Оглавление

[Введение 4](#_Toc358893259)

[1. Требования к программе 5](#_Toc358893260)

[1.1. Техническое задание 5](#_Toc358893261)

[1.1.1. Введение 5](#_Toc358893262)

[1.1.2. Основание для разработки 5](#_Toc358893263)

[1.1.3. Назначение разработки 6](#_Toc358893264)

[1.1.4. Требования к программе или программному изделию 6](#_Toc358893265)

[1.1.5. Технико-экономические показатели 8](#_Toc358893266)

[1.1.6. Стадии и этапы разработки 8](#_Toc358893267)

[1.1.7. Порядок контроля и приемки 9](#_Toc358893268)

[1.2. Варианты использования 9](#_Toc358893269)

[1.2.1. Вариант использования «Start Game» 10](#_Toc358893270)

[1.2.2. Вариант использования «Touch the mouse» 10](#_Toc358893271)

[1.2.3. Вариант использования «Exit to main menu» 11](#_Toc358893272)

[1.2.4. Вариант использования «Login to Facebook» 11](#_Toc358893273)

[1.2.5. Вариант использования «Logout» 13](#_Toc358893274)

[1.2.6. Вариант использования «Save score on server and show leaderboard» 13](#_Toc358893275)

[2. Анализ 15](#_Toc358893276)

[2.1. Классы анализа 15](#_Toc358893277)

[2.2. Диаграмма активности 18](#_Toc358893278)

[2.3. Диаграмма состояний 19](#_Toc358893279)

[2.4. Диаграммы последовательностей 20](#_Toc358893280)

[3. Проектирование 22](#_Toc358893281)

[3.1. Проектные классы 22](#_Toc358893282)

[3.2. Конечные автоматы 22](#_Toc358893283)

[4. Реализация 24](#_Toc358893284)

[4.1. Тестирование 24](#_Toc358893285)

[4.2. Развертывание 25](#_Toc358893286)

[4.3. Работа приложения 27](#_Toc358893287)

[4.4. Фрагменты кода приложения 29](#_Toc358893288)

[5. Руководство пользователя 32](#_Toc358893289)

[5.1. Назначение программы 32](#_Toc358893290)

[5.2. Условия выполнения программы 32](#_Toc358893291)

[5.3. Выполнение программы 32](#_Toc358893292)

[Заключение 34](#_Toc358893293)

[Список используемой литературы 35](#_Toc358893294)

Введение

С появлением Windows 8 разработчики и пользователи получили новую редакцию операционной системы под названием Windows RT (Windows RunTime) для планшетных и других компьютеров на базе ARM-процессоров. Windows RT не совместима с существующими Windows-приложениями и поставляется с новыми планшетами Microsoft Surface в качестве предустановленной системы. На Windows RT могут быть использованы только программы, написанные с использованием Windows Runtime (интерфейс программирования приложений Modern), за исключением Microsoft Office 2013 и настольной версии Internet Explorer 10. Разработчики не смогут создавать приложения с помощью интерфейса Win32 API для запуска на Windows RT.

В качестве задания для курсовой работы была взята разработка игрового приложения для Windows 8. Приложение должно быть размещено в Windows Store и быть доступно для загрузки любому пользователю.

1. Требования к программе
   1. Техническое задание

### Введение

Основываясь на интересных историях приложения Gameforcats для iPad, было решено написать такое же приложение для Windows Store для запуска на ноутбуках-трансформерах и планшетах на Windows 8. Приложение обеспечивает веселое времяпрепровождение для владельцев домашних животных. Пользователь может авторизовываться через Facebook, загружать рекорды на сервер и смотреть, кто из его друзей набрал больше баллов в игре, чем он. Рекорды пользователя хранятся на сервере, и можно продолжить игру в любое время и на любом устройстве, авторизовываясь в приложении через Facebook.

Программное обеспечение представляет собой Windows Store приложение, преднозначенное для распространения через Магазин Windows (Windows Store) для всех устройств, работающих на Windows 8.

В качестве базы данных используется SQL Server в Windows Azure, в качестве веб-сервера для Windows Communication Foundation используется Free WebSite в Windows Azure.

Наименование: «Game for cats».

Приложение разработано для работы на устройствах с Windows 8 или Windows RT.

Приложение ориентировано на широкий круг пользователей-владельцев домашних животных.

### Основание для разработки

Разработка ведётся на основании задания на курсовую работу по дисциплине «Объектно-ориентированное программирование».

Документ утверждён ТПУ, каф. ОСУ в феврале 2013 г.

Тема разработки: Создание приложения для Windows Store и интеграцией с социальными сетями с использованием объектно–ориентированного стиля программирования.

### Назначение разработки

Реализуемое программное обеспечение предназначено для владельцев домашних животных в развлекательных целях.

### Требования к программе или программному изделию

* + - 1. **Требования к функциональным характеристикам**

Приложение должно удовлетворять следующим требованиям к функциональным характеристикам:

1. авторизация через Facebook;
2. хранение токена доступа в RoamingSettings для отсутствия необходимости авторизовываться каждый раз при запуске приложения;
3. загрузка результатов игры на сервер и загрузка результатов с сервера в приложение;
4. просмотр друзей на Facebook, играющих в приложение, и вывод списка друзей с максимальными достижениями в игре;
5. возможность игры без авторизации на Facebook;
6. увеличение числа очков при нажатии пользователем на объект «мышь».
   * + 1. **Требования к надежности**

Устойчивое функционирование программы обеспечивается надежностью технических средств.

Контроль входной информации осуществляется путем проверки введенных пользователем данных, и в случае, если данные не соответствуют типу, сообщить пользователю об этом.

* + - 1. **Условия эксплуатации**

Климатические условия эксплуатации, при которых должны обеспечиваться заданные характеристики, должны удовлетворять требованиям, предъявляемым к техническим средствам в части условий их эксплуатации.

Программа не требует проведения каких-либо видов обслуживания.

Конечный пользователь программы должен обладать практическими навыками работы с графическим пользовательским интерфейсом операционной системы.

* + - 1. **Требования к составу и параметрам технических средств**

В состав технических средств должен входить IBM-совместимый персональный компьютер (ПЭВМ), включающий в себя процессор с тактовой частотой не менее 200 МГц, оперативную память объёмом не менее 64 Мб, свободное пространство на жестком диске объёмом не менее 10 Мб, монитор с разрешением не менее 1024 на 768 точек, клавиатуру и мышь, совместимые с используемым персональным компьютером. Персональный компьютер должен работать под управлением Windows 8, Windows 8 Pro, либо должен использоваться планшет Microsoft Surface.

* + - 1. **Требование к информационной и программной совместимости**

Исходные коды программы должны быть реализованы на языке C# на платформе .Net Framework 4.5. В качестве интегрированной среды разработки программы должна быть использована среда Microsoft Visual Studio 2012.

Системные программные средства, используемые программой, должны быть представлены лицензионной локализованной версией операционной системы Microsoft® Windows® 8.

* + - 1. **Требования к маркировке и упаковке**

Требования к маркировке и упаковке не предъявляются.

* + - 1. **Требования к транспортированию и хранению**

Требования к транспортированию и хранению не предъявляются.

* + - 1. **Требования к программной документации**

В состав программной документации должны быть включены:

* техническое задание;
* диаграммы UML;
* результаты модельного, интеграционного и системного тестирования;
* руководство пользователя.

### Технико-экономические показатели

Ориентировочная экономическая эффективность и предполагаемая годовая потребность не рассчитываются. Экономические преимущества разработки по сравнению с аналогами не рассматриваются.

### Стадии и этапы разработки

Работа выполняется в пять этапов:

* разработка технического задания;
* анализ;
* проектирование;
* реализация программы;
* документирование.

На этапе программной реализации должна быть выполнена работа по программированию (кодированию) и отладке программы.

На этапе документирования должна быть выполнена разработка программных документов.

### Порядок контроля и приемки

В результате выполнения работы должны быть предоставлены следующие документы:

1. анализ предметной области;
2. техническое задание;
3. проектирование программного продукта;
4. описание программы;
5. руководство пользователя;
6. исходный код программы;
7. отчет о проделанной работе.

Все документы кроме отчета могут быть в электронном виде. Отчет должен быть распечатан.

* 1. Варианты использования

На рис. 1 представлена диаграмма вариантов использования приложения.

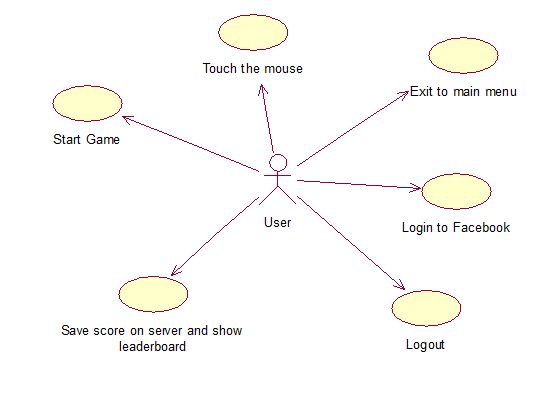
****

Рис. 1. Диаграмма вариантов использования

### Вариант использования «Start Game»

Главные актеры: пользователь.

Второстепенные актеры: отсутствуют.

Предусловия: пользователь находится в главном меню.

Основной поток:

1. Пользователь нажимает на кнопку «Play».
2. Пользователю показывается страница с игровым процессом.

Постусловия: отобразилась страница с игровым процессом.

Альтернативные потоки: отсутствуют.

### Вариант использования «Touch the mouse»

Главные актеры: пользователь.

Второстепенные актеры: отсутствуют.

Предусловия: приложение находится на странице с игровым процессом.

Основной поток:

1. Пользователь касается объекта «мышь» в процессе игры.
2. Система увеличивает игровой счет.
3. Система обновляет значение игрового счета на странице.
4. Система включает звук «попадания на мышь».
5. Система перемещает «мышь» в случайное место на игровом поле.

Постусловия: игровой счет пользователя увеличен.

Альтернативные потоки: отсутствуют.

### Вариант использования «Exit to main menu»

Главные актеры: пользователь.

Второстепенные актеры: отсутствуют.

Предусловия: пользователь находится на странице с игровым процессом.

Основной поток:

1. Счетчик нажатий находится в положении ноль.
2. Пока счетчик нажатий меньше 5.
   1. Если пользователь нажал на кнопку «Выход в меню», увеличить счетчик нажатий на 1.
   2. Если в течении 1000 мс нажатия на кнопку «Выход в меню» не происходит, обнулить счетчик нажатий.
3. Система открывает страницу «Главное меню».

Постусловия: пользователь переходит в главное меню.

Альтернативные потоки: отсутствуют.

### Вариант использования «Login to Facebook»

Главные актеры: пользователь.

Второстепенные актеры: отсутствуют.

Предусловия: пользователь находится в главном меню, пользователь не авторизован в Facebook, доступно подключение к Интернет.

Основной поток:

1. Пользователь нажимает кнопку «Login to Facebook».
2. Система показывает форму для ввода логина и пароля на Facebook.
3. Пользователь правильно вводит свои логин и пароль от аккаунта на Facebook.
4. Система авторизует пользователя на Facebook и сохраняет токен доступа пользователя к API Facebook.
5. Система отправляет счет пользователя на сервер Game for cats и сравнивает его со счетом на сервере. Если счет на сервере больше, чем в приложении, загружается счет с сервера.
6. Система выполняет поиск друзей пользователя на Facebook (тех, кто как и пользователь играет в Game for cats) и ранжирует друзей по степени убывания игрового счета.
7. Система отображает пользователю «Таблицу рекордов его друзей».

Постусловия:

1. Система получила токен доступа к API Facebook пользователя.
2. Система сохранила значение игрового счета пользователя на сервере.
3. Пользователь видит игровые счета в приложении его друзей на Facebook.

Альтернативные потоки:

1. Неправильно введены логин/пароль.

**1.2.4.1. Альтернативный поток варианта использования «Login to Facebook»: «Неправильно введены логин/пароль»**

Краткое описание: система уведомляет пользователя, что введенные им данные для входа на Facebook (логин и пароль).

Главные актеры: пользователь.

Второстепенные актеры: отсутствуют.

Предусловия:

1. Пользователь не авторизован в Facebook.
2. Доступно подключение к Интернет.
3. Пользователю отображается форма ввода логина и пароля для авторизации на Facebook.

Альтернативные потоки:

1. Альтернативный поток начинается с шага 2 основного потока.
2. Система не получит токен доступа к API Facebook.
3. Система не отправит счет пользователя на сервер.
4. Система не загрузит список достижений друзей пользователя на Facebook.

Постусловия: нет.

### 1.2.5. Вариант использования «Logout»

Главные актеры: пользователь.

Второстепенные актеры: отсутствуют.

Предусловия:

1. Пользователь находится в главном меню.
2. Пользователь авторизован в Facebook.

Основной поток:

1. Пользователь нажимает кнопку «Logout».
2. Система удаляет токен доступа к Facebook.
3. Система переходит в режим локального счета. Счет теперь не синхронизируется с сервером.

Постусловия: пользователь не авторизован на Facebook. Счет пользователя не синхронизируется с сервером.

Альтернативные потоки: отсутствуют.

### 1.2.6. Вариант использования «Save score on server and show leaderboard»

Главные актеры: пользователь.

Второстепенные актеры: отсутствуют.

Предусловия: доступно Интернет-соединение.

Основной поток: пользователь запускает приложение или выходит в главное меню.

Основной поток:

1. Счет пользователя отправляется на сервер и сравнивается с сохраненным счетом. Если счет на сервере больше, чем в приложении, загружается счет с сервера.
2. Система происходит поиск друзей пользователя на Facebook, кто играет в это приложение, и ранжирует по степени убывания игрового счета.
3. Система отображает таблицу рекордов пользователю.

Постусловия: счет сохранен на сервере.

Альтернативные потоки: отсутствуют.

1. Анализ
   1. Классы анализа

Целью этапа анализа является создание аналитической модели, которая состоит из классов анализа и прецендентов, которые описывают взаимодействие между экземплярами классов анализа для выполнения определенного варианта использования [2].

Классы анализа представляют собой абстракции предметной области, не относящиеся к области решения.

На рис. 2 представлены классы анализа для разрабатываемого приложения Windows 8.

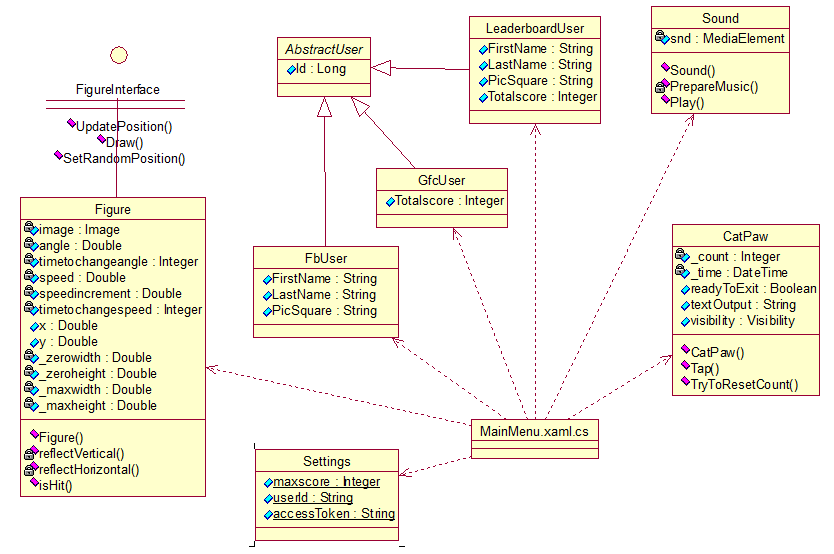


Рис. 2. Классы анализа

Интерфейс *FigureInterface* используется для наследования от него класса *Figure*. Это сделано для следующих версий программы, в которых типов фигур может быть несколько.

Методы интерфейса *FigureInterface*:

* UpdatePosition() – вычисляет новые координаты местоположения экземпляра класса, наследуемого от интерфейса *Figure*, на форме;
* Draw() – перерисовка объекта;
* SetRandomPosition() – установка объекта в случайную позицию на форме.

Класс *Figure* реализует интерфейс *FigureInterface*.

Поля класса *Figure*:

* Image – изображение, которое рисуется на форме;
* angle – угол направления движения;
* Timetochangeangle – время, по истечении которого угол направления движения изменится;
* Speed – скорость перемещения фигуры по форме;
* Speedincrement – ускорение фигуры;
* Timetochangespeed – время, по истечении которого скорость объекта изменится;
* X – координата по оси абсцисс;
* Y – координата по оси ординат;
* \_zerowidth – абсолютная координата левой стороны формы;
* \_zerohight – абсолютная координата верхней стороны формы;
* \_maxwidth – абсолютная координата правой стороны формы;
* \_maxheight – абсолютная координата нижней стороны формы;

Методы класса *Figure*:

* Figure() – конструктор класса;
* reflectVertical() – отразить по вертикали;
* reflectHorisontal() – отразить по горизонтали;
* isHit() – пользователь нажал на него.

Абстрактный класс *AbstractUser* предназначен для наследования от него классов *FbUser*, *GfcUser* и *LeaderboardUser*.

Поля класса *AbstractUser*:

* Id – ID пользователя в Facebook.

Класс *GfcUser* предназначен для хранения информации об игроках приложения.

Поля класса *GfcUser*:

* Totalscore – показывает общий счет конкретного пользователя.

Класс *LeaderboardUser* предназначен для построения таблицы рекордов.

Поля класса *LeaderboardUser*:

* FirstName – имя пользователя в Facebook;
* LastName – фамилия пользователя в Facebook;
* PicSquare – фотография пользователя;
* Totalscore – счет пользователя.

Класс *Sound* предназначен для воспроизведения звукового файла при нажатии на «мышь».

Поля класса *Sound*:

* Snd – звуковой файл на жестком диске.

Методы класса *Sound*:

* Sound() - конструктор;
* PrepareMusic() – открывает файл;
* Play() – воспроизводит звуковой файл.

Класс *CatPaw* предназначен для реализации кнопки выхода в главное меню

Поля класса *CatPaw*:

* \_count;
* \_time;
* readytoExit;
* textOutput;
* visibility.

Методы класса *CatPaw*:

* CatPaw();
* Tap();
* TryToResetCount().

Класс *Settings* предназначен для локального хранения настроек приложения.

Поля класса *Settings*:

* Maxscore – хранение (доступ и запись) значения максимального счета пользователя;
* userId – хранение (доступ и запись) значения ID пользователя в Facebook;
* accessToken – хранение (доступ и запись) значения токена доступа к API Facebook.

* 1. Диаграмма активности

На рис. 3 предствлена обобщенная диаграмма активности приложения.

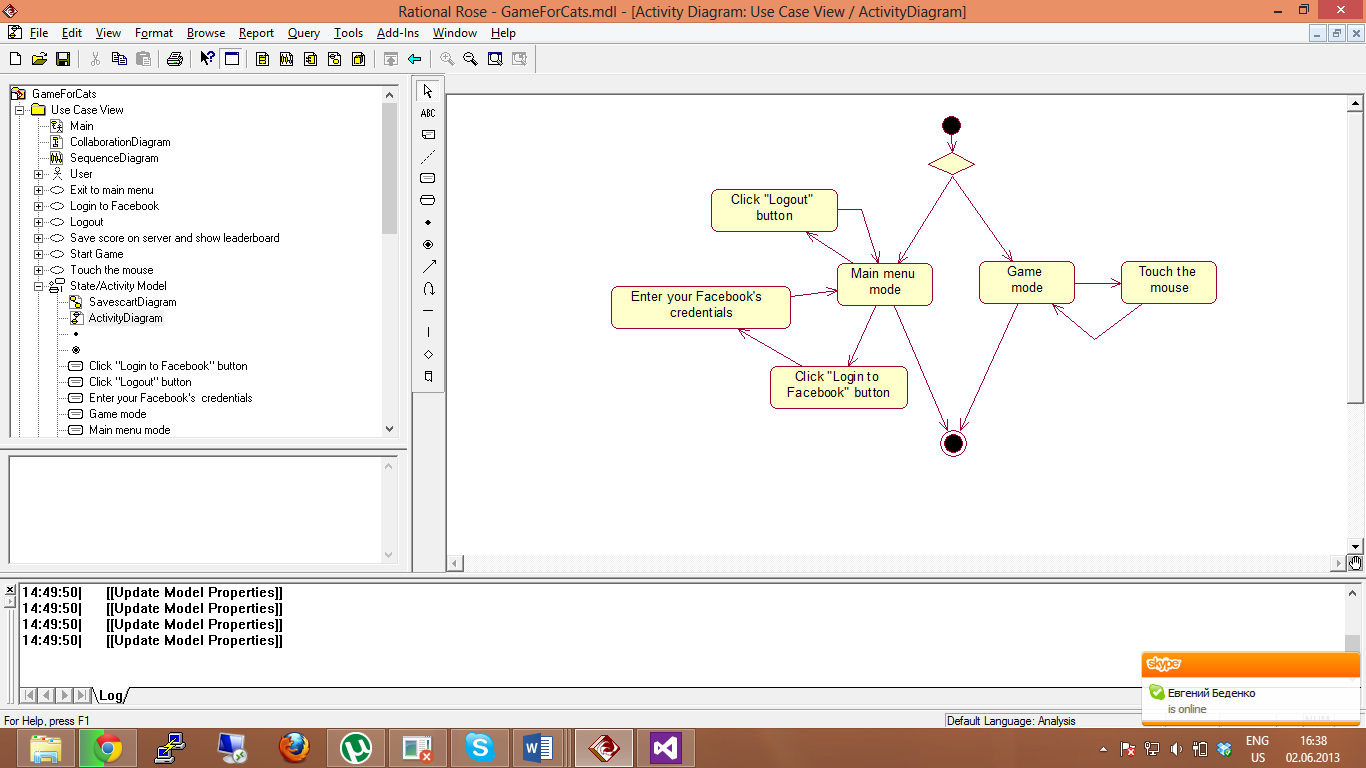


Рис. 3. Обобщенная диаграмма активности

Как видно из диаграммы, на странице «Main menu mode» пользователь может авторизоваться в Facebook, выйти из аккаунта Facebook, поделиться текущим счетом и начать игру. На странице «Game mode» пользователь может касаться мыши и выйти в главное меню.

* 1. Диаграмма состояний

На рис. 4 показана диаграмма состояний приложения Game for cats.

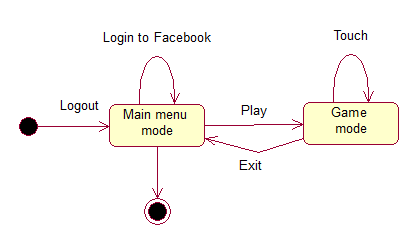


Рис. 4. Диаграмма состояний приложения Game for cats

Как видно из диаграммы, в приложении есть 2 состояния: Main menu mode и Game mode. Первое состояние – это главное меню, страница, которую пользователь видит при запуске приложения или после нажатия кнопки «Выход в главное меню». На ней отображается текущий счет пользователя, его имя и фамилия в Facebook, фотография и кнопка «Рассказать друзьям о моем счете».

* 1. Диаграммы последовательностей

На рис. 5 показана обобщенная диаграмма последовательностей для приложения, WCF-сервиса и базы данных SQL.

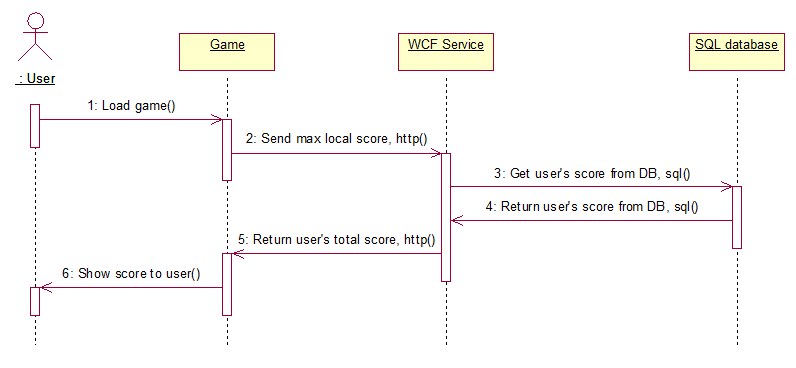


Рис. 5. Диаграмма последовательности, отражающая цикл существования объектов в приложении

На этой диаграмме показано, как взаимодействуют между собой приложение, Windows Communication Foundation Service (SCF Service), и база данных SQL.

При загрузке приложения или перехода в главное меню проверяется наличие токена доступа к API Facebook и соединения Интернет. Затем вызывается функция Sendmaxlocalscore() WCF-сервиса, в параметрах которой передается текущий локальный счет и ID пользователя. WFC-сервис проверяет наличие пользователя в базе данных SQL в облаке, и если пользователь найден, то сравнивает счет и сохраняет наибольший. После чего наибольшее значение счета возвращается функцией в приложение.

На диаграмме последовательности рис. 6 показан процесс авторизации пользователя в социальной сети Facebook, сохранение токена доступа к API Facebook в локальном хранилище приложения и запрос на сервер за сохраненным счетом пользователя.

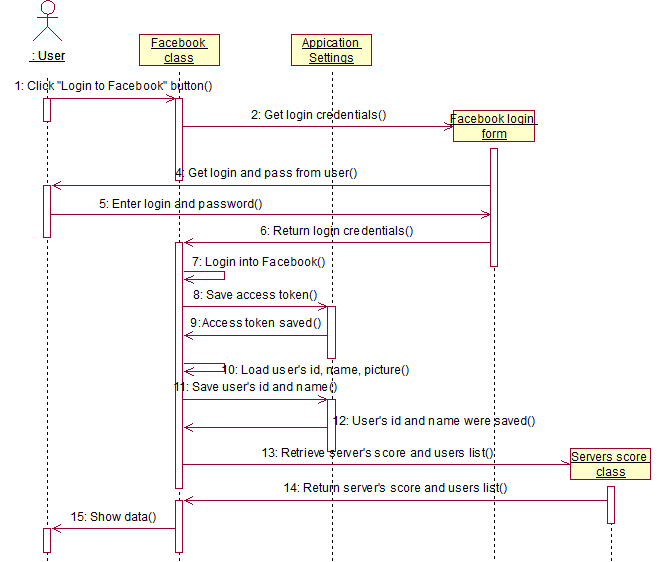


Рис. 6. Диаграмма последовательности, отражающая процесс авторизации приложения через Facebook, получения пользовательских данных и списка друзей

При нажатии на кнопку «Login to Facebook» в главном меню пользователю отображается форма входа в службу Facebook. После введения логина и пароля пользователя на Facebook система пытается авторизоваться, и если процесс завершается успешно, то система сохраняет токен доступа к API Facebook в настройках. Зная токен доступа, система может в дальнейшем обращаться к API Facebook без ввода логина и пароля пользователя. После этого система загружает имя пользователя, его фамилию и фотографию профиля из социальной сети Facebook, и обращается к серверу Game for cats за сохраненными настройками пользователя.

1. Проектирование

На этапе проектирования на основе ранее разработанной аналитической модели создается новая модель, которая помимо предметной областидолжна также охватывать последующую реализацию. Проектная модель является более детализированной и должна учитывать конкретные технические решения.

В следующем разделе приведены проектные классы, которые были выявлены в ходе уточнения аналитической модели, описанной ранее.

* 1. Проектные классы

Основное отличие проектных классов от аналитических состоит в том, что их описания должны быть максимально полными, поскольку на их основе будет проходить этап реализации.

Поскольку диаграмма аналитических классов на рис. 2 содержит полное описание классов, достаточное для реализации ПО, эту диаграмму можно считать диаграммой проектных классов.

* 1. Конечные автоматы

На последнем этапе проектирования перед непосредственным построением нередко для моделирования динамического аспекта поведения классов применяются конечные автоматы. Спецификация UML 2 выделяет два типа конечных автоматов – это поведенческие и протокольные автоматы.

Первый тип использует состояния, переходы и события, которые определяют поведение системы. В свою очередь, состояния протокольных автоматов не могут определять поведение [2].

Каждый объект системы, обладающий определенным поведением, может находиться в определенных состояниях, переходить из состояния в состояние, совершая определенные действия в процессе реализации сценария поведения объекта. Диаграммы состояний позволяют графически отразить поведения системы с точки зрения теории конечных автоматов [5].

На рис. 4 показана диаграмма состояний для приложения.

После запуска приложение находится в режиме «Main menu mode». Пользователь может авторизоваться в Facebook, выйти из Facebook, поделиться своим счетом и начать новую игру. При старте игры приложение переходит в режим «Game mode». В этом режиме пользователь может коснуться мыши или выйти в главное меню.

1. Реализация
   1. Тестирование

В этом разделе приведены тесты для приложения:

* 1. нажатие на фигуру;

[TestMethod]

public void TestMethodTapFigure()

{

Figure f = new Figure(1366,768);

f.x = 500;

f.y = 500;

if (!f.isHit(new Point(510, 520)))

{

Assert.Fail("Событие Tap было совершено в центр фигуры, но не было распознано как нажатие на фигуру");

}

}

* 1. нажатие на пустую область, где нет фигуры;

[TestMethod]

public void TestMethodTapNotFigure()

{

Figure f = new Figure(1366, 768);

f.x = 500;

f.y = 500;

if (f.isHit(new Point(0, 10)))

{

Assert.Fail("Событие Tap было совершено в место, отличное от месторасположения фигуры, но было распознано как нажатие на фигуру");

}

}

* 1. проверка на перемещение фигуры;

[TestMethod]

public void TestMethodCheckMovement()

{

//создаем фигуру

Figure f = new Figure(1366, 768);

//устанавливаем ее позицию на экране

f.x = 50;

f.y = 50;

//вычисляем следующие координаты сами

double \_x = 50;

double \_y = 50;

double \_speed = 1;

double \_angle = 1;

\_x += \_speed \* Math.Cos(\_angle) / 2;

\_y += \_speed \* Math.Sin(\_angle) / 2;

//вызываем метод, который вычислит их в классе Figure

f.UpdatePosition();

//сверяем

Assert.AreEqual(new Point(\_x,\_y), new Point(f.x, f.y));

}

* 1. проверка условия выхода в главное меню класса CatPaw;

[TestMethod]

public void TestMethodCatPawReadyToExit()

{

CatPaw catPaw = new CatPaw();

for (int i=0; i<6;i++)

catPaw.Tap();

Assert.IsTrue(catPaw.readyToExit);

}

* 1. проверка сообщения textOutput класса CatPaw.

[TestMethod]

public void TestMethodCatPawTextOutput()

{

CatPaw catPaw = new CatPaw();

for (int i = 0; i < 1; i++)

catPaw.Tap();

Assert.IsTrue(catPaw.textOutput.Equals("tap a paw 4 times to pause"));

}

* 1. Развертывание

Приложение было размещено в Магазине приложений Windows Store.

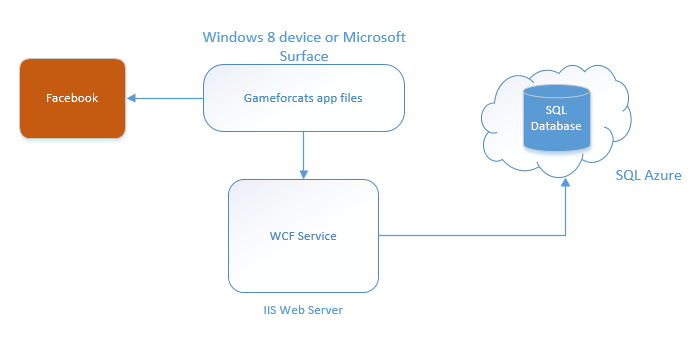


Рис. 7. Диаграмма развертывания приложения Game for cats

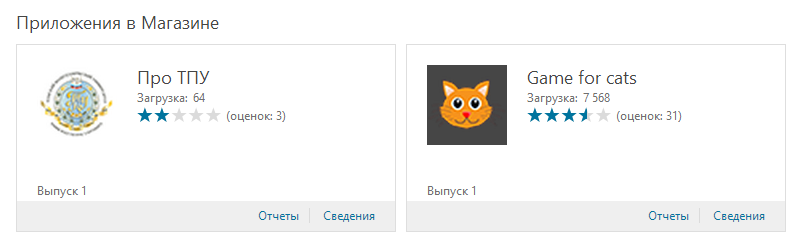


Рис. 8. Приложение в инфопанели Центра разработки приложений Windows

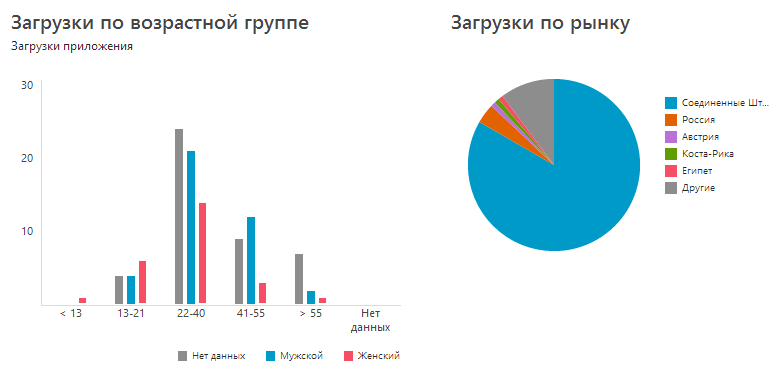


Рис. 9. Статистика загрузок приложения по возрастной группе и по рынкам

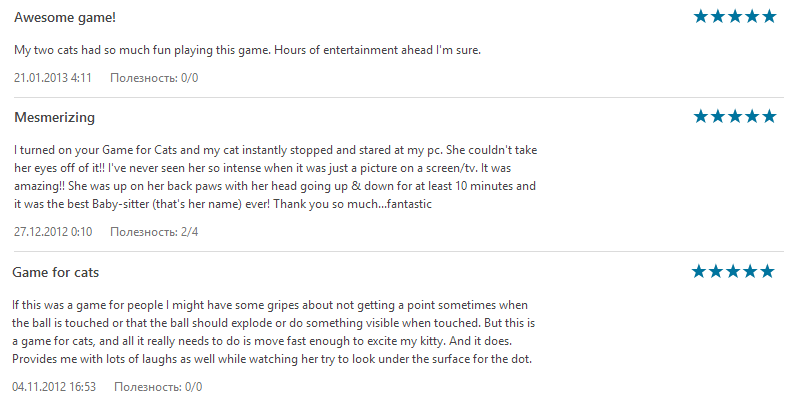


Рис. 10. Некоторые отзывы об использовании приложения от пользователей в США, Канаде и России

* 1. Работа приложения

На рис. 11, рис. 12, рис. 13, рис. 14 показаны скриншоты приложения в различных режимах.



Рис. 11. Главное меню. Пользователь не авторизован в приложении через Facebook

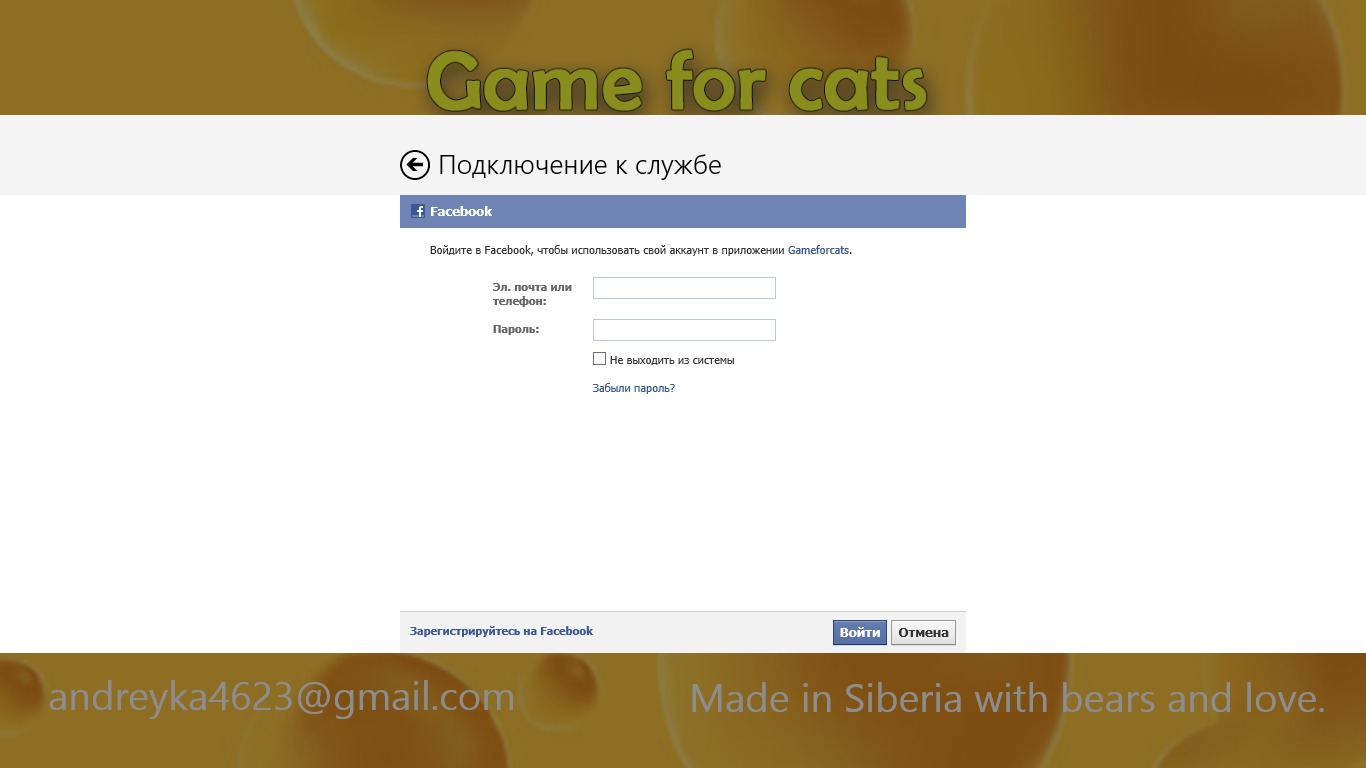


Рис. 12. Авторизация в приложении с использованием Facebook



Рис. 13. Главное меню. Пользователь авторизован в приложении через Facebook



Рис. 14. Режим игры

* 1. Фрагменты кода приложения

В этом разделе опуликован исходный код некоторых частей приложения.

1. Хранение токена доступа к Facebook;

public static String accessToken

{

get

{

var settings = ApplicationData.Current.RoamingSettings;

String \_accessToken = "";

if (settings.Values.ContainsKey("access\_token"))

{

\_accessToken = (String)settings.Values["access\_token"];

}

return \_accessToken;

}

set

{

ApplicationData.Current.RoamingSettings.Values["access\_token"] = value;

}

}

1. Воспроизведение музыки при нажатии на «Мышь»;

private async void PrepareMusic()

{

snd = new MediaElement();

StorageFolder folder = await Package.Current.InstalledLocation.GetFolderAsync("Assets");

StorageFile file = await folder.GetFileAsync("mouse.wav");

var stream = await file.OpenAsync(FileAccessMode.Read);

snd.SetSource(stream, file.ContentType);

}

1. Код класса FbUser;

class FbUser : AbstractUser

{

public String FirstName { set; get; }

public String LastName { set; get; }

public String PicSquare { set; get; }

}

1. Код метода GetAllUser() WCF-сервиса;

public String GetAllUsers()

{

SqlConnectionStringBuilder connString1Builder = new SqlConnectionStringBuilder();

connString1Builder.DataSource = "ebpwlp9r2o.database.windows.net";

connString1Builder.InitialCatalog = "gameforcats";

connString1Builder.PersistSecurityInfo = true;

connString1Builder.UserID = "gameforcats@ebpwlp9r2o";

connString1Builder.Password = "Cpes465443";

connString1Builder.ConnectTimeout = 30;

String result = "";

try

{

SqlConnection connection = new SqlConnection(connString1Builder.ToString());

connection.Open();

SqlCommand command = connection.CreateCommand();

command.CommandText = "SELECT Facebookid, Totalscore FROM dbo.users ORDER BY Totalscore DESC;";

using (SqlDataReader Reader = command.ExecuteReader())

{

while (Reader.Read())

{

result += Reader.GetValue(0).ToString() + ":::" + Reader.GetValue(1).ToString() + ":::";

}

}

}

catch (SqlException e)

{

result = "error";

}

return result;

}

1. Код метода «Нажатие на мышь».

private async void i\_Tapped(object sender, TappedRoutedEventArgs e)

{

//кот поймал мышку

if (mousepointer.isHit(e.GetPosition(Grid1)))

{

sound.Play();

score++;

Settings.maxscore = Settings.maxscore + 1;

txtScore.Text = Convert.ToString(score);

mousepointer.SetRandomPosition();

imgPointer.Visibility = Visibility.Collapsed;

var d = new DispatcherTimer();

TimeSpan t = new TimeSpan(0, 0, 0, 0, 100);

d.Interval = t;

d.Start();

d.Tick += (sender2, o) =>

{

imgPointer.Visibility = Visibility.Visible;

d.Stop();

};

}

}

1. Руководство пользователя
   1. Назначение программы

Программа предназначена широкому кругу пользователей – владельцев домашних животных.

* 1. Условия выполнения программы

В состав технических средств должен входить IBM-совместимый персональный компьютер (ПЭВМ), включающий в себя процессор с тактовой частотой не менее 200 МГц, оперативную память объёмом не менее 64 Мб, свободное пространство на жестком диске объёмом не менее 10 Мб, монитор с разрешением не менее 1024 на 768 точек, клавиатуру и мышь, совместимые с используемым персональным компьютером. Персональный компьютер должен работать под управлением Windows 8, Windows 8 Pro, либо должен использоваться планшет Microsoft Surface.

* 1. Выполнение программы

При запуске программы пользователь видит на экране форму (рис. 11), где находятся 2 кнопки: «Play» и «Login with Facebook», а также поле с отображением набранного количества очков «Your score».

Для начала игры следует нажать кнопку «Play». Откроется страница с игровым процессом (рис. 14). В процессе игры следует нажать на постоянно перемещающийся указатель «Мыши». Если происходит нажатие на мышь, происходит увеличение счета.

Цель игры – набрать больше очков, чем у ваших друзей.

Для выхода в главное меню необходимо нажать 5 раз подряд на кнопку «Выход в главное меню» с частотой нажатия не реже, чем раз в 1 секунду.

Для авторизации в Facebook требуется нажать кнопку «Login with Facebook», после чего откроется страница для ввода логина и пароля на Facebook (рис. 12). Если авторизация в Facebook была выполнена успешно, главное меню изменится так, как показано на рис. 13.

Заключение

В результате проделанной работы было реализовано программное обеспечение «WinRT приложение для Windows 8» под названием «Game for cats». Данное приложение может быть установлено на компьютер и планшетное устройство пользователя через Магазин приложений Windows (Windows Store).

Данное приложение позволяет:

1. авторизовывать пользователя через Facebook;
2. хранить счет пользователя на сервере;
3. отображать “Таблицу рекордов” – список друзей, играющих в приложение, отсортированный по убыванию;
4. быть установленным централизованным образом на любом устройсте с WinRT на борту;
5. публиковать информацию о достижениях на Facebook.

Данные манипуляции удовлетворяют «Техническому заданию» и, следовательно, ПО готово для конечного использования. ПО было реализовано в рамках курсового проекта по дисциплине «Технология разработки программного обеспечения» и готово для проверки преподавателем.

Список используемой литературы

1. ГОСТ 19.106-78. Требования к программным документам, выполненным печатным способом;
2. ГОСТ 19.201-78. Техническое задание. Требования к содержанию и оформлению;
3. ГОСТ 19.402-78. Описание программы;
4. ГОСТ 19.505-79. Руководство оператора. Требования к содержанию и оформлению;
5. Вендров А. В. Практикум по проектированию программного обеспечения экономических информационных систем. – М.: Финансы и статистика, 2002;
6. Вендров А. В. Проектирование программного обеспечения экономических информационных систем: Учебник. – М.: Финансы и статистика, 2000. – 352 с.;
7. Орлов С. А. Технологии разработки программного обеспечения. – СПб.: Питер, 2003. – 480 с.;
8. Арлоу Д., Нейштадт И. UML 2 и Унифицированный процесс. Практический объектно–ориентированный анализ и проектирование, 2–е издание. – Пер. с англ. – СПб: Символ–Плюс, 2007. – 624 с., ил.