Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Северо-Осетинский государственный университет

имени Коста Левановича Хетагурова»

**КУРСОВАЯ РАБОТА**

«Приложение для игрового изучения осетинского языка»

Выполнил:

студент 1 курса направления

«Прикладная математика и информатика»

Плиев Давид Таймуразович

Научный руководитель:

старший преподаватель кафедры

прикладной математики и информатики

Макаренко М.Д.

**2023 г.**

**Оглавление**

[**Введение** 3](#_Toc136291679)

[**Глава 1. Теоретическая часть** 5](#_Toc136291680)

[**Глава 2. Практическая часть.** 7](#_Toc136291681)

[1. Инструкция пользователя 7](#_Toc136291682)

[2. Инструкция системного администратора 9](#_Toc136291683)

[3. Инструкция программиста 10](#_Toc136291684)

[**Заключение** 15](#_Toc136291685)

[**Список использованной литературы** 16](#_Toc136291686)

## **Введение**

На сегодняшний день можно найти массу способов изучить язык. Компьютерные технологии открывают новые горизонты в области образования. Поэтому, разработка приложений для игрового изучения языков приобретает все большую актуальность. Игровые приложения позволяют обучающимся взаимодействовать с языком в игровой форме, что существенно повышает эффективность и интерес к процессу изучения.

В данной курсовой работе рассматривается создание приложения для игрового изучения осетинского языка в среде разработки Visual Studio Windows Forms на языке программирования C#. Осетинский язык является важным компонентом культурного наследия осетинского народа и имеет свою уникальную грамматику и лексику. Однако, несмотря на его значимость, существует совсем немного образовательных ресурсов и приложений, специально разработанных для изучения осетинского языка.

Целью данной курсовой работы является реализация приложения, которое предназначено для игрового изучения осетинского языка. Приложение должно предоставлять поле с животными, где пользователь должен поймать каждое животное, чтобы отобразилось и озвучилось соответствующее слово на осетинском языке. Такой подход позволит улучшить запоминание и осознанное усвоение осетинской лексики.

Для достижения этой цели необходимо поставить следующие задачи:

1. Проектирование пользовательского интерфейса: Необходимо разработать привлекательный и интуитивно понятный интерфейс приложения. Пользователь должен иметь возможность взаимодействовать с полем, перемещать персонажа (главного героя) по нему и ловить животных.
2. Реализация игровой механики: Приложение должно обеспечивать возможность перемещения по полю, а также ловли животных. Пользователь должен иметь возможность управлять персонажем с помощью стрелок клавиатуры и поймать животное при достижении. После поимки животного должно появляться соответствующее слово на осетинском языке.
3. Создание словаря: Для реализации функции отображения слов на осетинском языке необходимо интегрировать осетинский словарь, содержащий соответствующие слова и их переводы.
4. Тестирование и отладка: После разработки приложения необходимо провести тестирование, чтобы обнаружить и исправить возможные ошибки и неполадки.
5. Оформление инструкций: В конце работы необходимо составить инструкции, в которых будут описаны все функциональные возможности приложения, принятые решения при разработке, инструкции для пользователя и программиста и возможности дальнейшего улучшения приложения.

Актуальность данной темы обусловлена несколькими факторами. Во-первых, сохранение и изучение языков, таких как осетинский, имеет важное культурное и историческое значение. Приложение для игрового изучения осетинского языка будет способствовать его популяризации среди молодого поколения и поможет сохранить языковое наследие.

Во-вторых, использование компьютерных игр в образовании становится все более распространенным подходом, который с успехом применяется для обучения различных задач. Создание игрового приложения для изучения осетинского языка поможет привлечь и заинтересовать новых обучающихся, особенно тех, кто предпочитает интерактивные формы обучения.

Так, разработка приложения для игрового изучения осетинского языка на платформе C# является актуальной задачей, которая вносит вклад в сохранение и популяризацию осетинского языка, а также предоставляет новые возможности для эффективного обучения языку с помощью имеющихся технологий.

## **Глава 1. Теоретическая часть**

Код представляет собой реализацию приложения для игрового изучения осетинского языка, в котором пользователь должен поймать животных для того, чтобы отобразились слова на осетинском языке.

Требования к функциональным характеристикам программы:

1. Пользовательский интерфейс: Приложение должно иметь привлекательный и интуитивно понятный интерфейс, позволяющий пользователю легко взаимодействовать с формой и ловить животных.
2. Генерация животных: При каждом запуске приложения должно генерироваться ровно 10 животных на поле. Животные должны быть расположены в случайно сгенерированных местах на поле, не выходить за его пределы.
3. Отображение слов на осетинском: После пойманного животного должно появляться соответствующее слово на осетинском языке. Для этого используется словарь, где каждому животному соответствует его название на осетинском языке.

Основные этапы реализации программы:

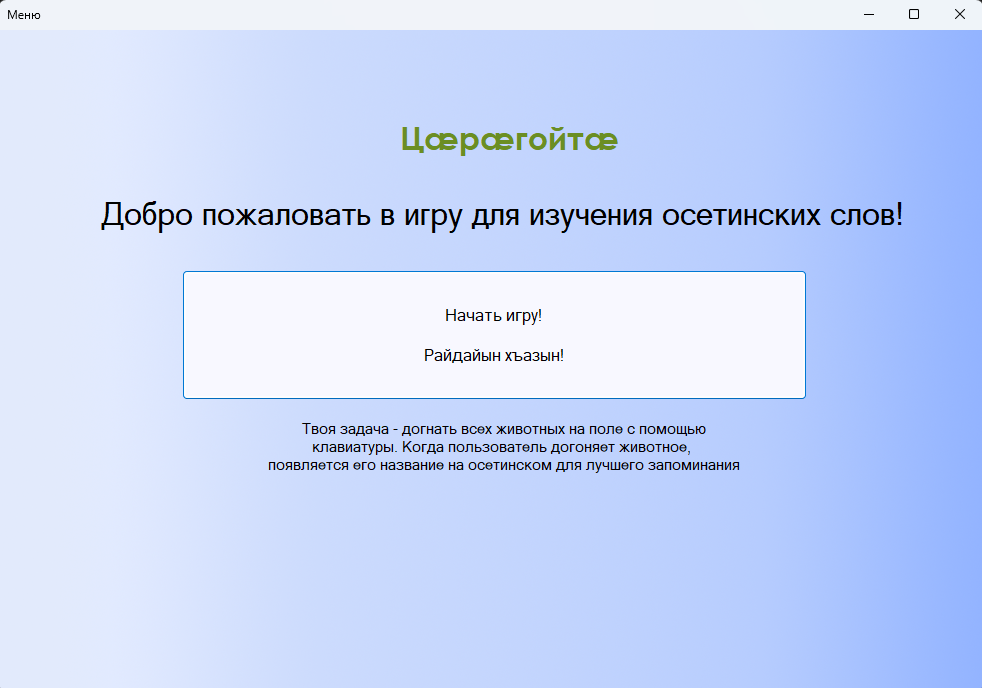
1. Проектирование пользовательского интерфейса: Был разработан лаконичный и интуитивно понятный интерфейс приложения. Пользователь может перемещаться по полю и ловить животных с помощью главного персонажа.
2. Генерация животных: Приложение генерирует ровно 10 компонентов формы PictureBox с изображениями животных. Каждое животное будет получать случайные координаты с помощью использования класса Random.
3. Отображение слов на осетинском: После поимки животного в нижней части экрана появляется надпись с соответствующим словом на осетинском языке. Для этого используется словарь, где каждому животному соответствует его название на осетинском языке.
4. Обработка пользовательских действий: Приложение реагирует на нажатия клавиш пользователя, позволяя главному персонажу перемещаться по полю. Проверяется столкновение главного персонажа с животными, после чего животное исчезает с поля, так как было поймано.
5. Создан таймер для реализации движения животных по форме.
6. Анимация и визуальные эффекты: Реализована анимация движения главного персонажа с помощью использования изображения формата gif. Реализована анимация "удара" при поимке животного. Отредактирован специальный кадр из gif изображения, добавлен «эффект удара». При сближении главного героя и животного меняется изображение первого. Также реализован таймер, который удаляет надпись с осетинским словом после некоторого времени.
7. Завершение игры: При поимке всех животных появляется сообщение о том, что все животные пойманы.

## **Глава 2. Практическая часть.**

### 1. Инструкция пользователя

Для работы программы необходимо запустить приложение расширения .exe с названием «Цӕрӕгойтӕ». На экране появится основное меню.

Далее, если вы хотите начать игру, необходимо нажать на кнопку слева с надписью «Начать игру!», под которой так же располагается краткое описание, инструкция игры.



После начала игры откроется новое окно, в котором можно увидеть поле, на котором двигаются животные.

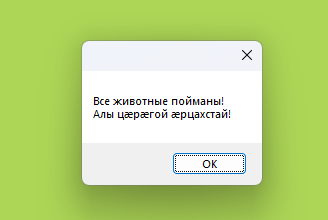


В верхней левой части формы располагается главный герой, которым можно управлять. Для перемещения главного героя по полю необходимо использовать клавиши стрелок на клавиатуре.

Задача – догнать каждого животного. После столкновения главного героя с животным появится его название на осетинском языке. Также воспроизведется соответствующее осетинское слово, что позволит легче запомнить его.



При поимке всех животных появится сообщение о том, что игра успешно завершена и все животные пойманы.



### 2. Инструкция системного администратора

Для работы программы требуются определенные программные и технические средства. Ниже приведены требования к системе для корректной работы программы:

Требования к программным средствам:

* Операционная система: Windows 7 и выше.
* .NET Framework: Для работы программы необходимо наличие установленной версии .NET Framework 4.7.2 или выше.

Требования к техническим средствам:

* ОЗУ: Рекомендуется не менее 2 ГБ оперативной памяти.
* Место на диске: Для установки программы и хранения необходимых ресурсов требуется не менее 100 Мб места на физическом носителе.

Дополнительные требования:

* Экран с разрешением не менее 1920 на 1080 пикселей для полного отображения поля в игровом режиме приложения.
* Клавиатура: Для взаимодействия с программой необходима клавиатура для нажатия клавиш стрелок.
* Колонки или наушники: для воспроизведения звука.

При установке программы системному администратору нужно проверить, чтобы система соответствовала описанным выше требованиям.

### 3. Инструкция программиста

Язык программирования: C#

Среда разработки: Visual Studio 2022  
Интерфейс программирования приложений: Windows Forms

Версия .NET Framework: 4.7.2

Описание кода:

Класс **Pair** с двумя свойствами: *PictureBox*(типа *PictureBox*) и *Direction*(типа *Point*). Создан для того, чтобы облегчить работу с логикой игры в основном классе.

public class Pair

{

public PictureBox PictureBox { get; set; }

public Point Direction { get; set; }

}

В основном классе объявлены следующие объекты:

* **animalPictureBoxes** - список объектов типа *Pair*, хранящих *PictureBox* и *Direction*.

List<Pair> animalPictureBoxes = new List<Pair>();

* Объявлен объект класса Random (используется для генерации случайных чисел).

Random random = new Random();

* **animals** – массив, который содержит названия животных.

string[] animals = { "wolf", "fox", "bear", "hare", "duck", "turtle", "squirrel", "cow", "cat", "dog" };

* **dictOssetianAnimals** - словарь, который связывает названия животных с их переводами на осетинский.

Dictionary<string, string> dictOssetianAnimals = new Dictionary<string, string>()

{

{"animalwolf", "бирæгъ"}, {"animalfox", "рувас"}, {"animalbear", "арс"}, {"animalhare", "тæрхъус"}, {"animalduck", "бабыз"},

{"animalturtle", "уæртджын хæфс"}, {"animalsquirrel", "æхсæрæг"}, {"animalcow", "хъуг"}, {"animalcat", "гæды"}, {"animaldog", "куыдз"}

};

В конструкторе класса вызывается функция CreateAnimalPictureBox для генерации 10 животных на поле.

public FormAnimalsOnField()

{

InitializeComponent();

this.WindowState = FormWindowState.Maximized;

//генерируем ровно 10 PictureBox:

CreateAnimalPictureBox(10);

}

Методы, которые используются в классе

public partial class FormAnimalsOnField : Form

1. Метод **CreateAnimalPictureBox** создает указанное количество PictureBox. Вызывает метод **AddAnimalPictureBox** для каждого животного. Координаты PictureBox генерируется случайным образом с помощью random.

private void CreateAnimalPictureBox(int n)

{

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

{

AddAnimalPictureBox(random.Next(10, 1600), random.Next(300, 800), animals[i]);

}

}

1. Метод **AddAnimalPictureBox** создает новый PictureBox и устанавливает свойства: изображение, размер, положение, тэг и цвет фона. Затем он добавляет PictureBox в список **animalPictureBoxes** и сразу на форму.

private void AddAnimalPictureBox(int x, int y, string animal)

{

PictureBox newAnimal = new PictureBox();

string resourceName = "animal" + animal;

newAnimal.Image = (Image)Resources.ResourceManager.GetObject(resourceName);

newAnimal.Size = new Size(110, 110);

newAnimal.SizeMode = PictureBoxSizeMode.StretchImage;

newAnimal.Location = new Point(x, y);

newAnimal.Tag = resourceName;

newAnimal.BackColor = Color.Transparent;

animalPictureBoxes.Add(new Pair { PictureBox = newAnimal, Direction = new Point(7, 1) });

this.Controls.Add(newAnimal);

}

1. Обработчик события **FormAnimalsOnField\_KeyDown** реагирует на нажатия клавиш пользователем:
   * Изменяет изображение главного персонажа в зависимости от нажатой клавиши для красоты внешнего вида приложения.
   * Проверяет, не выходит ли главный персонаж за пределы формы, прежде чем двигать его на форме.
   * Проверяет, что все животные пойманы
2. Таймер **timerAnimation\_Tick** изменяет изображение главного персонажа на изображение с ударом. После этого таймер останавливается.

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

{

pictureBoxMainCharacter.Image = Resources.mainCharacterForward;

timerAnimation.Stop();

}

1. Таймер **timer\_Tick** обрабатывает движение животных на форме:
   * Перебирает все PictureBox из списка **animalPictureBoxes**.
   * Проверяет столкновение каждого животного с главным персонажем.
   * Если столкновение произошло, удаляет животное с формы и из списка, воспроизводит звук(название животного), отображает Label (название животного на осетинском языке) с помощью метода **createLabel** и запускает таймер анимации удара.
   * Если столкновение не произошло, перемещает животное в указанном направлении. Если животное оказалось у границ формы, его направление движения меняется на противоположное.
2. Метод **createLabel** создает и отображает Label с названием животного на осетинском языке, создает таймер.

private void createLabel(string name)

{

Label label = new Label();

label.Text = $"{dictOssetianAnimals[name]}";

label.Name = name;

label.TextAlign = ContentAlignment.MiddleCenter;

label.Location = new Point((this.ClientSize.Width - label.Width-220) / 2, this.ClientSize.Height - 100);

label.Size = new Size(400, 40);

label.ForeColor = Color.OrangeRed;

label.Font = new Font("Timew New Roman", 28, FontStyle.Regular);

this.Controls.Add(label);

//создаем таймер для label

Timer labelTimer = new Timer();

labelTimer.Interval = 1200;

labelTimer.Tick += LabelTimer\_Tick;

labelTimer.Tag = label;

labelTimer.Start();

}

1. Таймер LabelTImer\_Tick для того, чтобы убрать label по прошествии времени

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

{

Timer timer = (Timer)sender;

timer.Stop();

//проходит время таймера, убираем label

this.Controls.Remove((Label)timer.Tag);

}

1. Обработчик события **global\_FormClosed** нужен для очищения оперативной памяти от приложения(закрытия форм)

private void global\_FormClosed(object sender, EventArgs e)

{

Application.Exit(); }

## **Заключение**

В рамках работы над данным курсовым проектом было разработано приложение для игрового изучения осетинского языка на языке программирования C#, используя визуальные компоненты Windows Forms.

В процессе реализации проекта изучены возможности интегрированной среды разработки Microsoft Visual Studio, освоена работа с различными элементами интерфейса, с системой контроля версий git, получены навыки программирования на соответствующем языке программирования, закреплены все полученные на 1 курсе знания основ программирования, алгоритмов и структур данных.

В проекте реализована основная часть, позволяющая пользователю изучать лексику осетинского языка. В будущем можно расширить функционал программы, добавив возможность выбора уровня сложности. Можно добавить другие режимы обучения, например, тесты для лучшего запоминания лексики, систему результатов для отслеживания прогресса по изучению осетинского языка, улучшить UI/UX приложения, сделав его еще более удобным и приятным в использовании.

## **Список использованной литературы**

* 1. Чарльз Петцольд «Программирование для Microsoft Windows на С#. 2002 г. - с. 22
  2. Нортроп, Т. Основы разработки приложений на платформе Microsoft .NET Framework. Учебный курс Microsoft. Перевод с англ./ Т. Нортроп, Ш. Уилдермьюс, Б. Райан. - М.: «Русская редакция», 2007. - с. 845
  3. Робинсон, С. C# для профессионалов. Том 2. / С. Робинсон, О. Корнес, Д. Глин, Б. Харвей. - М. :Лори, 2003. - 998 с.
  4. Интернет-ресурс metanit: <https://metanit.com/sharp/tutorial/4.9.php>
  5. Интернет-ресурс learn.microsoft.com: https://learn.microsoft.com/ru-ru/dotnet/api/system.collections.generic.dictionary-2?view=net-7.0