пакета вопросов – png файл), но это не приведет к аварийному завершению программы - пользователь получит соответствующее сообщение и работа процедуры обработки события завершится. После чего пользователь сможет исправить введенные значения. На рисунке 2.6 представлена часть кода с элементом Try-Catch.

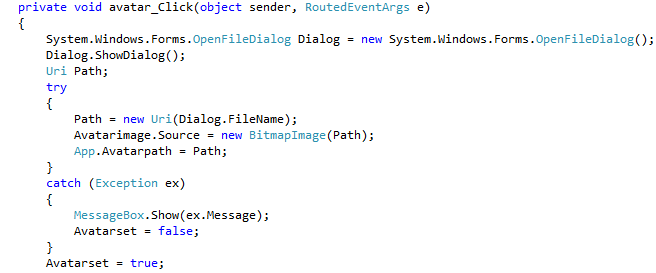


Рисунок 2.6 – Часть кода с Try-Catch

Суть данной команды Try-Catch состоит в том, что программа проверяет формат файла аватар, значение которого представляет собой переменная Avatarimage.

Если же формат файла является неподходящим, программа выводит системное сообщение об ошибке.

1. МЕТОДИКА ИСПЫТАНИЯ И ТЕСТИРОВАНИЯ

Для проверки работоспособности программы следует создать тестовый пакет вопросов, который представлен на рисунке 3.1.

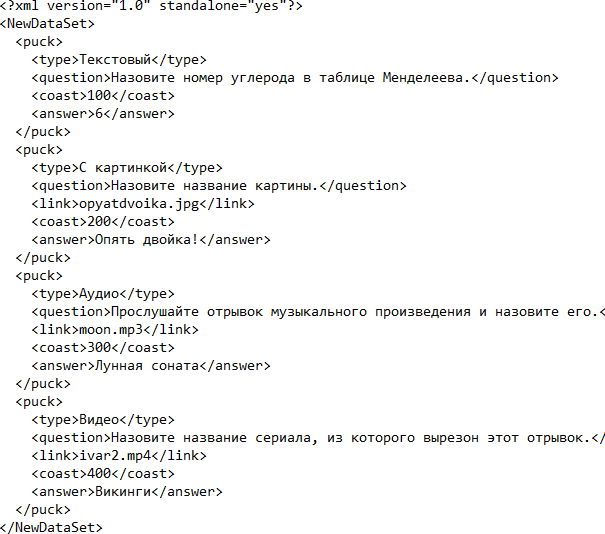


Рисунок 3.1 – Тестовый пакет вопросов

Затем следует ввести персональные данные, как показано на рисунке 3.2.



Рисунок 3.2 – Форма с тестовыми данными

Для того чтобы начать игру, необходимо сначала выбрать вопрос и ответить на него. Скриншот представлен на рисунке 3.3.

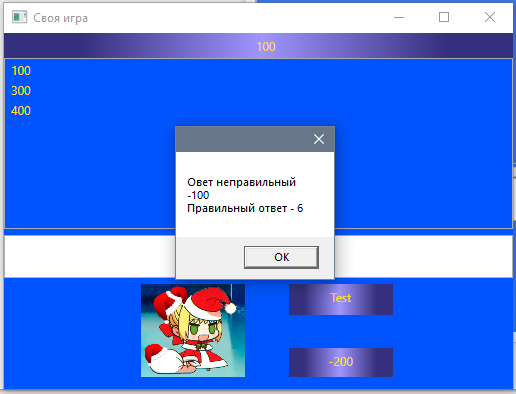


Рисунок 3.3 – Сообщение о том, что ответ неправильный

Далее необходимо выбрать видео вопрос и ответить на него неправильно. Скриншот представлен на рисунке 3.4.

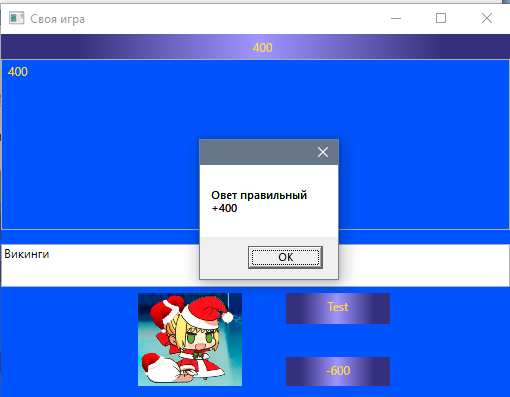


Рисунок 3.4 – Сообщение о том, что ответ правильный.

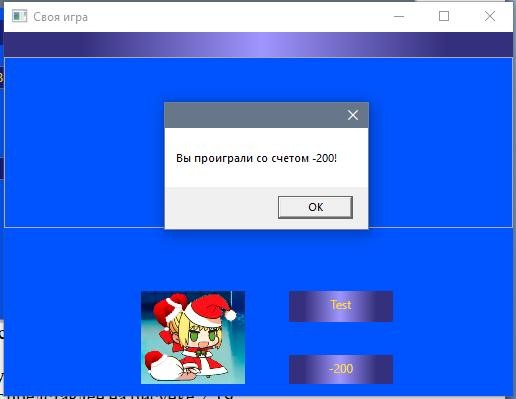
После ответов на все вопросы, счет игрока остался отрицательным, что привело к проигрышу, пример проигрыша представлен на рисунке 3.5.

Рисунок 3.5- Сообщение о проигрыш

ЗАКЛЮЧЕНИЕ

Курсовая работа выполнена в соответствии с заданием и соответствует требованиям выполнению курсовой работы по дисциплине «Основы алгоритмизации и программирование»

Результаты работы:

* + - Разработана программа интерактивной игры “Своя игра” в среде разработки Microsoft Visual Studio 2010.
    - Работоспособность программы проверена с помощью большого количества тестов.

Программа проста в использовании, командный интерфейс интуитивно понятен, изменить его можно только через программный код в настройках трех форм. К программе прилагается дополнительное приложение “quester”, позволяющее создавать собственные пакеты вопросов.

Разработанный программный продукт может применяться для одиночной игры в “Свою игру”.

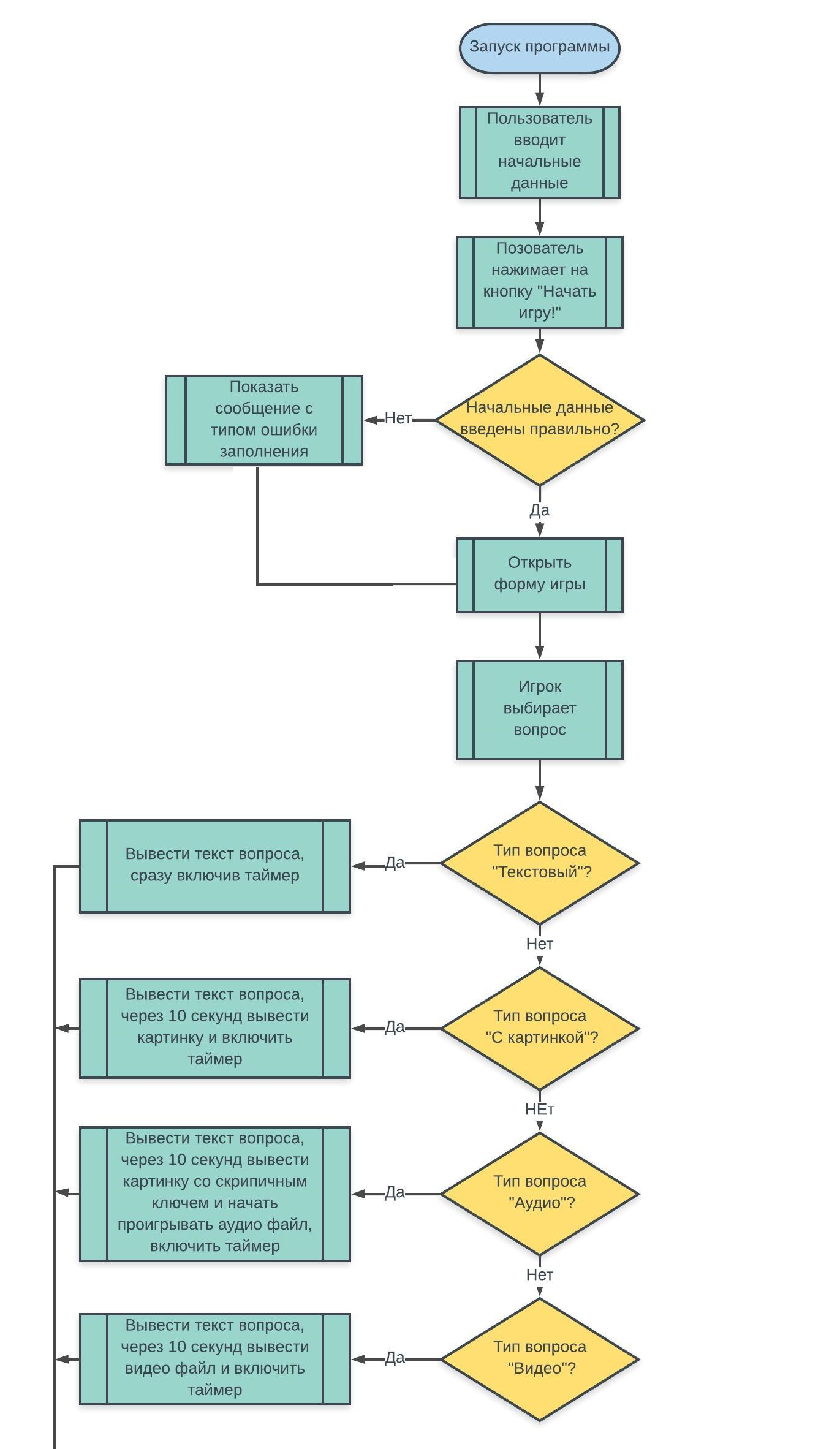
Дальнейшее развитие работы предполагается в создании ботов – противников человека и в улучшении интерфейса.

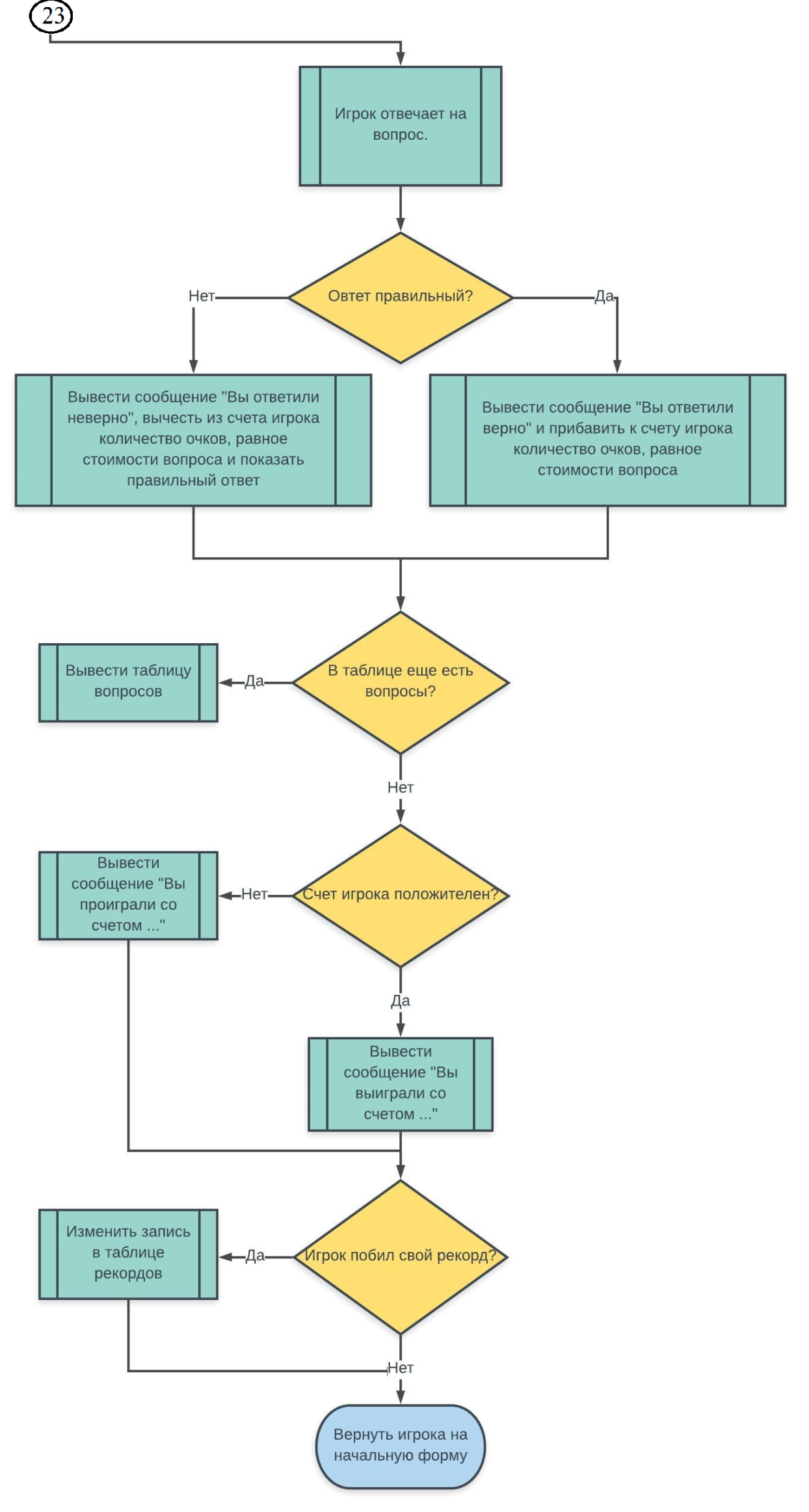
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Дрейер М. C# для школьников: Учебное пособие. Перевод с англ. под ред. В. Биллига - М.: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2009. - 128 с.
2. Фаронов В. Программирование на языке С #. – СПб.: Питер, 2007, - 240с.
3. Шилдт Герберт. С# 4.0 – полное руководство, 2011 Вильямс, 1056 с
4. Лабор В. В. Си Шарп: Создание приложений для Windows — Мн.: Харвест, 2003. -384 с.
5. Мартынов Н.Н. С# для начинающих. – М.: КУДИЦ – ПРЕСС, 2007. - 272 с.
6. Свободная энциклопедия WikiPedia [Электронный ресурс]. Режим доступа: https://ru.wikipedia.org/wiki/Своя\_игра.
7. Сайт разработчика SIgame: Компьютерная игра [Электронный ресурс]. Режим доступа: https://vladimirkhil.com/si/game.

ПРИЛОЖЕНИЕ 1

**Блок-схема алгоритма**

****

****

ПРИЛОЖЕНИЕ 2

**Оптический носитель информации**