

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	4
1 Постановка задачи	7
1.1 Анализ предметной области	7
1.2 Техническое задание на разработку ГОСТ 19.201-78	7
2 Проектирование	13
2.1 Функциональная структура программы	13
2.2 Схема модулей программы	14
2.3 Паспорта модулей	15
3 Программная документация	39
3.1 Руководство оператора ГОСТ 19.505-79	39
4 Тестирование программы	79
4.1 Программа, методика и результаты испытаний ГОСТ 19.301-79	79
4.2 Протокол испытаний	82
ЗАКЛЮЧЕНИЕ	89
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	90
ПРИЛОЖЕНИЕ А	91

ВВЕДЕНИЕ

В современном мире, где мобильные технологии прочно вошли в нашу повседневную жизнь, разработка приложений для смартфонов представляет собой перспективное и востребованное направление. Именно поэтому в качестве дипломного проекта мной была выбрана разработка мобильной игры - области, которая не только соответствует современным тенденциям цифровизации, но и вызывает у меня искренний профессиональный интерес и творческий энтузиазм.

Создание мобильных приложений - это уникальный синтез технических знаний и творческого подхода, позволяющий разрабатывать не просто функциональные, но и по-настоящему увлекательные цифровые продукты, способные приносить реальную пользу миллионам пользователей по всему миру. В условиях постоянно растущей конкуренции на рынке мобильных приложений особенно важно создавать решения, которые сочетают в себе инновационность, удобство использования и образовательную ценность.

Основной целью данной дипломной работы является создание оригинальной мобильной игры под названием «Дождь из слов», которая направлена на развитие и совершенствование когнитивных способностей пользователей. Это приложение призвано помочь людям всех возрастов в увлекательной игровой форме тренировать такие важные навыки, как оперативная память, скорость реакции, концентрация внимания. Особое внимание уделяется разработке простого, интуитивно понятного и в то же время захватывающего игрового процесса, который будет мотивировать пользователей к регулярным тренировкам.

Для реализации проекта выбраны современные и надежные технологии разработки:

- Кроссплатформенный фреймворк .NET MAUI, позволяющий создавать высокопроизводительные приложения для различных мобильных платформ.
- Мощный и гибкий язык программирования C#, обеспечивающий

реализацию сложной игровой логики.

- Профессиональная среда разработки Visual Studio, предоставляющая все необходимые инструменты для эффективной работы.

Для успешной реализации проекта необходимо выполнить следующие ключевые этапы:

- изучение .NET MAUI – глубокий анализ фреймворка, включая его инструменты для разработки интерфейсов и обеспечения кросс-платформенной поддержки;
- совершенствование навыков C# – углубленное освоение языка для эффективной реализации логики приложения и оптимизации его производительности;
- исследование архитектурных подходов – изучение современных принципов проектирования игровых и обучающих приложений для обеспечения масштабируемости, надёжности и удобства поддержки кода;
- разработка ПО – создание программного продукта в соответствии с ТЗ, включая реализацию всех функциональных возможностей и пользовательских интерфейсов;
- тестирование – написание тестов для проверки корректности работы приложения и устранения возможных ошибок;
- документирование – подготовка подробной документации с описанием архитектуры, функционала и руководством по использованию приложения;
- такой технологический стек был выбран не случайно - он полностью соответствует современным стандартам качества в разработке мобильных приложений, обеспечивает высокую производительность, масштабируемость решения и позволяет создавать приложения с продуманной архитектурой.

Данный проект имеет значительную практическую ценность как для конечных пользователей, которые получают качественный инструмент для развития своих когнитивных способностей, так и для моего профессионального роста в области мобильной разработки. Результатом работы станет полнофункциональное, тщательно проработанное приложение

с эргономичным дизайном, стабильной работой и увлекательным геймплеем, которое сможет занять достойное место на рынке образовательных мобильных игр.

1 Постановка задачи

1.1 Анализ предметной области

Цель дипломного проекта — создание и тестирование мобильного приложения «Дождь из слов», предназначенного для развития когнитивных навыков пользователей.

Приложение представляет собой интерактивную обучающую платформу, помогающую тренировать:

- скорость печати;
- внимательность;
- концентрацию;
- другие аспекты интеллектуальной деятельности.

Ключевая задача — сочетание эффективного обучения с увлекательным геймплеем, чтобы обеспечить интуитивно понятный и привлекательный пользовательский опыт. Благодаря этому приложение подойдет широкой аудитории, интересующейся саморазвитием и улучшением когнитивных функций.

1.2 Техническое задание на разработку ГОСТ 19.201-78

1.2.1 Введение

Наименование программы

Наименование программы — «Дождь из слов».

Краткая характеристика области применения

Программа предназначена для интеллектуального развития пользователей посредством тренировки когнитивных способностей, таких как внимательность, скорость печати, скорость мышления и логическое мышление, в формате мобильного приложения.

1.2.2 Основания для разработки

Основание для проведения разработки

Основанием для разработки данного дипломного проекта является задание, выданное преподавателем РССК «РГРТУ» Афанасьевым Д.А. на дипломный проект, согласованное с председателем цикловой комиссии

информационных систем и программирования Юдаевым А.Н. и утверждено заместителем директора по учебной работе РССК «РГРТУ» Савельевой О.В. на дипломный проект от 03.04.2025.

Наименование и условное обозначение темы разработки

Наименование темы разработки — «Разработать мобильное приложение «Дождь из слов».

Условное обозначение темы разработки — 73/09.02.07.

1.2.3 Назначение разработки

Функциональное назначение

Функциональным назначением программы является предоставление пользователям инструмента для тренировки когнитивных навыков, включая скорость печати, внимательность, сосредоточенность, в интерактивной и увлекательной форме.

Эксплуатационное назначение

Приложение предназначено для использования на мобильных устройствах под управлением операционных систем Android.

1.2.4 Требования к программе или программному изделию

Требования к функциональным характеристикам

Программа должна обеспечивать выполнение следующих функций:

- тренировка когнитивных навыков посредством интерактивных заданий;
- адаптация скорости печати в зависимости от уровня подготовки пользователя;
- отображение статистики пользователя выполнения заданий;
- обеспечение интуитивно понятного графического интерфейса.

Требования к надежности

Требования к обеспечению надежного функционирования программы не предъявляются, однако приложение должно быть устойчивым к типичным пользовательским ошибкам.

Время восстановления после отказа

Время восстановления после не фатального сбоя операционной системы не должно превышать 20 минут при соблюдении условий эксплуатации технических и программных средств.

Время восстановления после фатального сбоя не должно превышать времени, необходимого для устранения неисправностей технических средств и переустановки программного обеспечения.

Отказы из-за некорректных действий оператора

Отказы программы возможны вследствие некорректных действий пользователя при взаимодействии с операционной системой. Для минимизации таких отказов приложение должно быть разработано с учетом ограничения доступа пользователя к системным функциям.

Условия эксплуатации

Климатические условия эксплуатации должны соответствовать требованиям, предъявляемым к мобильным устройствам (смартфонам и планшетами) в части условий эксплуатации.

Требования к видам обслуживания

Программа не требует проведения специальных видов обслуживания.

Требования к численности и квалификации персонала

Минимальное количество персонала составляет 2 штатные единицы: системный администратор и пользователь (оператор).

Системный администратор должен иметь высшее профильное образование и выполнять следующие задачи:

- поддержание работоспособности технических средств;
- установка и поддержка операционных систем;
- установка приложения.

Оператор должен обладать базовыми навыками работы с мобильными устройствами и графическим интерфейсом операционной системы.

Требования к составу и параметрам технических средств

В состав технических средств входят мобильные устройства (смартфоны или планшеты) со следующими минимальными характеристиками:

- процессор: 4-ядерный, 2 ГГц или выше;
- свободное место на накопителе: 500 МБ;
- оперативная память: 6 ГБ и более;
- операционная система: Android 10.0.

Требования к информационной и программной совместимости

- программа должна быть совместима с операционными системами Android.
- информационная структура должна поддерживать отображение входных данных (заданий) и результатов (статистики) на экране устройства.

Требования к исходным кодам и языкам программирования

Исходные коды программы должны быть реализованы на языке программирования C# с использованием фреймворка .NET MAUI. В качестве среды разработки должна использоваться Microsoft Visual Studio 2022.

Требования к защите информации и программ

Требования к защите информации и программ не предъявляются.

Специальные требования

Программа должна обеспечивать взаимодействие с пользователем через интуитивно понятный графический интерфейс, оптимизированный для сенсорных экранов мобильных устройств.

Требования к программной документации

В состав сопровождающей документации должны входить:

- техническое задание на разработку (ГОСТ 19.201-78);
- руководство оператора (ГОСТ 19.505-79);
- программа, методика и результаты испытаний (ГОСТ 19.301-79).

1.2.5 Технико-экономические показатели

Коммерческое использование программы не предусмотрено.

1.2.6 Стадии и этапы разработки

Стадии разработки

Разработка должна быть проведена в две стадии:

- разработка технического задания;
- разработка технического проекта.

Этапы разработки

- разработка технического задания;
- разработка программы;
- разработка программной документации;
- испытания программы.

Содержание работ по этапам

На этапе разработки технического задания должны быть выполнены следующие работы:

- постановка задачи;
- определение и уточнение требований к техническим средствам;
- определение требований к программе;
- определение стадий, этапов и сроков разработки программы и документации;
- выбор языка программирования, фреймворка и среды разработки;
- согласование и утверждение технического задания.

На этапе разработки программной документации должны быть подготовлены документы в соответствии с требованиями ГОСТ.

На этапе испытаний программы должны быть выполнены следующие виды работ:

- разработка, согласование и утверждение программы и методики испытаний;
- проведение приемосдаточных испытаний;
- корректировка программы и документации по результатам испытаний.

На этапе подготовки и передачи программы должна быть выполнена сдача программы и документации преподавателю.

1.2.7 Порядок контроля и приёмки

Приемосдаточные испытания должны проводиться в колледже преподавателем до 2 июня 2025 года. Испытания программы должны проводиться в соответствии с разработанной студентом и согласованной преподавателем «Программой и методикой испытаний».

2 Проектирование

2.1 Функциональная структура программы

Таблица 1 – Спецификация функций

Название	Действие	Объект	Вид
Игровой процесс			
Игра - Дождь из слов			
InitializeAudio	Инициализация музыки	Интерфейс	Основной
DrawKeyboard	Создание клавиатуры	Интерфейс	Основной
timer_complex	Адаптация сложности	Интерфейс	Основной
ApplyTheme	Инициализация тем приложения	Интерфес	Основной
StartTimers	Инициализация и запуск таймеров	Интерфейс	Основной
LanguageUI	Переделывание под разные языки	Интерфейс	Основной
KeyUp_Clicked	Передача параметров после нажатия	Игровое поле	Основной
Timer_Tick	Главный таймер	Поле текста таймера	Основной
CreateWord	Создание падающего слова	Игровое поле	Основной

Продолжение таблицы 1 - Спецификация модулей

Timer3_Tick	Игровой таймер	Игровое поле	Основной
PlayCorrectWordEffect	Эффект слова	Игровое поле	Основной
ClearOne_Clicked	Удаление одной буквы	Игровое поле	Основной
ClearAll_Clicked	Удаление всех букв	Игровое поле	Основной
pause_Clicked	Пауза игры	Интерфейс	Основной
start_Clicked	Продолжение игры	Интерфейс	Основной
exmenu	Выход в меню	Интерфейс	Основной
againbtn_Clicked	Начать заново	Интерфейс	Основной
Сервисные			
Настройки	Выбор нужных настроек	Интерфейс	Основной
Выход	Выход из игры	Интерфейс	Основной

2.2 Схема модулей программы

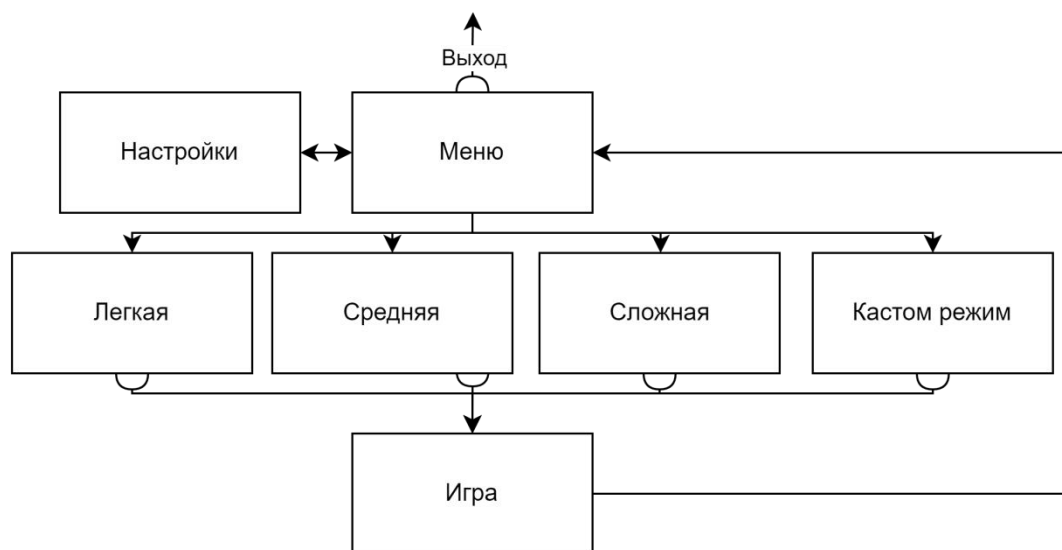


Рисунок 1 - Схема модулей программы

2.3 Паспорта модулей

2.3.1 Модуль InitializeAudio

Данный модуль инициализирует музыку и звуки , при условии, если музыка включена в настройках.

Таблица 2 - Структура

Имя	Тип	Описание
audio	var	Проверка включенной музыки в настройках
_audioService	IAudioService	Переменная для управления
InitializeAsync	void	Создает поток музыки
PlayGameMusic	void	Воспроизводит музыку

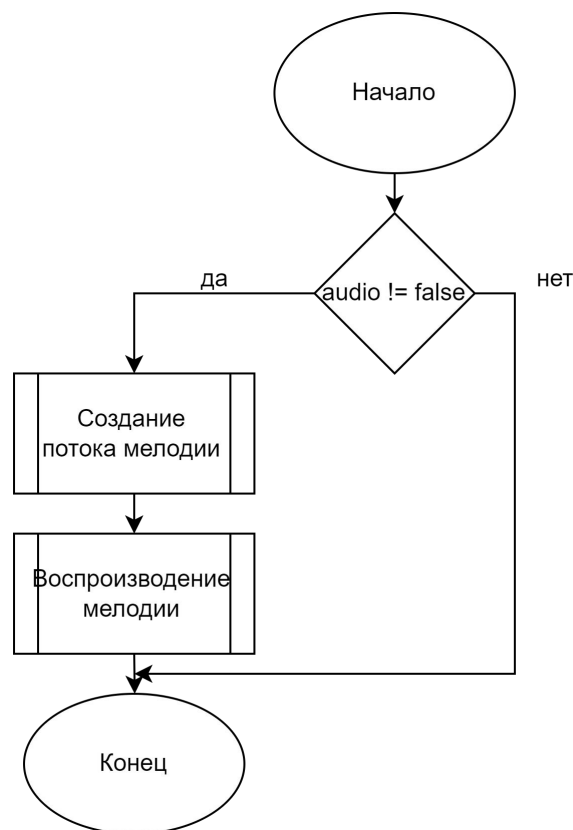


Рисунок 2 - Общий алгоритм модуля

2.3.2 Модуль DrawKeyboard

Данный модуль создает кнопки на виртуальной клавиатуре, определяет язык и размещает их в правильном порядке.

Таблица 3 - Структура

Имя	Тип	Описание
language	var	Определяет какой язык установлен
DrawEnglishKeyboard	void	Определение кнопок в английской раскладке
DrawRussianKeyboard	void	Определение кнопок в русской раскладке
AddKey	void	Размещение и создание кнопок

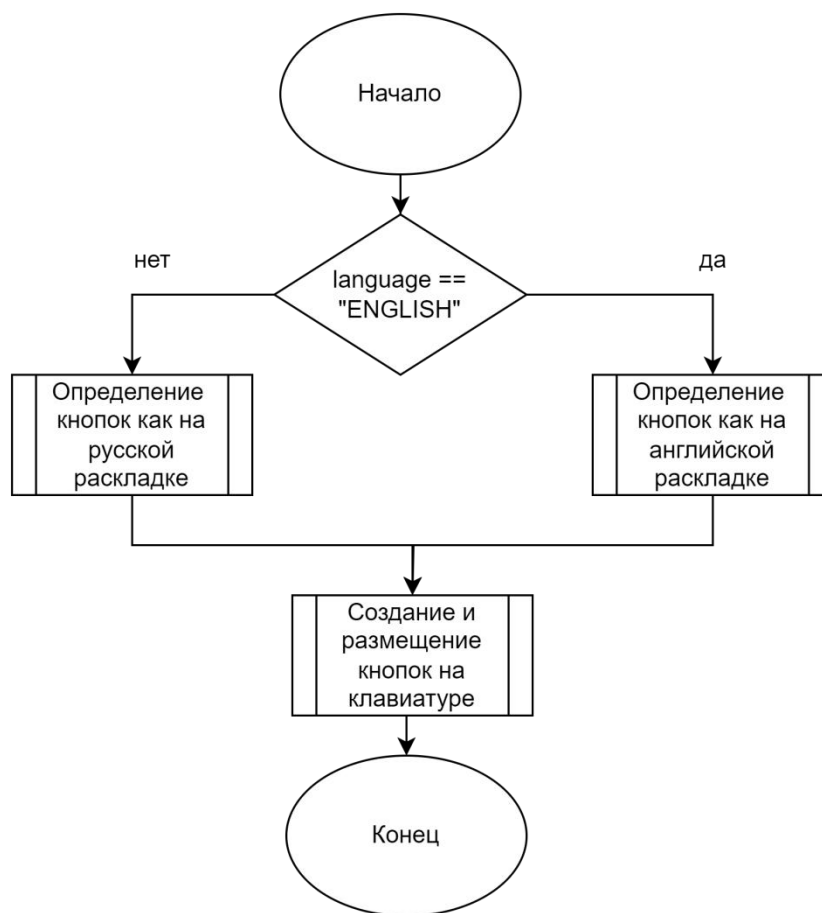


Рисунок 3 - Общий алгоритм модуля

2.3.3 Модуль `timer_complex`

Данный модуль определяет сложность игры, количество букв в словах, интервал появления нового слова, общее время игры, список слов.

Таблица 4 - Структура

Имя	Тип	Описание
labels	List<Label>	Поле ввода
language	var	Определение языка
complextime	int	Уровень игры
_time	TimeSpan	Общее время игры
words	List<string>	Список игровых слов
complex	int	Передача значения, интервал появления нового слова

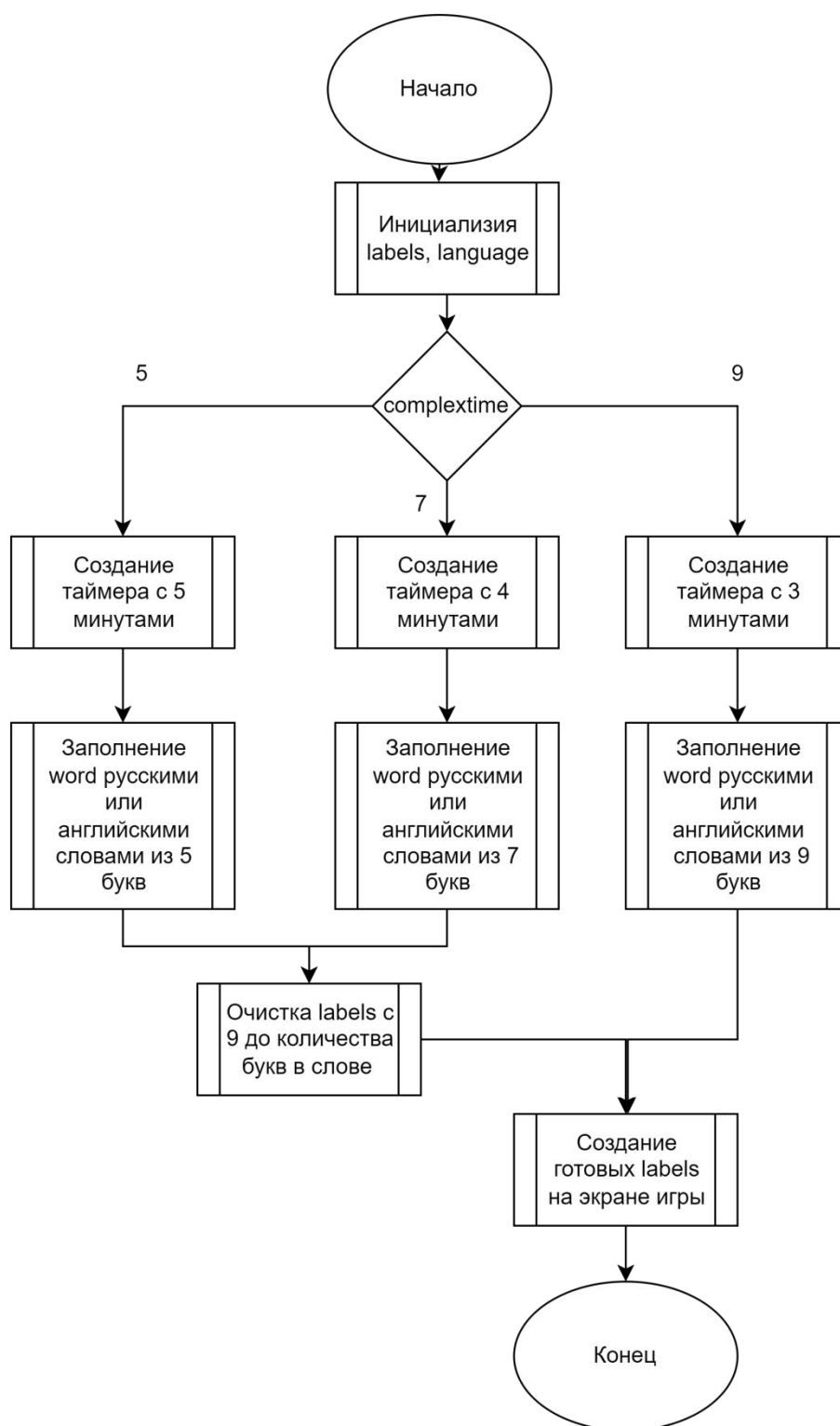


Рисунок 4 - Общий алгоритм модуля

2.3.4 Модуль ApplyTheme

Данный модуль отвечает за цвета всех элементов на экране, в зависимости от того, какая тема была выбрана в настройках. На выбор есть белая, розовая и черная тема.

Таблица 5 - Структура

Имя	Тип	Описание
theme	var	Определяет какая тема была выбрана
ApplyWhiteTheme	void	Создает интерфейс белой темы
ApplyPinkTheme	void	Создает интерфейс розовой темы
ApplyBlackTheme	void	Создает интерфейс черной темы

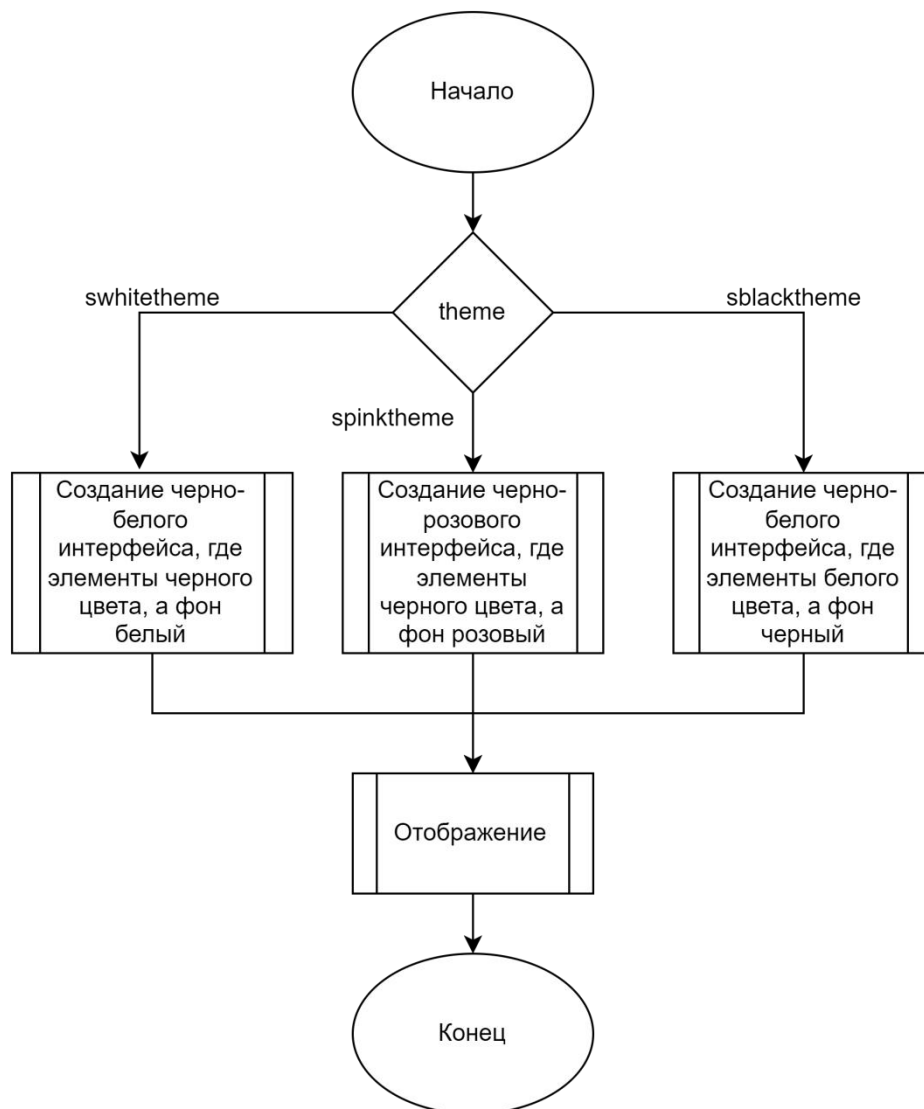


Рисунок 5 - Общий алгоритм модуля

2.3.5 Модуль StartTimers

Данный модуль инициализирует и запускает 2 таймера: основной и игровой.

Таблица 6 - Структура

Имя	Тип	Описание
_timer	IDispatcherTimer	Основной таймер, определяет общее время игры и интервал между падением слов
_timer3	IDispatcherTimer	Игровой таймер, проверяет правильно написано слово или оно уже достигло конца игрового поля

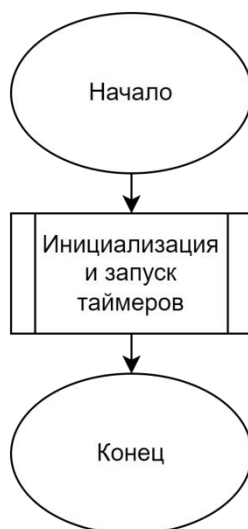


Рисунок 6 - Общий алгоритм модуля

2.3.6 Модуль LanguageUI

Данный модуль переделывает интерфейс под выбранный язык.

Таблица 7 - Структура

Имя	Тип	Описание
language	var	Хранит выбранный язык

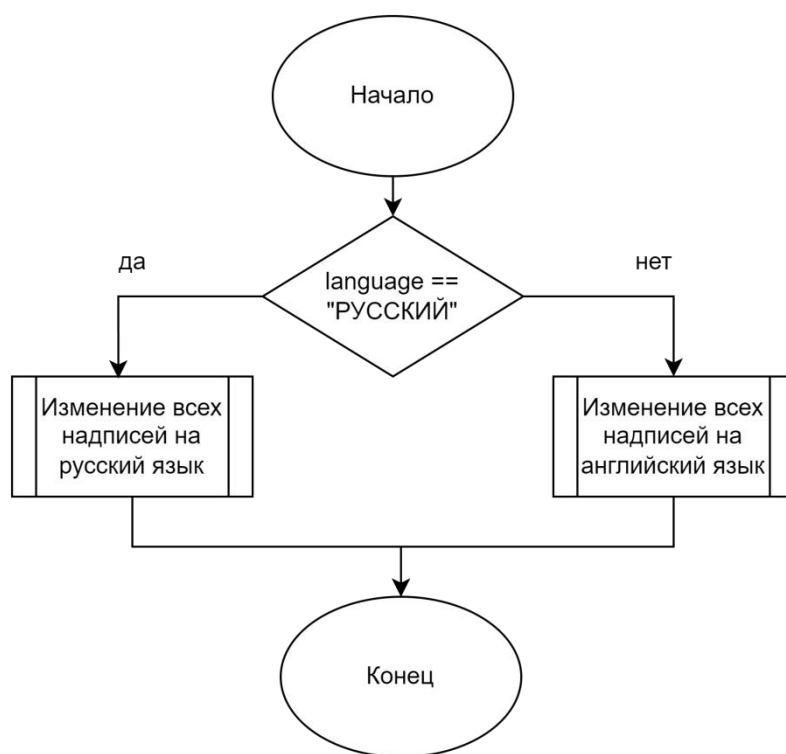


Рисунок 7 - Общий алгоритм модуля

2.3.7 Модуль KeyUp_Clicked

Данный модуль отвечает за передачу текста кнопки в поле ввода слова. Если игрок нажал на паузу или игра окончена, работает проверка, чтобы во время паузы/конца игры писать нельзя было.

Таблица 8 - Структура

Имя	Тип	Описание
absmenu	Frame	Если оно открыто, то писать мы не можем
button	Button	Определяем какая кнопка была нажата
letter	var	Отвечает за текст кнопки
cellindex	Int	Отвечает за количество написанных уже букв
labels	List<Label>	Отвечает за поле ввода

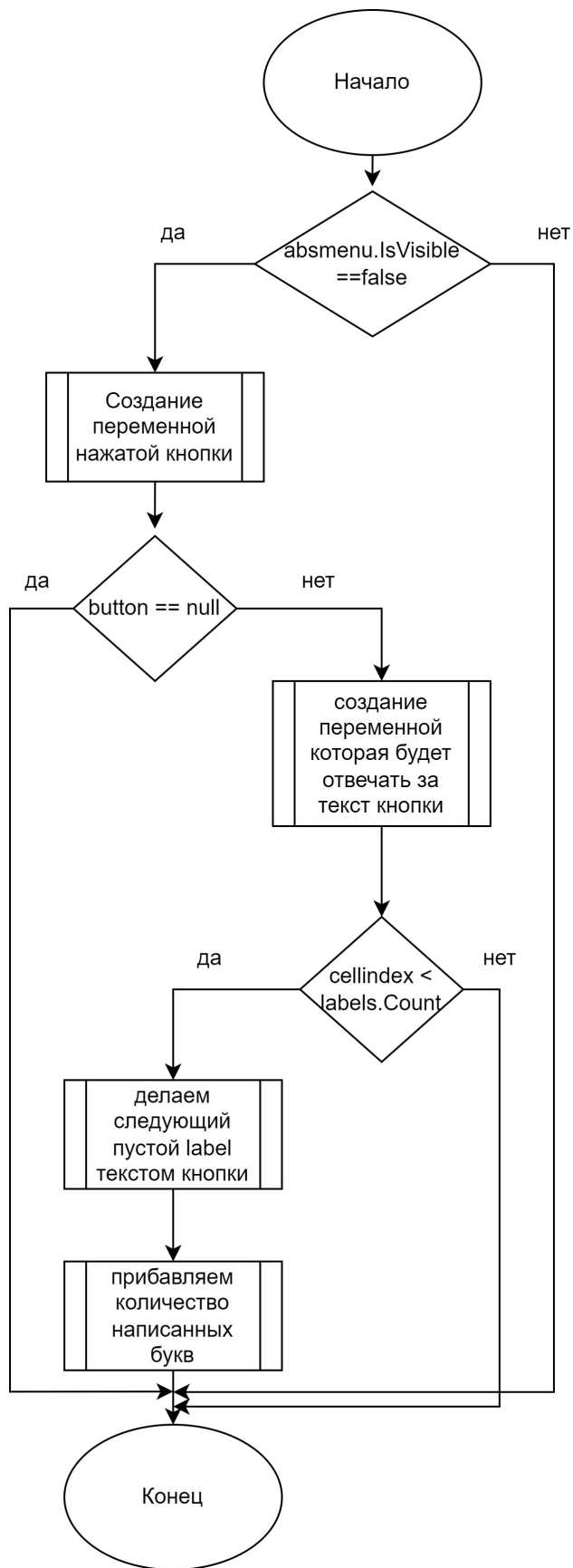


Рисунок 8 - Общий алгоритм модуля

2.3.8 Модуль Timer_Tick

Данный модуль отвечает за общее время игры и интервал между появлениями новых падающих слов, в зависимости от выбранной сложности.

Таблица 9 - Структура

Имя	Тип	Описание
_time	TimeSpan	таймер, обновляющийся каждую секунду
tim	Label	Надпись показывающая оставшееся время
_remainingSeconds	int	Время для подсчета интервала выпадения слова
label1	Label	Если время кончилось, останавливаем анимацию всех слов
_timer	IDispatcherTimer	Останавливаем если игра кончилась
_timer3	IDispatcherTimer	Останавливаем если игра кончилась
paus	Button	Короткая информация о состоянии игры
continuebtn	Button	Если игра кончилась, скрываем кнопку
againbtn	Button	Если игра кончилась, показываем кнопку
absmenu	Frame	Если игра кончилась, показываем окно
CreateWord	void	Функция для создания слова

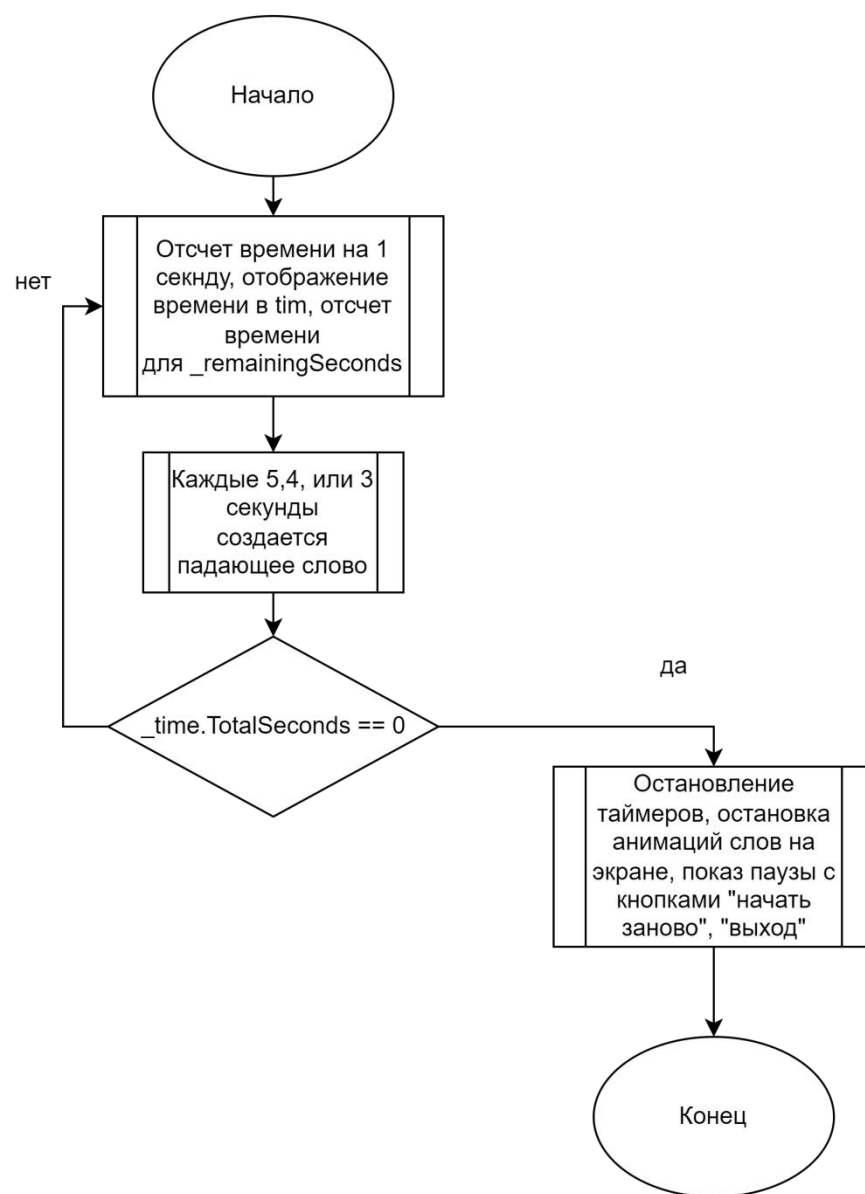


Рисунок 9 - Общий алгоритм модуля

2.3.9 Модуль CreateWord

Данный модуль отвечает за создание падающих слов, они появляются на 100 пикселей сверху экрана, чтобы у игрока было время подготовиться.

Таблица 10 - Структура

Имя	Тип	Описание
displayInfo	var	Берем информацию о параметрах телефона
discreenWidth	var	Вычисляем допустимую ширину для слова

Продолжение таблицы 10 - Структура

randomword	int	Берем случайный индекс списка слов определенной сложности
randomX	double	Создаем начальную точку по оси X
label	var	Создание параметров слова
field	Grid	Добавляем слово на игровое поле
TranslateTo	Task<bool>	Создаем анимацию падения слова
PlayExplaSound	void	Включение звука появления слова



Рисунок 10 - Общий алгоритм модуля

2.3.10 Модуль Timer3_Tick

Данный модуль служит для игрового таймера, проверяет правильно написано слово, или оно уже коснулось конца игрового поля.

Таблица 11 - Структура

Имя	Тип	Описание
currentWordWin	string	Складывает написанные буквы в одно слово
field	Grid	Игровое поле, со словами

Продолжение таблицы 11 - Структура

visibleLabels	Label	Видимое слово которое нужно написать
point	int	Прибавляет/отнимает очки за слово, в зависимости от результата
poin	Label	Отображает очки
PlayCorrectWordEffect	void	Отображает анимацию слова
PlayWinSound	void	Включает звук что слово написано верно
PlayLossSound	void	Включает звук что слово дошло до конца

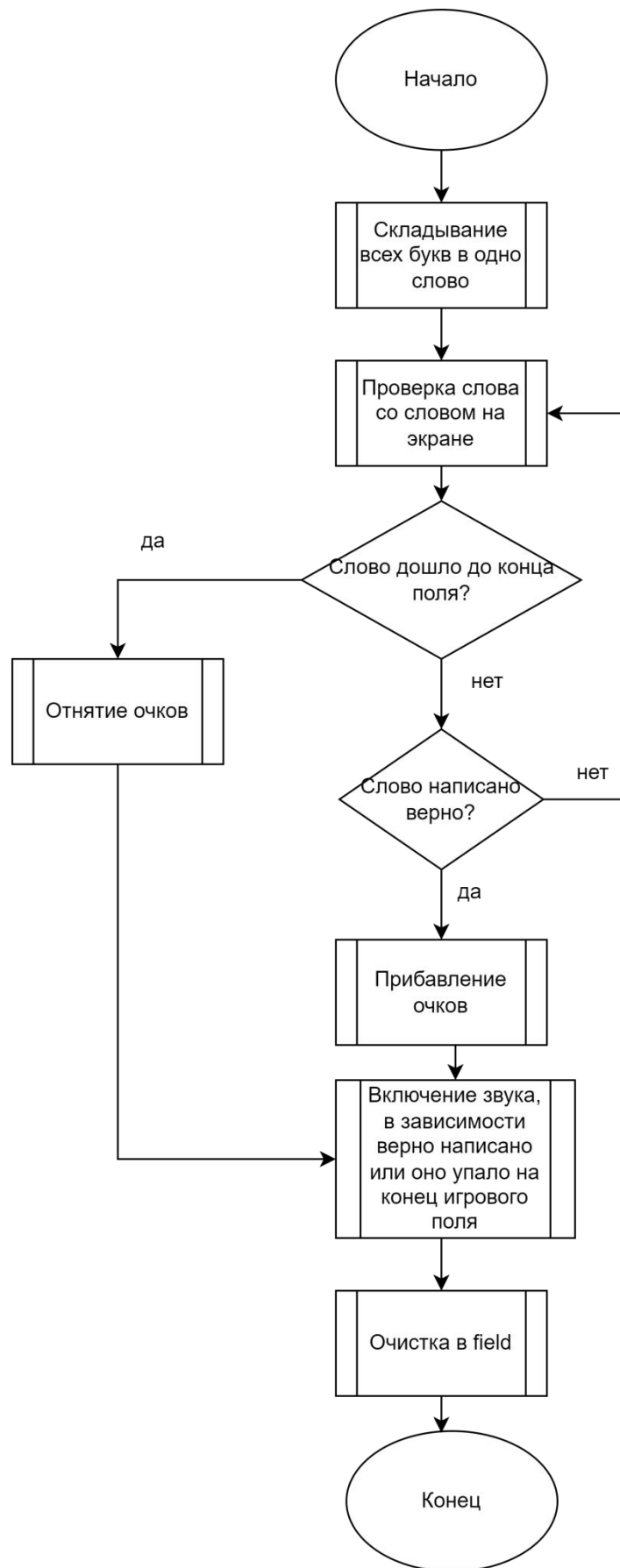


Рисунок 11 - Общий алгоритм модуля

2.3.11 Модуль PlayCorrectWordEffect

Данный модуль отвечает за эффект при правильном написании слова или если слово достигло конца игрового поля. Включается и отключается в настройках.

Таблица 12 - Структура

Имя	Тип	Описание
anim	var	Проверяет включена ли анимация в настройках
checkplay	bool	Словно дошло до конца поля или его написали правильно. Включается либо анимация слово написано верно, либо оно дошло до конца игрового поля
label	Label	Падающее слово
CancelAnimations	void	Останавливаем текущую анимацию падения
ScaleTo	void	Маштарибуем это слово
WhenAll	void	Поднимаем слово
field	Grid	Удаляем слово с игрового поля
ReleaseLabel	void	Освобождаем метку для следующего слова

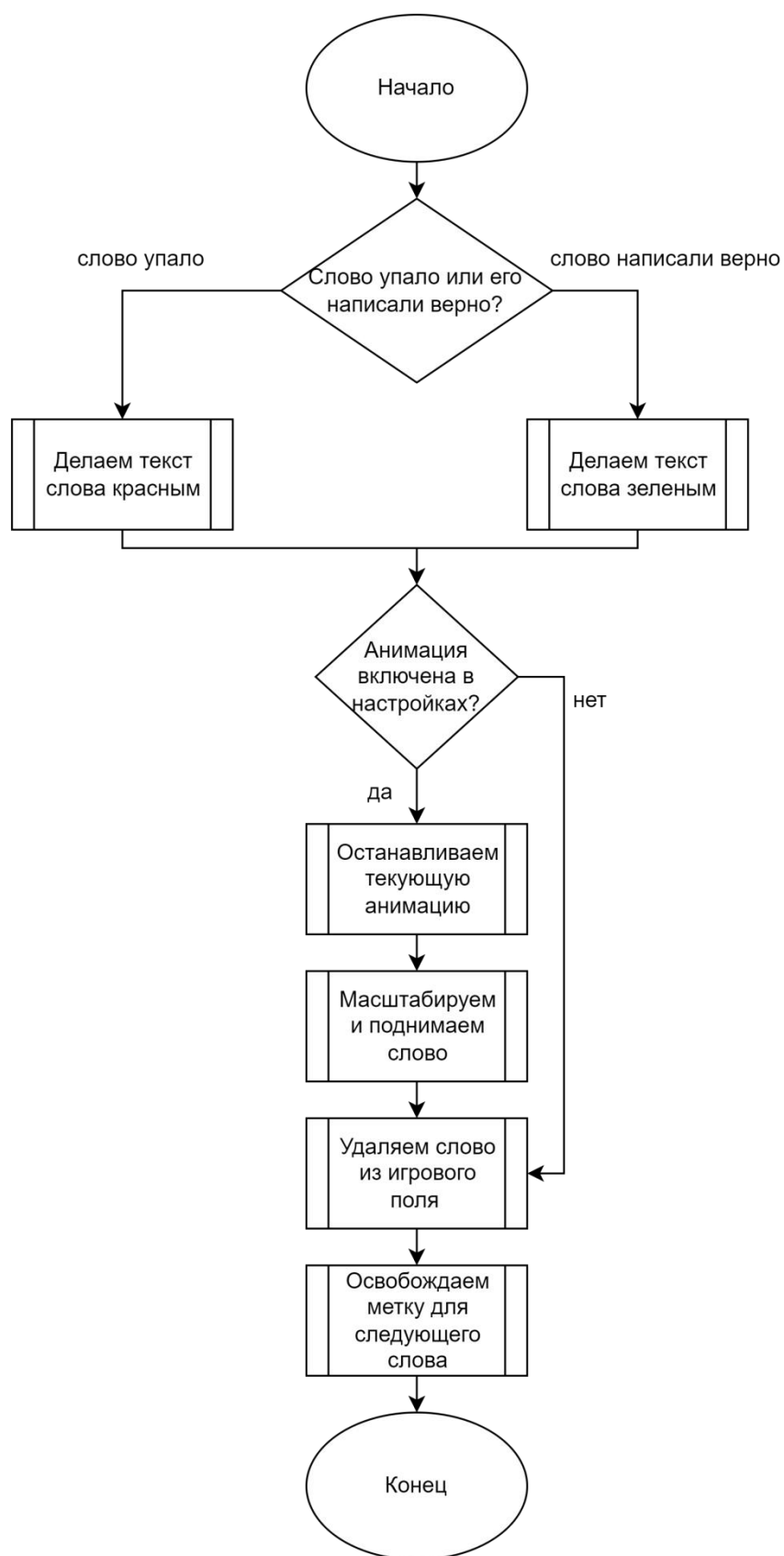


Рисунок 12 - Общий алгоритм модуля

2.3.12 Модуль ClearOne_Clicked

Данный модуль служит очистки последней написанной букве в поле ввода слова. Если открыто окно паузы/конца игры кнопку нажать нельзя.

Таблица 13 - Структура

Имя	Тип	Описание
absmenu	Frame	Окно паузы/конца игры
cellindex	int	Количество написанных букв
labels	List<Label>	Поле ввода

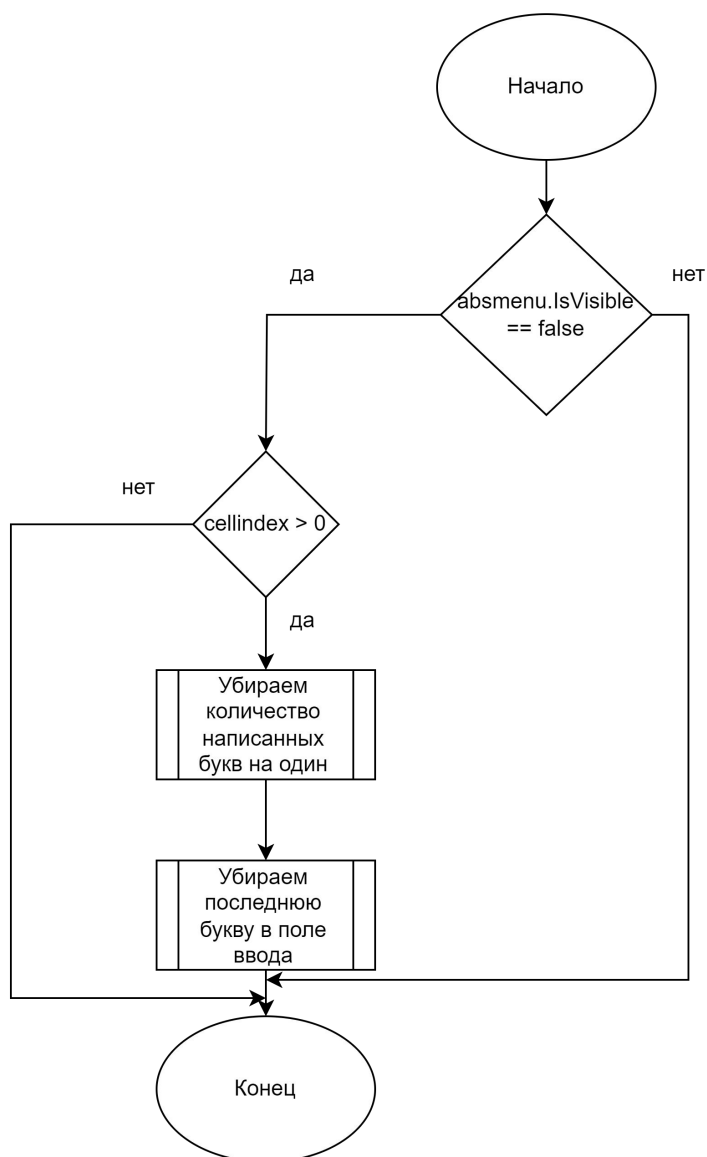


Рисунок 13 - Общий алгоритм модуля

2.3.13 Модуль ClearAll_Clicked

Данный модуль отвечает за очистку всего поля ввода.

Таблица 14 - Структура

Имя	Тип	Описание
absmenu	Frame	Окно паузы/конца игры
cellindex	int	Количество написанных букв
labels	List<Label>	Поле ввода

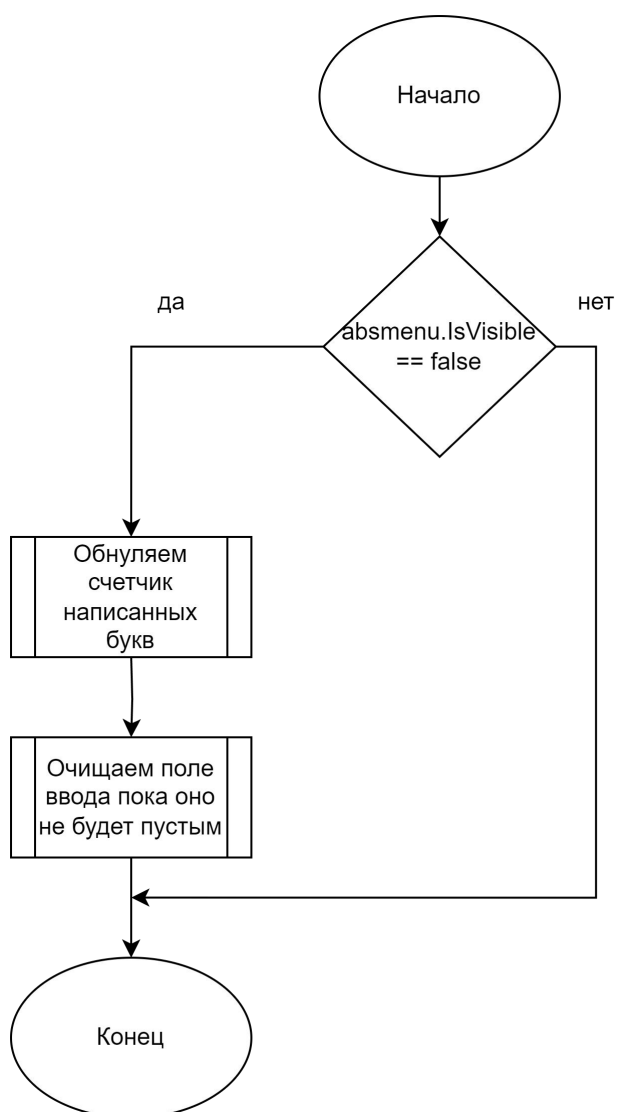


Рисунок 14 - Общий алгоритм модуля

2.3.14 Модуль pause_Clicked

Данный модуль служит остановкой всего игрового процесса, открывает окно с паузой. С двумя кнопками на выбор: «Продолжить», «Выйти».

Таблица 15 - Структура

Имя	Тип	Описание
absmenu	Frame	Окно с паузой
_timer	TimeSpan	Основной таймер
_timer3	TimeSpan	Игровой таймер
OnPauseMusic	void	Функция остановки музыки
label1	Label	Все слова на игровом поле
CancelAnimations	void	Остановка анимации падения слова

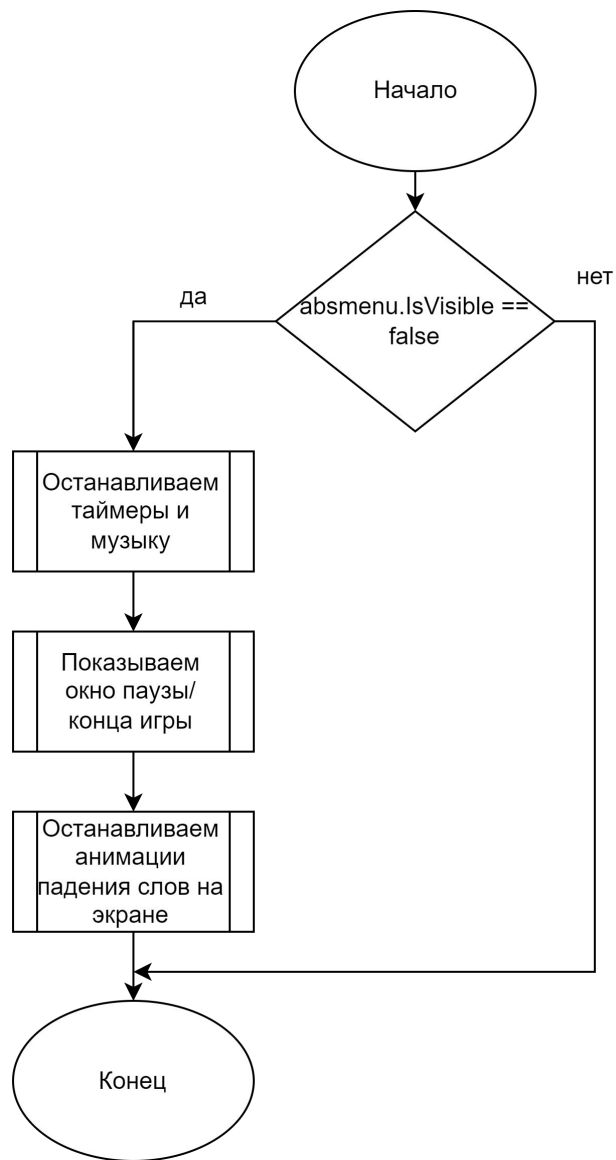


Рисунок 15 - Общий алгоритм модуля

2.3.15 Модуль start_Clicked

Данный модуль отвечает за продолжение игры. Если игрок нажал на окне паузы кнопку «Продолжить».

Таблица 16 - Структура

Имя	Тип	Описание
absmenu	Frame	Закрываем окно с паузой
_timer	TimeSpan	Основной таймер
_timer3	TimeSpan	Игровой таймер
OnResumeMusic	void	Функция

		продолжения музыки
label1	Label	Все слова на игровом поле
TranslateTo	void	Продолжение анимации падения слова

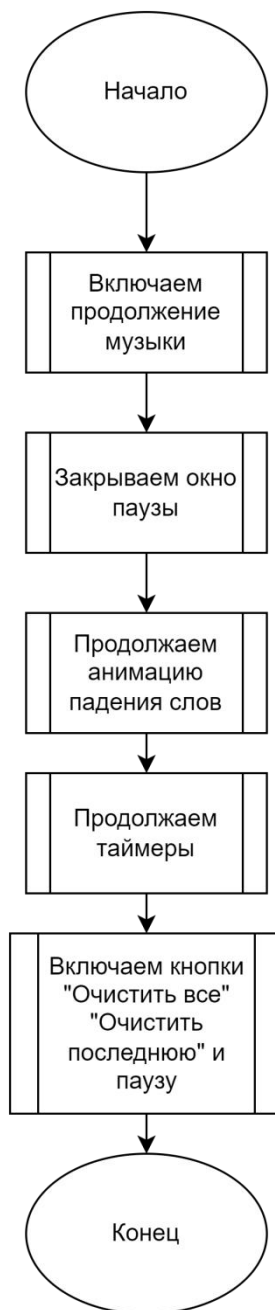


Рисунок 16 - Структура

2.3.16 Модуль exmenu

Данный метод закрывает страницу игры. При нажатии кнопки «Играть» в окне Меню появится кнопка «Продолжить», можно будет продолжить игру

с момента её закрытия. При нажатии кнопки «Настройки» или изменении их параметров кнопка «Продолжить» пропадёт.

Таблица 17 - Структура

Имя	Тип	Описание
PlayMenuMusic	void	Включаем меню музыку, музыка игры автоматически выключается
PopModalAsync	Task<Page>	Переход на предыдущую страницу



Рисунок 17 - Общий алгоритм модуля

2.3.17 Модуль againbtn_Clicked

Данный модуль сработает когда закончится время и игрок в окне паузы/конца игры нажмет кнопку «Начать заново».

Таблица 18 - Структура

Имя	Тип	Описание
absmenu	Frame	Скрываем окно паузы/конца игры
againbtn	Button	Скрываем кнопку
continuebtn	Button	Показываем кнопку при следующем открывании окна паузы
point	int	Обнуляем игровые очки
_remainingSeconds	int	Обновляем подсчет интервала между появлением слов
language	var	Хранит выбранный язык
LanguageUI	void	Проверяем отображение интерфейса выбранного языка
timer_complex	void	Настраиваем игру заново
StartTimers	void	Запускаем таймеры



Рисунок 18 - Общй алгоритм модуля

3 Программная документация

3.1 Руководство оператора ГОСТ 19.505-79

3.1.1 Назначение программы

Программа предназначена для проведения досуга.

Функциональное назначение

Функциональным назначением программы является досуг и развлечение.

Состав функций

1 Функции управления игровым процессом:

- функция старта игры;
- функция нажатия буквы на виртуальной клавиатуре;
- функция очистки последней буквы в поле ввода;
- функция очистки всего поля ввода;
- функция паузы/продолжения игры;
- функция конца игры;
- функция перезапуска игры.

2 Функции таймеров и подсчётов:

- функция отсчета игрового времени;
- функция проверки статуса слова на экране;
- функция подсчёта очков.

3 Функция работы с игровым полем:

- функция создания игрового поля;
- функция включения музыки, если она включена в настройках;
- функция определения выбранного языка;
- функция создания клавиатуры выбранного языка;
- функция создания, анимации и очистки падающего слова;
- функция анимации правильно написанного слова или если слово

коснулось конца игрового поля;

- функция вывода времени и очков.

4 Функции уровня сложности:

- функция установки сложности игры;
- функция настройки игры с выбранной сложностью.

5 Функции интерфейса:

- функция отображения паузы игры;
- функция управления настройками;
- функция звука слова появления/падения на конец поля/правильно

написанного, если они включены в настройках.

6 Системные функции:

- функция продолжения игры с того места, где закончили;
- функция обработки навигации между экранами;
- функция управления темами приложения;
- функция управления языками приложения;
- функция управления музыкой;
- функция управления анимациями;
- функция выхода из приложения.

3.1.2 Условия выполнения программы

Климатические условия эксплуатации

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

Требования к составу и параметрам технических средств

В состав технических средств входят мобильные устройства (смартфоны или планшеты) со следующими минимальными характеристиками:

- процессор: 6-ядерный, 2 ГГц или выше;
- свободное место на накопителе: 500 МБ;
- оперативная память: 6 ГБ и более;
- операционная система: Android 10.0.

Требования к информационной и программной совместимости

- программа должна быть совместима с операционными системами Android.
- информационная структура должна поддерживать отображение входных данных (заданий) и результатов (статистики) на экране устройства.

Требования к программным средствам, используемым программой

Системные программные средства, используемые программой, должны быть представлены лицензионной локализованной версией ОС Android 10.0 или новее с платформой .NET MAUI (C# / .NET 8.0).

3.1.3 Выполнение программы

Загрузка и запуск программы

В случае успешного запуска программы на экране после отображения загрузки движка .NET MAUI с заменённым изображением на логотип разработчика (рисунок 19), дальше будет загружено меню показанное на рисунке 20.



Рисунок 19 - Отображение загрузки движка



Рисунок 20 - Меню

На рисунке 21 показаны кнопки управления:

- кнопка «играть» отвечает за переход к выбору сложности игры (рисунок 22);
- кнопка «настройки» отвечает за переход к странице настроек приложения (рисунок 23).
- кнопка «выход» отвечает за выход из приложения (рисунок 24).



Рисунок 21 - кнопки управления



Рисунок 22 - кнопка «играть»



Рисунок 23 - кнопка «настройки»



Рисунок 24 - кнопка «выход»

При нажатии на кнопку «настройки» происходит переход на страницу параметров приложения(рисунок 25). На странице находится:

- кнопка «язык»(рисунок 26);
- кнопка «темы»(рисунок 27);
- переключатель «звук»(рисунок 28);
- переключатель «анимации»(рисунок 29);



МЕНЮ НАСТРОЕК

ЯЗЫК

ТЕМЫ

ЗВУК 

АНИМАЦИИ 



Рисунок 26 - кнопка «язык»



Рисунок 27 - кнопка «темы»



Рисунок 28 - переключатель «звук»



Рисунок 29 - переключатель «анимации»

При нажатии на кнопку «язык» появится на выбор два языка: русский и английский(рисунок 30). При выборе языка, меняется как весь интерфейс(рисунок 31 и рисунок 32), так и игровая клавиатура и выпадающие слова.



РУССКИЙ ENGLISH

Рисунок 30 - выбор языка



SETTINGS MENU

LANGUAGE

РУССКИЙ

ENGLISH

THEME

SOUND



ANIMATIONS



Рисунок 31 - интерфейс страницы настроек с английским языком



Рисунок 32 - интерфейс меню с английским языком

При нажатии на кнопку «темы» появится выбор из трёх тем: белая, розовая и чёрная(рисунок 33). При выборе темы, меняется весь интерфейс всего приложения, в цветовой гамме, которую выбрали(рисунок 34, рисунок 35, рисунок 36, рисунок 37, рисунок 38 и рисунок 39).



Рисунок 33 - выбор тем приложения



Рисунок 34 - интерфейс меню настроек с розовой темой



Рисунок 35 - интерфейс меню с розовой темой



Рисунок 36 - интерфейс выбора сложности с розовой темой

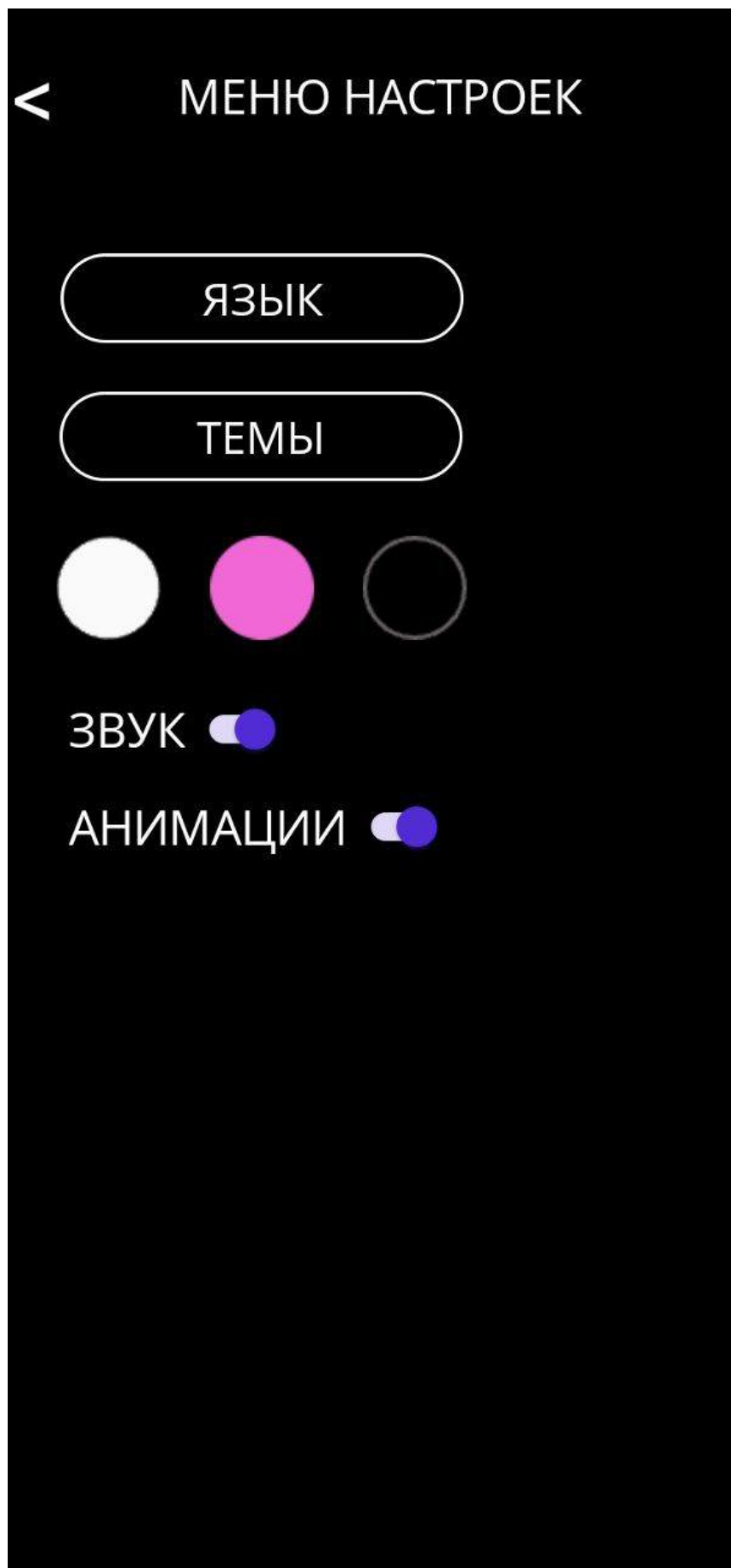


Рисунок 37 - интерфейс меню настроек с чёрной темой



Рисунок 38 - интерфейс меню с чёрной темой



Рисунок 39 - интерфейс выбора сложности с чёрной темой

При нажатии переключателя «звук» он уйдёт в состояние «выключен»(рисунок 40) и выключит как музыку во всём приложении, так и звуки в игровой странице.



Рисунок 40 - выключенный переключатель «звук»

При нажатии переключателя «анимации» он уйдёт в состояние «выключен»(рисунок 41) и выключит анимации выпадающих слов когда слово коснулось конца игрового поля или слово было написано верно.



Рисунок 41 - выключенный переключатель «анимации»

Чтобы начать игру, нужно нажать кнопку «играть», которая показывает уровни сложности(рисунок 42), и выбрать уровень сложности. Уровни сложности различаются в:

- время игры;
- интервал между выпадением новым словом;
- количество букв, выпадающих слов;
- количество букв, которые нужно написать в поле ввода;
- количество очков за правильно написанное слово.



Рисунок 42 - выбор сложности игры

При выборе уровня «Легкая» время игры будет 5 минут, интервал между новыми словами составляет 5 секунд, количество букв выпадающих слов и тех, что нужно успеть написать в поле ввода - 5 букв, количество очков за правильно написанное слово - 20 очков. Легкая сложность показана на рисунке 43. Тема приложения - «Белая».



4:47

0

файлы

акулы

— — — — —



Рисунок 43 - игра со сложностью «Легкая» и белой темой

При выборе уровня «Средняя» время игры будет 3 минуты, интервал между новыми словами составляет 4 секунды, количество букв выпадающих слов и тех, что нужно успеть написать в поле ввода - 7 букв, количество очков за правильно написанное слово - 30 очков. Средняя сложность показана на рисунке 44. Тема приложения - «Розовая».

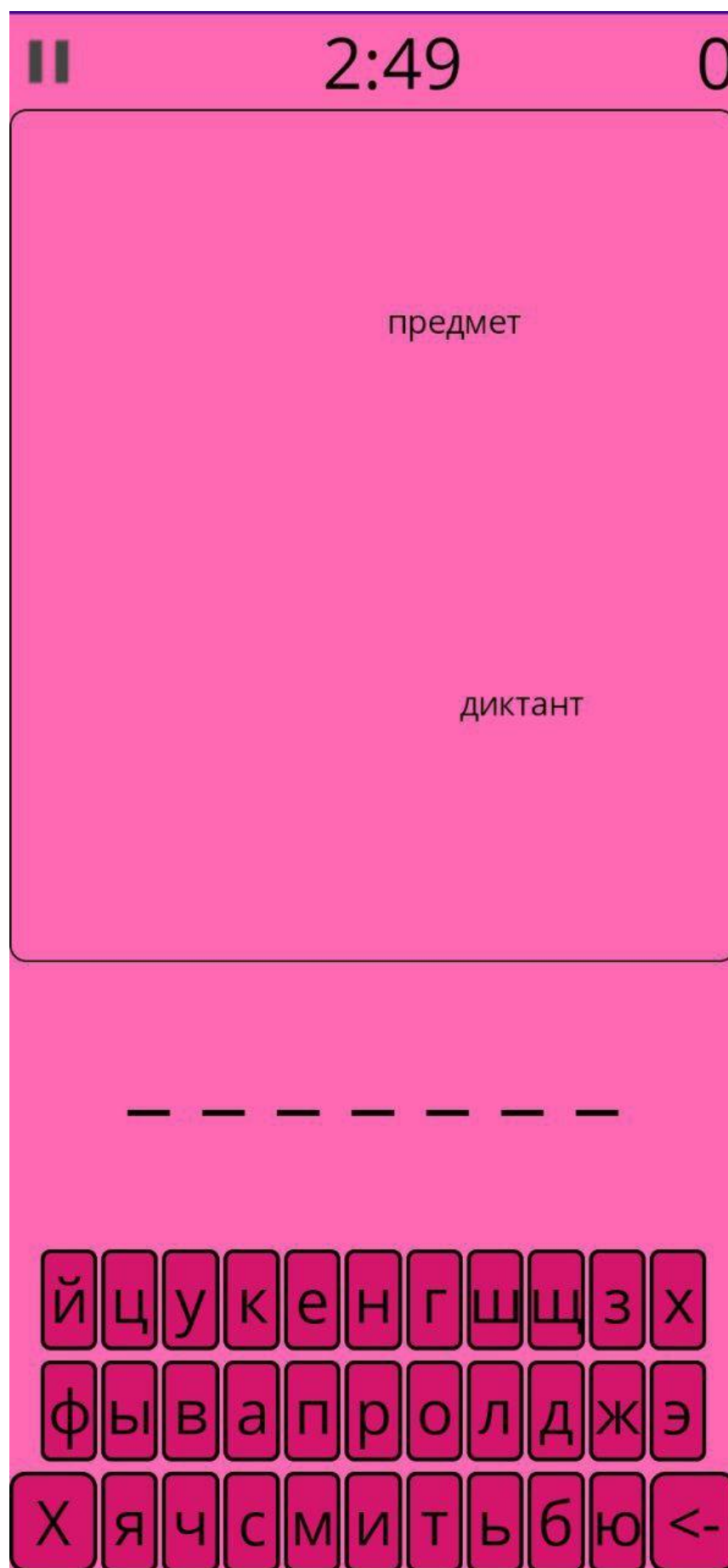


Рисунок 44 - игра со сложностью «Средняя» и розовой темой

При выборе уровня «Сложная» время игры будет 2 минуты, интервал между новыми словами составляет 2 секунды, количество букв выпадающих слов и тех, что нужно успеть написать в поле ввода - 9 букв, количество очков за правильно написанное слово - 50 очков. Сложная сложность показана на рисунке 45. Тема приложения - «Черная».

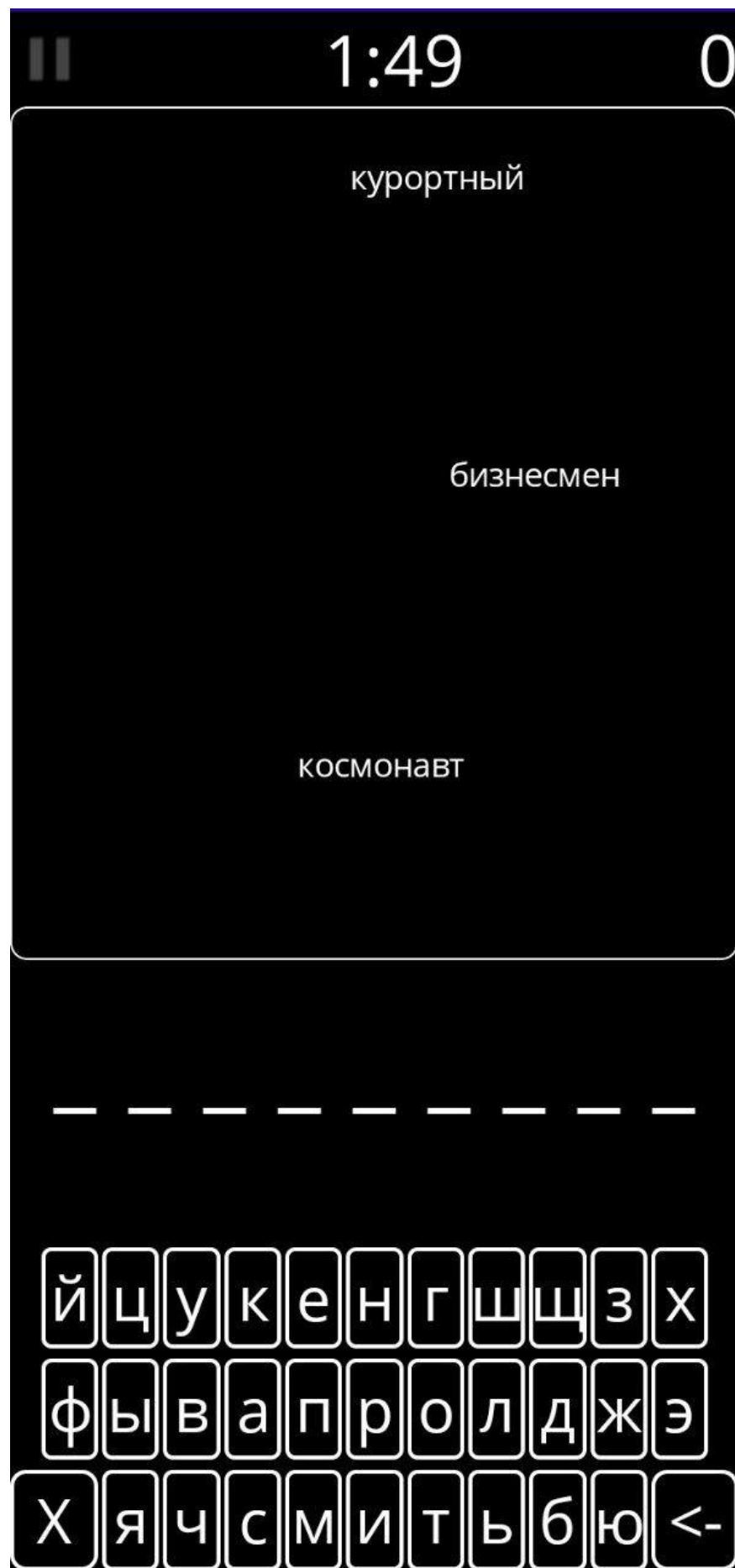


Рисунок 45 - игра со сложностью «Сложная» и черной темой

При выборе сложности «Кастом» откроется страница для настройки игры со своими правилами(рисунок 46). На странице можно выбрать:

- скорость падения слов;
- количество прибавляемых/отнимающих очков за правильно написанное слово/если слово коснулось конца игрового поля;
- время игры;
- количество букв падающих слов и которые нужно написать в поле ввода.



КАСТОМНЫЙ РЕЖИМ

СКОРОСТЬ

10-100000

ОЧКИ

1-10000

ВРЕМЯ

1-10000 мин

КОЛИЧЕСТВО БУКВ

5,7,9.

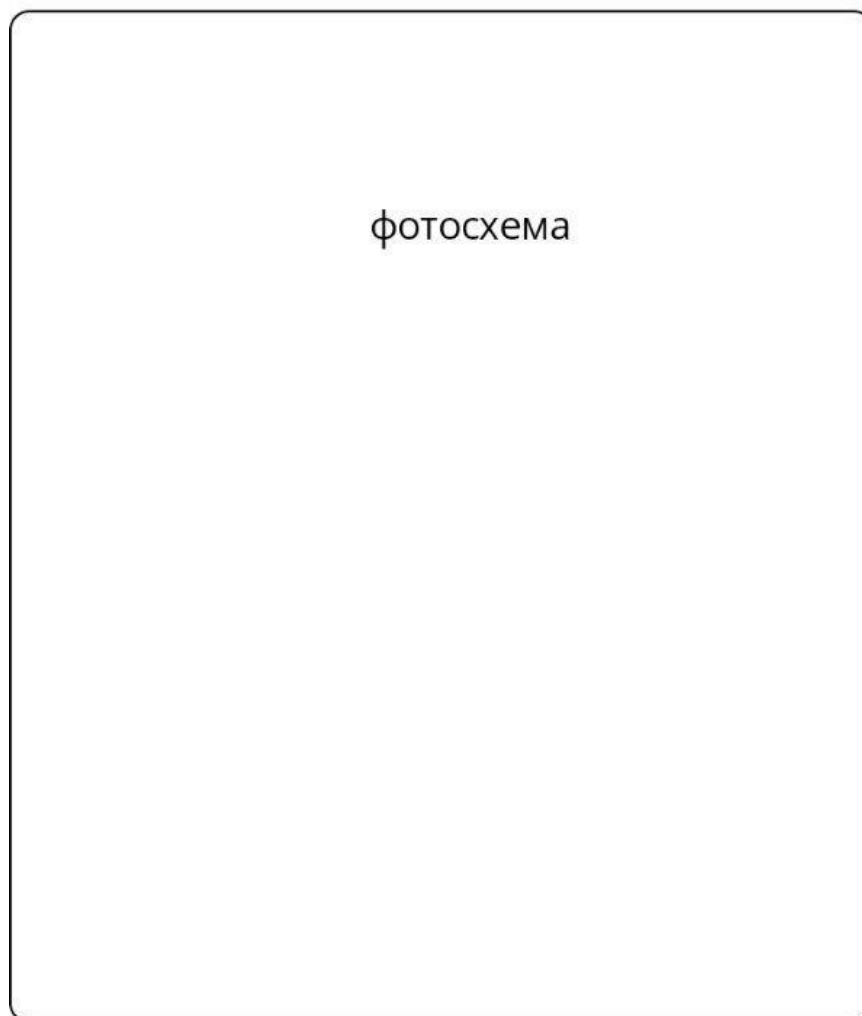
ИГРА

Если написать неподходящие значения(подсказки указаны в полях ввода) и нажать кнопку «игра», ничего не произойдет, это указывает что нужно указать корректные данные. После заполнения правильных значений, приложение перейдет в игру уже с указанными параметрами(рисунок 47).



457:51

500



фотосхема



Рисунок 47 - игра со сложностью «Кастом» и белой темой

При выборе английского языка в настройках сложности не меняются, меняются выпадающие слова и виртуальная клавиатура.(рисунок 48)



4:48

0

sunny

apple

— — — — —

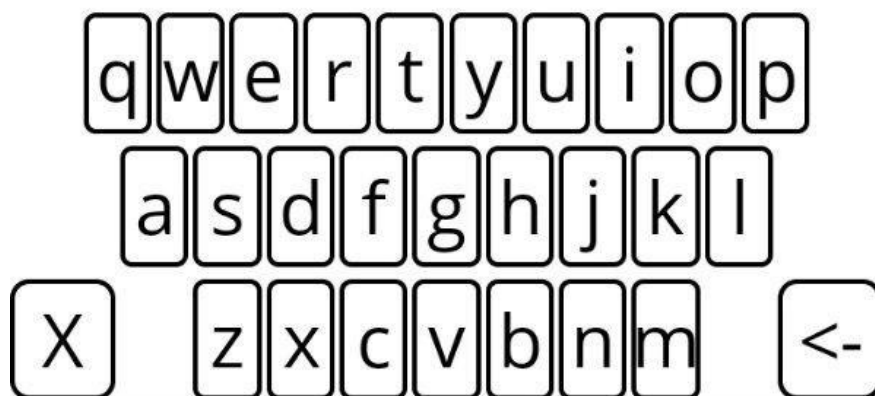


Рисунок 47 - игра с английским языком

При нажатии на кнопку «Пауза» в левом верхнем углу, игра останавливается вместе с музыкой и появится окно с паузой(рисунок 49). В окне расположены две кнопки «продолжить» и «выйти», при продолжении игра продолжится, при выходе, приложение перебросит в окно меню. В меню при выборе сложности появится кнопка «Продолжить»(рисунок 50). При нажатии на нее, игра продолжится с того момента, где она была остановлена.



4:59

0



— — — — —



Рисунок 49 - окно паузы



Рисунок 50 - кнопка «Продолжить» при выборе сложности

При условии, если игра была запущена, после нажатии на кнопку «Выход» в главном меню, высветится сообщение об уверенности выйти. При нажатии «да» приложение будет закрыта.(рисунок 51)

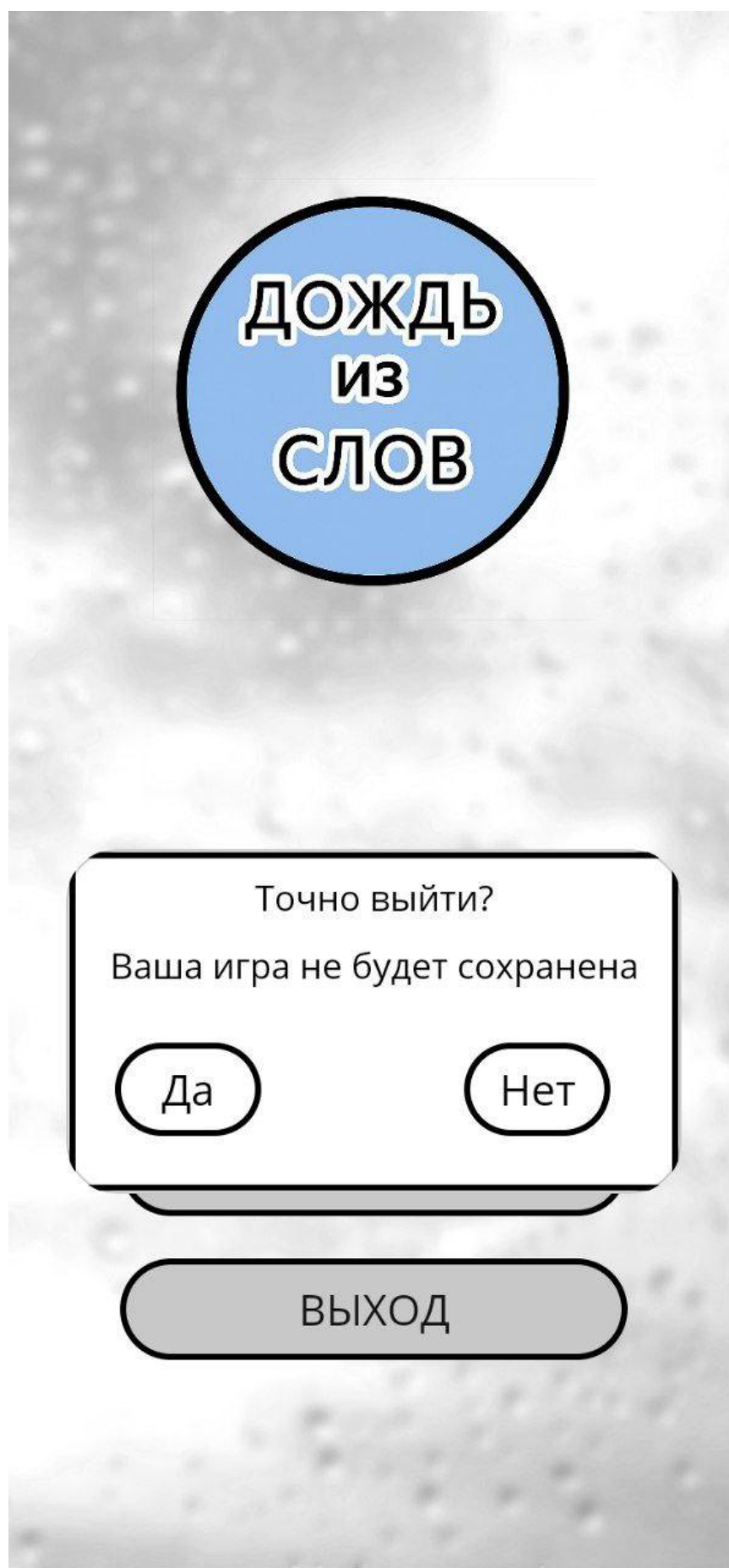


Рисунок 51 - сообщение о точном выходе

Если игровое время окончено, будет показано окно с окончанием игры. На нем будет изображено две кнопки: «заново» и «Выход». При нажатии на кнопку «заново» игра начнется с начала с теми же параметрами.(рисунок 52)



— — — — —



Рисунок 52 - окно с окончанием игры

4 Тестирование программы

4.1 Программа, методика и результаты испытаний ГОСТ 19.301-79

4.1.1 Объект испытаний

Наименование программы - «Дождь из слов».

Назначение разработки

Программа предназначена для развлечения и проведения досуга.

Условное обозначение темы разработки 73/09.02.07.

4.1.2 Цель испытаний

Цель проведения – проверка соответствия характеристик разработанной программы функциональным и отдельным иным видом требований, изложенным в документе техническое задание.

Основания для проведения испытаний

Основанием для проведения испытаний данного дипломного проекта является задание, выданное преподавателем РССК «РГРТУ» Афанасьевым Д.А. на дипломный проект, согласованное с председателем цикловой комиссии информационных систем и программирования Юдаевым А.Н. и утверждено заместителем директора по учебной работе РССК «РГРТУ» Савельевой О.В. на дипломный проект от 03.04.2025.

Место и продолжительность испытаний

Колледж, объект заказчика. Срок 2 месяца.

Перечень документов, предъявляемых на испытания

Состав программной документации должен включать в себя:

- техническое задание на разработку ГОСТ 19.201-78;
- руководство оператора ГОСТ 19.505-79;
- программа, методика и результаты испытаний ГОСТ 19.301-79.

Объем испытаний

Испытания проводятся в два этапа:

- ознакомительный;
- испытания.

Перечень проверок, проводимых на 1 этапе испытаний, должен включать в себя:

- проверку комплектности программной документации;
- проверку комплектности состава технических и программных

Перечень проверок, проводимых на 2 этапе испытаний

Перечень проверок, проводимых на 2 этапе испытаний, должен включать в себя:

- проверку соответствия технических характеристик программы;
- проверку степени выполнения требований функционального назначения программы.

Количественные и качественные характеристики, подлежащие оценке

Программа должна обеспечивать выполнение следующих функций:

- тренировка когнитивных навыков посредством интерактивных заданий;
- адаптация сложности заданий в зависимости от уровня подготовки пользователя;
- отображение статистики пользователя выполнения заданий;
- обеспечение интуитивно понятного графического интерфейса;
- сохранение пользовательских данных для отслеживания прогресса.

4.1.3 Требования к программе

При проведении испытаний функциональные характеристики (возможности) программы подлежат проверке на соответствие требованиям, изложенным в п. «Требования к функциональным выполняемым функций» технического задания.

4.1.4 Требования к программной документации

Состав программной документации должен включать в себя:

- техническое задание на разработку ГОСТ 19.201-78;
- руководство оператора ГОСТ 19.505-79;
- программа, методика и результаты испытаний ГОСТ 19.301-79;

- руководство программиста ГОСТ 19.504-79;
- руководство системного программиста ГОСТ 19.503-79.

4.1.5 Средства и порядок испытаний

Технические средства, используемые во время испытаний

В состав технических средств входят мобильные устройства (смартфоны или планшеты) со следующими минимальными характеристиками:

- процессор: 4-ядерный, 2 ГГц или выше;
- свободное место на накопителе: 500 МБ;
- оперативная память: 6 ГБ и более;
- операционная система: Android 10.0.

Требования к информационной и программной совместимости

- программа должна быть совместима с операционными системами Android.
- информационная структура должна поддерживать отображение входных данных (заданий) и результатов (статистики) на экране устройства.

Порядок проведения испытаний

Испытаний должны проводиться поэтапно согласно п. «Перечень этапов испытаний» настоящего документа.

Условия проведения испытаний

Испытания должны проводиться в нормальных климатических условиях по ГОСТ 22261-94. Условия проведения испытаний приведены ниже:

- температура окружающего воздуха, °С - 20 ± 5 ;
- относительная влажность, % - от 30 до 80;
- атмосферное давление, кПа – от 84 до 106;
- частота питающей электросети, Гц – $50 \pm 0,5$;
- напряжение питающей сети переменного тока, В - $220 \pm 4,4$.

Условия начала и завершения отдельных этапов испытаний

Необходимым и достаточным условием завершения 1 этапа испытаний и начала 2 этапа испытаний является успешное завершение проверок, проводимых на 1 этапе.

Условием завершения 2 этапа испытаний является успешное завершение проверок, проводимых на 2 этапе.

Требования к техническому обслуживанию

Требования к техническому обслуживанию не предъявляются.

4.1.6 Методика испытаний

Методика проведения проверки комплектности программной документации

Проверка комплектности программной документации на программное изделие производится визуально преподавателем. В ходе проверки сопоставляется состав и комплектность программной документации, представленный исполнителем, с перечнем документации.

Проверка считается завершенной в случае соответствия состава и комплектности программной документации, приведенному в указанном выше пункте.

Методика проведения проверки степени выполнения требований функционального назначения программы

Тестирование черного ящика или поведенческое тестирование – стратегия тестирования функционального поведения программы с точки зрения внешнего мира, при котором не используется знание о внутреннем устройстве тестируемого объекта. Под стратегией понимаются систематические методы отбора и создания тестов для тестового набора. Стратегия поведенческого теста исходит из технических требований и их спецификаций.

4.2 Протокол испытаний

Результаты предварительного тестирования в соответствии с таблицей 8.

Таблица 8 – Функциональное тестирование «чёрного ящика»

Тестовая спецификация						
Разработано: Евдокимов Станислав			Начальные установки: нет			
Источник текстовых данных:			Цели: Проверки степени выполнения требований функционального назначения программы			
Test Case#	Описание	Шаги теста	Ожидаемые результаты	Реальные результаты	Прошёл/Провал	Тестер /Дата
Тестирование окна меню						
1	Начало игры	Нажать кнопку «Играть», выбрать сложно	Открывается игра, запускается таймер и игра начинается	Открывается игра, запускается таймер и игра начинается	Прошёл	Алексей Калёных 25.05.2025
2	Выбор разной сложности	Выбрать из списка другую сложно	Также открывается игра, но с другими параметрами	Также открывается игра, но с другими параметрами	Прошёл	Алексей Калёных 25.05.2025
3	Нажать кнопки «Настро	Нажать на кнопку «Настро	Откроется страница с настройкам и	Откроется страница с настройкам и	Прошел	Алексей Калёных 25.05.2025

	йки»	йки»	приложения	приложения		
4	Нажати е «Выход »	Нажать на кнопку «Выход » на главном окне	Если игра не была запущена приложение закроется, если игра была запущена покажет сообщение о уточнении выхода	Если игра не была запущена приложение закроется, если игра была запущена покажет сообщение о уточнении выхода	Прошел	Алексей Калёных 25.05.202 5
Тестирование функций игры						
5	Нажати е «Пауза»	Нажать на паузу в левом верхнем углу	Открываетс я окно с паузой, игра останавлива ется	Открываетс я окно с паузой, игра останавлива ется	Прошёл	Алексей Калёных 25.05.202 5
6	Нажати е букв в виртуал ьной клавиат уре	Нажати е букв в виртуал ьной клавиат уре	Буква с кнопки передается в поле для ввода	Буква с кнопки передается в поле для ввода	Прошёл	Алексей Калёных 25.05.202 5
7	Срабат ывание эффекта	Не писать слово,	Будет эффект в зависимости	Будет эффект в зависимости	Прошёл	Алексей Калёных 25.05.202

	при правиль ном написан ии слова/сл ово дошло до конца	дождать ся когда упадет/ Написат ь правиль но слово, как на экране	от выполненно го шага, начисление/ отнятие игровых очков	от выполненно го шага, начисление/ отнятие игровых очков		5
8	Кнопка «выход » в окне паузы	Нажать кнопку «выход » в окне паузы	Откроется главное меню	Откроется главное меню	Прошёл	Алексей Калёных 25.05.202 5
9	Кнопка «очисти ть все»	Нажать кнопку «очисти ть все»	Очистится поле ввода	Очистится поле ввода	Прошёл	Алексей Калёных 25.05.202 5
10	Кнопка «очисти ть одну»	Нажать кнопку «очисти ть одну»	Очистится последняя буква	Очистится последняя буква	Прошёл	Алексей Калёных 25.05.202 5
Тестирование меню настроек						
11	Выбор другого языка	Нажать кнопку «язык» и	Поменяется весь интерфейс и игра будет	Поменяется весь интерфейс и игра будет	Прошёл	Алексей Калёных 25.05.202 5

		выбрать другой язык	на другом языке	на другом языке		
12	Выбор темы	Нажать кнопку «темы»	В зависимости от темы будет изменен интерфейс на выбранную тему	В зависимости от темы будет изменен интерфейс на выбранную тему	Прошёл	Алексей Калёных 25.05.202 5
13	Переключатель «звук»	Нажать на переключатель «звук»	Если выключить музыка во всем приложении будет выключен, если включить музыка будет включена	Если выключить музыка во всем приложении будет выключен, если включить музыка будет включена	Прошёл	Алексей Калёных 25.05.202 5
14	Переключатель «анимации»	Нажать на переключатель «анимации»	Если выключить анимации при правильнона	Если выключить анимации при правильнона	Прошёл	Алексей Калёных 25.05.202 5

		ции»	писанном слове/слово дошло до конца не будут отображатьс я, если включить анимации будут отображатьс я	писанном слове/слово дошло до конца не будут отображатьс я, если включить анимации будут отображатьс я		
Тестирование «Кастом» сложности						
15	Нажать «Касто м»	При выборе сложно сти выбрать «Касто м»	При нажатии откроется страница для написания параметров самой игры	При нажатии откроется страница для написания параметров самой игры	Прошёл	Алексей Калёных 25.05.202 5
16	Поля ввода в «Касто м»	Написат ь некорре ктные или пустые данные и	Ничего не произойдет	Ничего не произойдет	Прошёл	Алексей Калёных 25.05.202 5

		нажать кнопку «Игра»				
17	Кнопка «Игра»	Ввести правиль ные данные и нажать «игра»	Откроется игра с описанными параметрам и	Откроется игра с описанными параметрам и	Прошёл	Алексей Калёных 25.05.202 5

ЗАКЛЮЧЕНИЕ

Разработка мобильной игры «Дождь из слов» в рамках дипломного проекта позволила не только создать полезный и увлекательный продукт, но и углубить знания в области кроссплатформенной разработки, игрового дизайна и оптимизации пользовательского опыта.

Выбор технологий — .NET MAUI, C# и Visual Studio — полностью оправдал себя, обеспечив кроссплатформенность и удобство разработки. В ходе работы были успешно выполнены все ключевые этапы: от изучения фреймворка и проектирования архитектуры до тестирования и документирования. В результате получилось стабильное и функциональное приложение, сочетающее в себе интерактивный геймплей, образовательную ценность и интуитивно понятный интерфейс.

Практическая значимость проекта заключается в том, что игра способствует развитию когнитивных навыков у пользователей разных возрастов, а также демонстрирует эффективность выбранных технологий для создания современных мобильных приложений.

В перспективе проект может быть дополнен новыми игровыми режимами, системой достижений и мультиплеерными функциями, что повысит его вовлекающий потенциал. Таким образом, дипломная работа подтвердила свою актуальность, а полученный опыт открывает новые возможности для профессионального роста в сфере мобильной и игровой разработки.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Торсенссон, Й. Разработка игр на Unity. Руководство для начинающих [Текст] / Йонас Торсенссон. — М.: Издательский дом «ДМК Пресс», 2020. — 320с.
2. Капун, Дж. Unity в действии. Мультиплатформенная разработка игр на C# [Текст] / Джозеф Хокинс. — СПб.: Питер, 2019. — 512с.
3. Шарп, М. Начала программирования на C# [Текст]: Полное руководство / Майк Шарп. — СПб.: Питер, 2021. — 768с.
4. Албахари, Дж. C# 7.0. Карманный справочник. Скорая помощь для программистов на C# 7.0 [Текст]: Практическое руководство / Дж. Албахари, Б. Албахари. — СПб.: Издательство «Альфа-Книга», 2019. — 224с.
5. О'Нейл, П. Основы программирования на C# [Текст]: Практическое руководство / Пол О'Нейл. — М.: Вильямс, 2018. — 640с.
6. Хилман, Б. Разработка игр на Unity для начинающих [Текст]: Практическое руководство / Бен Хилман. — М.: ДМК Пресс, 2019. — 288с.
7. Майерс, Т. Unity и C# для профессионалов [Текст]: Полное руководство / Тим Майерс. — СПб.: Питер, 2021. — 464с.
8. ГОСТ 19.505-79. Единая система программной документации.
9. Руководство оператора. Требования к содержанию и оформлению [Текст].— Введ. 2002-07-01. -М.: ФГУП «СТАНДАРТИНФОРМ»У, 2022. – 27с.
10. ГОСТ 19.504-79. Единая система программной документации.
11. Руководство программиста. Требования к содержанию и оформлению [Текст]. – Введ. 2002-07-01. -М.: ФГУП «СТАНДАРТИНФОРМ»У, 2023. – 27с.
12. ГОСТ 19.201-78. Единая система программной документации.
13. Техническое задание. Требования к содержанию и оформлению [Текст]. Введ. 2002-07-01. -М.: ФГУП «СТАНДАРТИНФОРМ»У, 2018. – 27с.

ПРИЛОЖЕНИЕ А

Листинг главного меню

```
using Microsoft.Maui;
using Microsoft.Maui.ApplicationModel;
using Microsoft.Maui.Controls;
using System.Diagnostics;
using System.Numerics;

namespace rainwords;

public partial class Menu : ContentPage
{
    private readonly IAudioService _audioService;
    private MainPage page;
    private bool _isInitialized;
    private bool _musicda = true;
    public Menu(IAudioService audioService)
    {
        var stopwatch = Stopwatch.StartNew();
        InitializeComponent();
        _audioService = audioService;
        Task.Run(() =>
        {
            InitializeDefaults();
            InitializeAudio();
            InitializeUI();
        });
        back.Text = "<";
        stopwatch.Stop();
        Console.WriteLine($"Settings loaded in {stopwatch.ElapsedMilliseconds} ms");
        App.GamePaused += PauseMenu;
        App.GameResumed += ResumeMenu;
    }

    private void PauseMenu()
    {
        _audioService.StopMenuMusic();
        _musicda = false;
    }

    private void ResumeMenu()
    {
        if(!_musicda) _audioService.StartMenuMusic();
    }

    protected override void OnDisappearing()
    {
        base.OnDisappearing();
        // Отписка от событий при уходе со страницы
        App.GamePaused -= PauseMenu;
    }
}
```

```

        App.GameResumed -= ResumeMenu;
    }

    private void InitializeDefaults()
    {
        var currentTheme = Preferences.Default.Get("selthemedate", "");
        if (string.IsNullOrEmpty(currentTheme))
        {
            Preferences.Default.Set("selthemedate", "swhitetheme.png");
            Preferences.Default.Set("languagepickcheck", "РУССКИЙ");
            Preferences.Default.Set("sweff", true);
            Preferences.Default.Set("swanim", true);
        }
    }

    private async Task InitializeAudio()
    {
        var audio = Preferences.Default.Get("swsongs", true);
        if (audio)
        {
            await _audioService.InitializeAsync();
            _audioService.PlayMenuMusic();
        }
        _isInitialized = true;
    }

    private void InitializeUI()
    {
        Dispatcher.Dispatch(() =>
        {
            var language = Preferences.Default.Get("languagepickcheck", "");
            if (language == "ENGLISH")
            {
                UpdateTextToEnglish();
            }

            ApplyTheme();
            btnlist.IsVisible = true;
        });
    }

    private void UpdateTextToEnglish()
    {
        play.Text = "Play";
        setting.Text = "Setting";
        exit.Text = "Exit";
        confirmationexitone.Text = "Definitely get out?";
        confirmationexittwo.Text = "Your game will not be saved!";
        exitconf.Text = "Yes";
        non.Text = "No";
        selectcomplex.Text = "DIFFICULTY";
        contin.Text = "Continue";
        easy.Text = "Easy";
    }

```

```

        average.Text = "Average";
        hard.Text = "Hard";
        custom.Text = "Custom";
    }
    private void ApplyTheme()
    {
        var theme = Preferences.Default.Get("selthemedate", "");
        var buttons = btnlist.Children.OfType<Button>()
            .Concat(complex1.Children.OfType<Button>())
            .Concat(new[] { exitconf, non });

        string buttonStyleKey = "", labelStyleKey = "";
        string backgroundColor = "bgwhite.jpeg";
        Color BackInfor = Colors.White;
        Color StrokeInfo = Colors.Black;
        switch (theme)
        {
            case "swhitetheme.png":
                buttonStyleKey = "whitethemebutton";
                labelStyleKey = "whitethemelabel";
                backgroundColor = "bgwhite.jpeg";
                break;
            case "spinktheme.png":
                buttonStyleKey = "pinkthemebutton";
                labelStyleKey = "pinkthemelabel";
                backgroundColor = "bgpink.jpeg";
                BackInfor = Colors.HotPink;
                break;
            case "sblacktheme.png":
                buttonStyleKey = "blackthemebutton";
                labelStyleKey = "blackthemelabel";
                backgroundColor = "bgblack.jpeg";
                BackInfor = Colors.Black;
                StrokeInfo = Colors.White;
                break;
        }

        if (!string.IsNullOrEmpty(buttonStyleKey))
        {
            bg.Source = backgroundColor;
            var buttonStyle = (Style)Resources[buttonStyleKey];
            var labelStyle = (Style)Resources[labelStyleKey];
            confirmation.BackgroundColor = BackInfor;
            iamstroke.Stroke = StrokeInfo;
            foreach (var button in buttons)
            {
                button.Style = buttonStyle;
            }
            back.Style = labelStyle;
            confirmationexitone.Style = labelStyle;
            confirmationexittwo.Style = labelStyle;
            selectcomplex.Style = labelStyle;
        }
    }
}

```

```

    }
}
private async Task StartGame()
{
    if (!_isInitialized) return;

    if (Preferences.Default.Get("swsongs", true))
    {
        _audioService.PlayGameMusic();
    }

    page = new MainPage(_audioService);
    await Navigation.PushModalAsync(page, animated: false);
}
private async void Play_Clicked(object sender, EventArgs e)
{
    back.IsVisible = true;
    btnlist.IsVisible = false;
    btnlist.IsEnabled = false;
    complex1.IsVisible = true;
    complex1.IsEnabled = true;
    ApplyTheme();
}

private async void Setting_Clicked(object sender, EventArgs e)
{
    btnlist.IsEnabled = false;
    await Navigation.PushModalAsync(new Settings(_audioService), animated: false);
}

private void Exit_Clicked(object sender, EventArgs e)
{
    if (contin.IsVisible)
    {
        play.IsEnabled = false;
        setting.IsEnabled = false;
        exit.IsEnabled = false;
        confirmation.IsVisible = true;
    }
    else
    {
        Application.Current.Quit();
    }
}
private void exitconf_Clicked(object sender, EventArgs e) => Application.Current.Quit();

private void cancel(object sender, EventArgs e)
{
    play.IsEnabled = true;
    setting.IsEnabled = true;
}

```

```

        exit.IsEnabled = true;
        confirmation.IsVisible = false;
    }

    async private void buttoncomplex(object sender, EventArgs e)
    {
        int complexmenu;
        var button = sender as Button;
        btnlist.IsEnabled = complex1.IsEnabled = false;
        switch (button.CommandParameter)
        {
            case "playnext":
                back.IsVisible = false;
                Data.musplay = true;
                await Navigation.PushModalAsync(page, animated: false);
                break;
            case "playeasy":
                Data.musplay = true;
                complex1.IsEnabled = false;
                back.IsVisible = false;
                complexmenu = 0;
                Data.compl = complexmenu;
                Data.timecsm = 5;
                Data.speedcsm = 10000;
                Data.pointcsm = 20;
                await StartGame();
                //page = new MainPage(_audioService);
                //await Navigation.PushModalAsync(page);
                contin.IsVisible = true;
                break;
            case "playaverage":
                back.IsVisible = false;
                Data.musplay = true;
                complex1.IsEnabled = false;
                complexmenu = 1;
                Data.compl = complexmenu;
                Data.timecsm = 3;
                Data.speedcsm = 10000;
                Data.pointcsm = 30;
                await StartGame();
                //page = new MainPage(_audioService);
                //await Navigation.PushModalAsync(page);
                contin.IsVisible = true;
                break;
            case "playhard":
                Data.musplay = true;
                complex1.IsEnabled = false;
                back.IsVisible = false;
                complexmenu = 2;
                Data.compl = complexmenu;
                Data.timecsm = 2;
                Data.speedcsm = 10000;

```

```

        Data.pointscsm = 50;
        await StartGame();
        //page = new MainPage();
        //await Navigation.PushModalAsync(page);
        break;
    case "playcustom":
        back.IsVisible = false;
        CustomBuild f = new CustomBuild(_audioService);
        await Navigation.PushModalAsync(f);
        break;
    default:
        break;
}

complex1.IsVisible = false;
complex1.IsEnabled = false;

btnlist.IsVisible = true;
btnlist.IsEnabled = true;

}

private void back_menu(object sender, EventArgs e)
{
    back.IsVisible = false;
    complex1.IsVisible = false;
    complex1.IsEnabled = false;
    btnlist.IsVisible = true;
    btnlist.IsEnabled = true;
    ApplyTheme();
}
}

```

Листинг игры

```

using Microsoft.Maui;
using Microsoft.Maui.ApplicationModel;
using Microsoft.Maui.Controls;
using System.Diagnostics;
using System.Diagnostics.Metrics;
using System.Numerics;
using static System.Net.Mime.MediaTypeNames;
using Application = Microsoft.Maui.Controls.Application;

```

```

namespace rainwords
{

```

```

    public partial class MainPage : ContentPage
    {
        private readonly IAudioService _audioService;

```

```

private bool isManuallyPaused = true;
public MainPage(IAudioService audioService)
{

    InitializeComponent();
    _audioService = audioService;
    _audioService.IsMusicEnabled = false;
    CreatingLabels.Initialize(10);
    InitializeAudio();
    var language = Preferences.Default.Get("languagepickcheck", "");
    DrawKeyboard(language);
    timer_complex();
    minusone.Text = "<-";
    ApplyTheme();
    StartTimers();
    LanguageUI(language);
    App.GamePaused += PauseTimers;
}

private void PauseTimers()
{
    pause_Clicked(pause, EventArgs.Empty);
}

void LanguageUI(string language)
{
    if (language == "РУССКИЙ")
    {
        paus.Text = "ПАУЗА";
        continuebtn.Text = "ПРОДОЛЖИТЬ";
        exit.Text = "ВЫЙТИ";
        againbtn.Text = "ЗАНОВО";
    }
    else
    {
        paus.Text = "PAUSE";
        continuebtn.Text = "CONTINUE";
        againbtn.Text = "AGAIN";
        exit.Text = "EXIT";
    }
}

void DrawKeyboard(string language)
{
    if (language == "ENGLISH")

```

```

        DrawEnglishKeyboard();
    else if (language == "РУССКИЙ")
        DrawRussianKeyboard();
}

void DrawEnglishKeyboard()
{
    string[] row0 = "qwertyuiop".Select(c => c.ToString()).ToArray(); // 10 букв
    string[] row1 = "asdfghjkl".Select(c => c.ToString()).ToArray(); // 9 букв
    string[] row2 = "zxcvbnm".Select(c => c.ToString()).ToArray(); // 7 букв

    // Первая строка — QWERTYUIOP
    for (int i = 0; i < row0.Length; i++)
        AddKey(row0[i], 0, i * 2 + 2); // 1, 3, ..., 19

    // Вторая строка — ASDFGHJKL
    for (int i = 0; i < row1.Length; i++)
        AddKey(row1[i], 1, i * 2 + 3); // 2, 4, ..., 18

    // Третья строка — ZXCVBNM
    for (int i = 0; i < row2.Length; i++)
        AddKey(row2[i], 2, i * 2 + 5); // 4, 6, ..., 16
}

void DrawRussianKeyboard()
{
    string[] row0 = "йцукенгшщзх".Select(c => c.ToString()).ToArray(); // 11
    string[] row1 = "фывапроджэ".Select(c => c.ToString()).ToArray(); // 11
    string[] row2 = "ячсмитьбю".Select(c => c.ToString()).ToArray(); // 10

    for (int i = 0; i < row0.Length; i++)
        AddKey(row0[i], 0, i * 2 + 1); // Й–Х → 1, 3, ..., 21

    for (int i = 0; i < row1.Length; i++)
        AddKey(row1[i], 1, i * 2 + 1); // Ф–Э → 1, 3, ..., 21

    for (int i = 0; i < row2.Length; i++)
        AddKey(row2[i], 2, i * 2 + 3); // Я–Ю → 2, 4, ..., 20
}

void AddKey(string letter, int row, int column)
{
    var button = new Button
    {
        Text = letter,
        TextColor = Colors.Transparent,
        FontSize = 18,

```



```

        CornerRadius = 5,
        Style = GetTextColorByThemeButton(),
    };

    var label = new Label
    {
        Text = letter,
        Style = (Style)Resources["lbletter"],
        BackgroundColor = Colors.Transparent,
        //TextColor = GetTextColorByTheme(),
    };

    button.Clicked += KeyUp_Clicked;
    Grid.SetRow(button, row);
    Grid.SetColumn(button, column);
    Grid.SetColumnSpan(button, 2); // кнопка занимает 2 колонки
    Grid.SetRow(label, row);
    Grid.SetColumn(label, column);
    Grid.SetColumnSpan(label, 2);

    keyboard.Children.Add(button);
    keyboard.Children.Add(label);
}

```

```

void StartTimers()
{
    _timer = Application.Current.Dispatcher.CreateTimer();
    _timer.Interval = TimeSpan.FromSeconds(1);
    _timer.Tick += Timer_Tick;
    _timer3 = Application.Current.Dispatcher.CreateTimer();
    _timer3.Interval = TimeSpan.FromMilliseconds(1); // ~60 FPS
    _timer3.Tick += Timer3_Tick;
    _timer.Start();
    _timer3.Start();
}

```

```

private async void InitializeAudio()
{
    var audio = Preferences.Default.Get("swsongs", true);
    if (!audio) return;
    if (!_audioService.IsInitialized)
    {
        await _audioService.InitializeAsync();
    }
}

```

```

    }
    _audioService.PlayGameMusic();
}

protected override void OnAppearing()
{
    //проверяем что звук не отключен в настройках
    var audio = Preferences.Default.Get("swsongs", true);
    if (!audio) return; //если отключен не выполняем далее
    base.OnAppearing(); //вызов базовой реализации (страница включена)
    _audioService.PlayGameMusic(); //включаем игровую музыку
}

protected override void OnDisappearing()
{
    var audio = Preferences.Default.Get("swsongs", true); //проверяем что звук не отключен в
настройках

    if (!audio) return; //если отключен не выполняем далее
    base.OnDisappearing(); //вызывается, когда страница исчезает с экрана
    App.GamePaused -= PauseTimers; //отписываемся от события в App
                                                                    // Останавливаем музыку
    только если действительно уходим со страницы
    if (Navigation.NavigationStack.Count == 0 ||
        Navigation.NavigationStack.Last() is not MainPage)
    {
        _audioService.StopAllMusic(); //отключаем всю музыку
        _timer.Stop(); //остановка главного таймера
        _timer3.Stop(); //остановка игрового таймера
    }
}

private void OnPauseMusic()
{
    var audio = Preferences.Default.Get("swsongs", true);
    if (!audio) return;
    _audioService.PauseGameMusic();
}

private void OnResumeMusic()
{
    var audio = Preferences.Default.Get("swsongs", true);
    if (!audio) return;
    _audioService.ResumeGameMusic();
}

int cellindex = 0;

```

```

Random random = new Random();
int point = 0;

List<Label> labels = new List<Label>();

private void KeyUp_Clicked(object sender, EventArgs e)
{
    if (absmenu.IsVisible) return;
    var button = sender as Button;
    if (button == null) return;

    var letter = button.Text;
    if (cellindex < labels.Count)
    {
        labels[cellindex].Text = letter;
        cellindex++;
    }
}

}

(TimeSpan)_time;
IDispatcherTimer _timer;
IDispatcherTimer _timer3;
int complex = 0;
private int _remainingSeconds = 300;

private void Timer_Tick(object sender, EventArgs e)
{
    _time = _time.Add(new TimeSpan(0, 0, -1));

    string timeString = string.Format("{0}:{1:D2}", (int)_time.TotalMinutes, _time.Seconds);
    var language = Preferences.Default.Get("languagepickcheck", "");
    tim.Text = timeString;
    _remainingSeconds--;
    if (_time.TotalSeconds == 0)
    {
        foreach (var a in field.Children.ToList())
        {
            if (a is Label label1)
            {
                Microsoft.Maui.Controls.ViewExtensions.CancelAnimations(label1);
            }
        }
    }
}

```

```

        _timer.Stop();
        _timer3.Stop();
        paus.Text = language == "РУССКИЙ" ? "ИГРА ОКОНЧЕНА!" : "THE GAME IS
OVER!";

        paus.FontSize = 16;

        continuebtn.IsVisible = false;
        againbtn.IsVisible = true;
        absmenu.IsVisible = true;
    }
    if (!absmenu.IsVisible && _remainingSeconds % complex == 0)
    {
        CreateWord();
    }
}

private void CreateWord()
{
    var displayInfo = DeviceDisplay.MainDisplayInfo;
    var screenWidth = displayInfo.Width / displayInfo.Density;
    int randomword = random.Next(0, 70);
    randomX = random.Next(Convert.ToInt32(-screenWidth) + 250, Convert.ToInt32(screenWidth) -
250);

    var label = CreatingLabels.GetLabel(words[randomword], randomX);
    label.TextColor = GetTextColorByTheme();
    _audioService.PlayExplaSound();
    field.Children.Add(label);
    label.TranslateTo(randomX, 370, Data.speedcsm, Easing.Linear);
}

private Color GetTextColorByTheme()
{
    return Preferences.Default.Get("selthemedate", "") switch
    {
        "swhitetheme.png" => Colors.Black,
        "spinktheme.png" => Colors.Black,
        "sblacktheme.png" => Colors.White,
    };
}

private Style GetTextColorByThemeButton()
{
    return Preferences.Default.Get("selthemedate", "") switch
    {
        "swhitetheme.png" => (Style)Resources["whitethemebutton"],
        "spinktheme.png" => (Style)Resources["pinkthemebutton"],

```

```

        "sblacktheme.png" => (Style)Resources["blackthemebutton"],
    };
}

private void Timer3_Tick(object sender, EventArgs e)
{
    // Оптимизация: проверяем только видимые элементы
    var visibleLabels = field.Children.OfType<Label>().Where(l => l.IsVisible).ToList();

    // Оптимизация: кешируем wordwin
    string currentWordWin = "";
    switch (complextime)
    {
        case 0:
            currentWordWin = string.Concat(cell1.Text, cell2.Text, cell3.Text, cell4.Text,
cell5.Text);
            break;
        case 1:
            currentWordWin = string.Concat(cell1.Text, cell2.Text, cell3.Text, cell4.Text,
cell5.Text, cell6.Text, cell7.Text);
            break;
        case 2:
            currentWordWin = string.Concat(cell1.Text, cell2.Text, cell3.Text, cell4.Text,
cell5.Text, cell6.Text, cell7.Text, cell8.Text, cell9.Text);
            break;
    }

    // Оптимизация: проверяем только слова, которые могут совпадать по длине
    foreach (var child in visibleLabels.Where(l => l.Text.Length == currentWordWin.Length).ToList())
    {
        if (child.Text == currentWordWin)
        {
            PlayCorrectWordEffect(child, true);
            _audioService.PlayWinSound();
            point += Data.pointcsm;
            ClearInputCells();
            break;
        }
    }

    // Оптимизация: проверяем только элементы, которые достигли нижней границы
    foreach (var label in visibleLabels.Where(l => l.TranslationY >= 370).ToList())
    {
        PlayCorrectWordEffect(label, false);
        _audioService.PlayLossSound();
        if (point >= Data.pointcsm) point -= Data.pointcsm;
    }
}

```

```

    }

    poin.Text = point.ToString();
}
private void ClearInputCells()
{
    for (int i = 0; i < labels.Count; i++)
    {
        labels[i].Text = "";
    }
    cellindex = 0;
}
private async void PlayCorrectWordEffect(Label label, bool checkplay)
{
    var anim = Preferences.Default.Get("swanim", true);
    if (checkplay)
    {
        if (anim)
        {
            label.TextColor = Colors.Lime;
            Microsoft.Maui.Controls.ViewExtensions.CancelAnimations(label);
            await label.ScaleTo(1.3, 200, Easing.Linear);
            await Task.WhenAll(label.FadeTo(0, 300),
label.TranslateTo(label.TranslationX, label.TranslationY - 50, 300));
        }
        field.Children.Remove(label);
        CreatingLabels.ReleaseLabel(label);
    }
    else
    {
        if (anim)
        {
            label.TextColor = Colors.Red;
            Microsoft.Maui.Controls.ViewExtensions.CancelAnimations(label);
            await label.ScaleTo(1.3, 200, Easing.Linear);
            await Task.WhenAll(label.FadeTo(0, 300),
label.TranslateTo(label.TranslationX, label.TranslationY - 50, 300));
        }
        field.Children.Remove(label);
        CreatingLabels.ReleaseLabel(label);
    }
}
List<string> words = new List<string>();
int complextime = Data.compl;
double randomX;

void timer_complex()

```

```

{
    labels = new List<Label> { cell1, cell2, cell3, cell4, cell5, cell6, cell7, cell8, cell9 };
    var language = Preferences.Default.Get("languagepickcheck", "");
    lbent.Children.Clear();
    switch (complextime)
    {
        case 0:
            _time = new TimeSpan(00, Data.timecs, 00);
            complex = 5;

            labels.Remove(cell9);        labels.Remove(cell8);        labels.Remove(cell7);
labels.Remove(cell6);

            cell9.IsVisible = false; cell8.IsVisible = false; cell7.IsVisible = false;
cell6.IsVisible = false;

            if (language == "РУССКИЙ")
            {
                words = new List<string>

                { "много", "палец", "нитка", "башня", "бимок", "мышка", "бодро", "акулы", "алмаз", "ангел",

                "атака", "ведро", "вафля", "глядь", "герой", "двери", "драка", "ежели", "ехать", "нотка",

                "закат", "затея", "изгиб", "имена", "копья", "ковры", "листы", "бровь", "может", "питон",

                "точка", "ровно", "слеза", "сборы", "талия", "тепло", "суета", "утеха", "фобия", "фирма",

                "химия", "цапля", "цифра", "червь", "ткань", "шляпа", "салют", "ябеда", "кукла", "кость",

                "вилка", "тачка", "пачка", "чашка", "кошка", "файлы", "холод", "тепло", "осень", "весна",

                "вечно", "палец", "пятка", "шишка", "ветка", "мишка", "тесто", "кисло", "огонь", "пятно", };
            }
            else
            {
                words = new List<string> //2 строка 2 слово

                { "apple", "chair", "dance", "light", "smile", "ocean", "cloud", "peach", "happy", "grass",

                "berry", "breez", "mango", "lemon", "quilt", "music", "honey", "river", "dream", "books",

                "cocoa", "tulip", "clean", "plant", "bread", "sweet", "grape", "blush", "beach", "cream",

                "bloom", "chill", "merry", "maple", "petal", "fairy", "puppy", "jelly", "sunny", "hobby",

                "sunny", "grace", "angel", "flute", "leafy", "laugh", "piano", "donut", "linen", "soapy",

```

"green","creek","petal","smile","spark","chirp","climb","paint","dough","sheep",

"world","cozy","words","panda","toast","swirl","skate","froth","coral","coral",

};

}

break;

case 1:

_time = new TimeSpan(00, Data.timecs, 00);

complex = 4;

labels.Remove(cell9); labels.Remove(cell8);

cell9.IsVisible = false; cell8.IsVisible = false;

if (language == "РУССКИЙ")

{

words = new List<string>

{ "кипяток","телефон","наушник","стаканы","игрушка","заметки","самокат","охотник","покупка","капуста",

"морковь","картина","напиток","волосок","голосок","блокнот","галстук","ботаник","аксиома","алгебра",

"единица","дневник","диктант","деление","задание","корабль","яблочко","лесенка","занятия","мускулы",

"равнина","потолок","дверной","попинка","родинка","голубое","линейка","букварь","загадка","история",

"квадрат","отметка","площадь","предлог","предмет","процесс","рассказ","резинка","рисунок","рулетка",

"студент","теорема","тетрадь","учебник","ушастый","учитель","цилиндр","циркуль","частное","экзамен",

"планета","спутник","ледоход","носорог","паводок","сумерки","темнота","красный","конфеты","человек"},};

}

else

{

words = new List<string>

{

"blanket","popcorn","lantern","musical","bicycle","morning","rainbow","cupcake","jellybe","cottage",

"teacher","cookies","apricot","picture","imagine","flowers","breezes","journey","zephyrs","snuggle",

"almondy","dolphin","pajamas","balloon","drawing","cupcake","scribes","harvest","balance","humming",

"freedom","twinkle","sandals","daisyed","galaxy","blanket","unicorn","giggles","puddles","lantern",

"snuggle","eyelash","delight","bubbles","natural","harmony","bubbly","journey","painter","peaches",


```

"buttons", "smiling", "silence", "melody", "fashion", "cuddled", "peacock", "cascade", "whisper", "journal",

"popcorn", "tickles", "twinkle", "shimmer", "mellow", "chalked", "freedom", "sandals", "evening", "pastel",

    };

}

break;

case 2:

    _time = new TimeSpan(00, Data.timecs, 00);
    complex = 3;
    if (language == "РУССКИЙ")
    {

        words = new List<string>

{ "бриллиант", "диафрагма", "виновница", "визитница", "биография", "бизнесмен", "биосинтез", "гинеколог", "викторина", "география",

    "химчистка", "филология", "философия", "циферблат", "симметрия", "симуляция", "рисование", "киносеанс", "геометрия", "параллель",

    "лихорадка", "лицемерие", "мизантроп", "милостыня", "миллионер", "космонавт", "диспетчер", "дистанция", "живописец", "килограмм",

    "миссионер", "миниатюра", "бутерброд", "баскетбол", "курортный", "компьютер", "публичное", "кардиолог", "претензии", "именинник",

    "восемьсот", "должность", "знакомить", "пиратский", "подшипник", "безглазый", "непонятно", "осуждение", "последнее", "мальчишка",

    "прочность", "бесплатно", "фотосхема", "двадцатка", "затухание", "надбровый", "впрыску", "выставить", "пирожное", "драматизм",

    "брачность", "безгривый", "проведать", "подкрылье", "выморозка", "оцепление", "сметанник", "поисковик", "крепление", "спортсмен", };

    }

    else

    {

        words = new List<string>

{

    "adventure", "blueberry", "butterfly", "spaceship", "chocolate", "delighted", "apartment", "marshland", "jellybean", "evergreen",

    "harmonica", "rainstorm", "sunflower", "backdrops", "discovery", "feathered", "fireflies", "kangaroos", "lovelight", "landscape",

```

"classroom","evergreen","firelight","goldfinch","goldsmith","harmonica","honeycomb","jellybean","rainstorm","sandp
aper",

"moonlight","overjoyed","paintball","parklands","passenger","peacefuls","pinecones","playhouse","scrapbook","seashe
lls",

"snowflake","songbirds","sparkling","starfruit","stargazer","sunflower","sweetener","sweetshop","teacupful","whistlin
g",

"windchime","wristband","bluebirds","bumblebee","doughnuts","fairylike","fireflies","fireworks","flowerbed","goodni
ght",

"happiness","classroom","daydreams","earthworm","figurines","goldcrest","jellyroll","lakeshore","longboard","outerw
ear",

```

        };
    }
    break;
default:
    break;
}
for (int i = 0; i < labels.Count; i++)
{
    var label = new Label
    {
        Text = "_",
        Style = (Style)Resources["lbforent"],
        TextColor = GetTextColorByTheme()
    };
    labels[i].TextColor = GetTextColorByTheme();
    lbent.Children.Add(label);
}
}
private void ClearOne_Clicked(object sender, EventArgs e)
{
    if (absmenu.IsVisible) return;
    if (cellindex > 0)
    {
        cellindex--;
        labels[cellindex].Text = "";
    }
}
private void ClearAll_Clicked(object sender, EventArgs e)
{
    if (absmenu.IsVisible) return;
    cellindex = 0;
    for (int i = 0; i < labels.Count; i++)

```

```

        {
            labels[i].Text = "";
        }
    }

private void pause_Clicked(object sender, EventArgs e)
{
    if (absmenu.IsVisible) return;
    _timer.Stop();
    _timer3.Stop();
    OnPauseMusic();
    absmenu.IsVisible = true;

    foreach (var a in field.Children.ToList())
    {
        if (a is Label label1)
        {
            Microsoft.Maui.Controls.ViewExtensions.CancelAnimations(label1);
        }
    }
}

private void start_Clicked(object sender, EventArgs e)
{
    OnResumeMusic();
    foreach (var a in field.Children.OfType<Label>())
    {
        if (a is Label label1)
        {
            double remainingDistance = 370 - label1.TranslationY;
            uint newDuration = (uint)(10000 * (remainingDistance / (370 - (-100))));

            label1.TranslateTo(label1.TranslationX, 370, newDuration, Easing.Linear);
        }
    }

    _timer.Start();
    _timer3.Start();
    absmenu.IsVisible = false;
    clearone.IsEnabled = true;
    clear.IsEnabled = true;
}

private async void exmenu(object sender, EventArgs e)
{
    await Navigation.PopModalAsync(animated: false);
    _audioService.IsMusicEnabled = true;
    _audioService.PlayMenuMusic();
}

```

```

private void ApplyTheme()
{
    var theme = Preferences.Default.Get("selthemedate", "");

    switch (theme)
    {
        case "swhitetheme.png":
            ApplyWhiteTheme();
            break;
        case "spinktheme.png":
            ApplyPinkTheme();
            break;
        case "sblacktheme.png":
            ApplyBlackTheme();
            break;
    }
}

```

```

private void ApplyWhiteTheme()
{
    // Основные цвета
    main.BackgroundColor = Colors.White;
    coutscore.BackgroundColor = Colors.White;
    field.BackgroundColor = Colors.White;
    framefield.BorderColor = Colors.Black;
    // Текст
    tim.TextColor = Colors.Black;
    poin.TextColor = Colors.Black;
    paus.TextColor = Colors.Black;

    // Кнопки
    ApplyButtonStyle("whitethemebutton");

    // Метки слова
    foreach (var label in labels)
    {
        label.TextColor = Colors.Black;
    }

    // Метки ввода
    foreach (var label in lbent.Children.OfType<Label>())
    {
        label.TextColor = Colors.Black;
    }

    // Кнопки клавиатуры

```

```

        UpdateKeyboardButtons(Colors.White, Colors.Black);
    }

```

```

private void ApplyPinkTheme()
{
    main.BackgroundColor = Colors.HotPink;
    word.BackgroundColor = Colors.HotPink;
    coutscore.BackgroundColor = Colors.HotPink;
    field.BackgroundColor = Colors.HotPink;
    keyboard.BackgroundColor = Colors.HotPink;
    framefield.BorderColor = Colors.Black;
    tim.TextColor = Colors.Black;
    poin.TextColor = Colors.Black;
    paus.TextColor = Colors.Black;

    ApplyButtonStyle("pinkthemebutton");

    foreach (var label in labels)
    {
        label.TextColor = Colors.White;
    }

    foreach (var label in lbent.Children.OfType<Label>())
    {
        label.TextColor = Colors.Black;
    }

    UpdateKeyPinkButton(Colors.Black);
}

```

```

private void ApplyBlackTheme()
{
    main.BackgroundColor = Colors.Black;
    word.BackgroundColor = Colors.Black;
    coutscore.BackgroundColor = Colors.Black;
    field.BackgroundColor = Colors.Black;
    keyboard.BackgroundColor = Colors.Black;
    framefield.BorderColor = Colors.White;
    tim.TextColor = Colors.White;
    poin.TextColor = Colors.White;
    paus.TextColor = Colors.White;

    ApplyButtonStyle("blackthemebutton");

    foreach (var label in labels)
    {
        label.TextColor = Colors.White;
    }
}

```

```

    }

    foreach (var label in lbent.Children.OfType<Label>())
    {
        label.TextColor = Colors.White;
    }

    UpdateKeyboardButtons(Colors.Black, Colors.White);
}

private void ApplyButtonStyle(string styleKey)
{
    var style = (Style)Resources[styleKey];
    switch (styleKey)
    {
        case "whitethemebutton":
            continuebtn.Style = (Style)Resources["whitemenu"];
            againbtn.Style = (Style)Resources["whitemenu"];
            exit.Style = (Style)Resources["whitemenu"];
            break;
        case "pinkthemebutton":
            continuebtn.Style = (Style)Resources["pinkmenu"];
            againbtn.Style = (Style)Resources["pinkmenu"];
            exit.Style = (Style)Resources["pinkmenu"];
            break;
        case "blackthemebutton":
            continuebtn.Style = (Style)Resources["blackmenu"];
            againbtn.Style = (Style)Resources["blackmenu"];
            exit.Style = (Style)Resources["blackmenu"];
            break;
        default:
            break;
    }

    clearone.Style = style;
    clear.Style = style;
}

private void UpdateKeyboardButtons(Color bgColor, Color textColor)
{
    foreach (var view in keyboard.Children)
    {
        if (view is Button button)
        {
            button.BackgroundColor = bgColor;
        }
    }
}

```

```

        else if (view is Label label && label.Style == (Style)Resources["lbletter"])
        {
            label.TextColor = textColor;
        }
    }
}

private void UpdateKeyPinkButton(Color textColor)
{
    foreach (var view in keyboard.Children)
    {
        if (view is Button button)
        {
            button.BackgroundColor = Color.FromArgb("#D5156B");
        }
        else if (view is Label label && label.Style == (Style)Resources["lbletter"])
        {
            label.TextColor = textColor;
        }
    }
}

private void againbtn_Clicked(object sender, EventArgs e)
{
    absmenu.IsVisible = false;
    againbtn.IsVisible = false;
    continuebtn.IsVisible = true;
    paus.FontSize = 24;
    point = 0;
    _remainingSeconds = 300;
    var language = Preferences.Default.Get("languagepickcheck", "");
    LanguageUI(language);
    timer_complex();
    StartTimers();
}
}
}

```