

Список цитируемых источников

1. Баранов, Д. В. Современные информационные технологии. / Д. В. Баранов. — Томск : ИДО (ТУСУР), 2005. — 130 с.
2. Бухаркина, М. Ю. Мультимедийные учебник: что это? / М. Ю. Бухаркина // ИЯШ. — 2001. — №4. — с.13–16.
3. Инструктивно-методическое письмо «Об использовании информационных коммуникационных технологий в образовательном процессе с детьми с особенностями психофизического развития» // Специальная адукацыя. — 2016. — №4. — С. 44–51.
4. Демушкин, А. С. Компьютерные обучающие программы/ А.С. Демушкин // Информатика и образование. — 1995. — №3. — с. 18–21.

УДК: 004.4

Е. И. Дулько

Учреждение образования «Барановичский государственный университет», Барановичи, Республика Беларусь

Научный руководитель
А. И. Калько

РАЗРАБОТКА ИГРЫ «ПОИСК СЛОВ» С ПРИМЕНЕНИЕМ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПОДХОДА

Введение. В данной работе рассматривается разработка игры «Поиск слов» или же «Словомания», в которой используется объектно-ориентированный подход (ООП) для создания игрового приложения на языке программирования C#. Цель игры заключается в том, чтобы пользователь искал и выделял слова в матрице букв, которые могут быть расположены по вертикали, горизонтали и диагонали. В зависимости от уровня сложности игра предлагает различное количество слов и размер игрового поля [1]. Основные принципы ООП, такие как инкапсуляция, наследование и полиморфизм, позволяют легко расширять функционал игры и поддерживать чистоту архитектуры кода.

Основная часть. Графический интерфейс построен на базе Windows Forms. Приложение содержит несколько форм. Форма «Меню» изображена на рисунке 1.

Окно главного меню имеет следующий функционал:

1. Начать новую игру — сброс сохранения и начало новой игры с первого уровня.
2. Продолжить игру — продолжает игру с момента, когда пользователь вышел из игры.
3. Создать сервер — создает сервер, для игры с другим игроком по локальной сети.
4. Подключиться — подключает игрока к существующему серверу, для игры по локальной сети.

Основная форма «Игра» изображена на рисунке 2.



Рисунок 1 — Вид главного меню игры

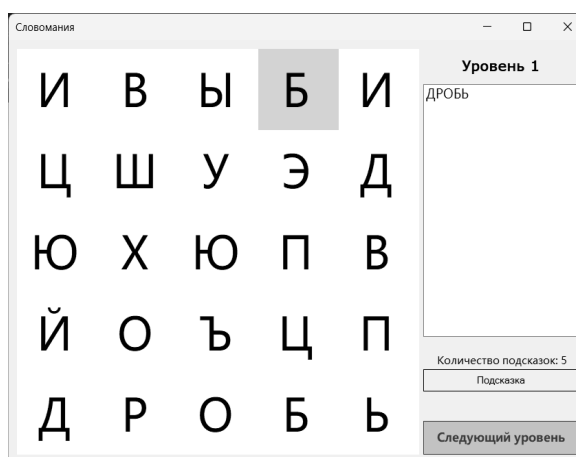


Рисунок 2 — Вид игровой формы

Окно игры содержит:

1. Матрицу букв — отображает матрицу букв, со спрятанными в ней словами.
2. Уровень — отображает текущий уровень сложности игры.
3. Список слов — отображает все спрятанные в матрице букв слова.
4. Подсказки — отображает количество подсказок для помощи поиска слов игроку.

5. Следующий уровень — кнопка перехода на следующий уровень, становится доступна после завершения текущего.

Форма «Создание сервера» изображена на рисунке 3.

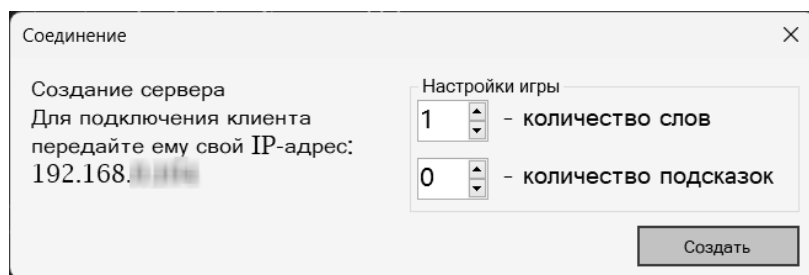


Рисунок 3 — Вид формы создания сервера для локальной игры

Окно «Создание сервера» имеет следующий функционал:

1. Строка информации — сообщает пользователю его IP-адрес для подключения.
2. Настройки — дает возможность выбрать количество слов и подсказок для начала локальной игры.
3. Создать — создает сервер, к которому может подключиться другой пользователь.

Форма «Подключение» изображена на рисунке 4.

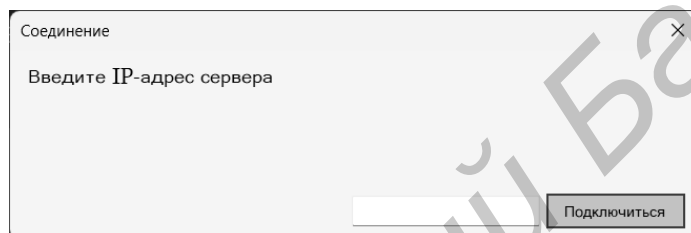


Рисунок 4 — Вид окна присоединения к игре

Окно подключения содержит:

1. Строка информации — указывает, удалось ли связаться с сервером.
2. Строка ввода — строка ввода IP-адреса сервера.
3. Подключение — соединяет игрока с сервером и начинает игру.

Описание основных аспектов работы приложения [2]:

1. Хранение списка слов, которые используются в игре. Слова добавляются из файла.
2. Различные уровни сложности. В зависимости от пройденного уровня, матрица букв будет увеличиваться, а количество слов для поиска изменяться.
3. Генерация матрицы букв. Игровое поле создается динамически в виде квадратной матрицы, в которую случайным образом помещаются слова. Пользователь может искать слова с помощью мыши, выделяя их на игровом поле.
4. Проверка правильности найденного слова. Если слово правильно выделено, оно выделяется зеленым цветом и автоматически вычеркивается из списка, предоставленного игроку.
5. Сохранение состояния игры. Прогресс игры постоянно сохраняется, и пользователь может продолжить игру позже. Эта функция реализована с использованием сериализации игровых данных.
6. Подсказки. Пользователь может воспользоваться подсказкой, которая обведет первую букву слова, выбранного пользователем в списке.
7. Мультиплеерная игра. Игра поддерживает сетевую игру, где один из игроков может создать сервер, а другой игрок подключится к нему через локальную сеть. В сетевой игре игроки соревнуются на скорость нахождения слов, кто первым найдет все слова — тот и выиграл. Для передачи данных между клиентом и сервером используется протокол TCP/IP, а данные передаются в формате JSON. Пример мультиплеерной игры по локальной сети на одном компьютере изображен на рисунке 5.

Приложение разработано с использованием ООП, что позволило создать гибкую структуру кода. Основные классы включают:

1. Form1 — Главное окно игры (Windows Forms). Управляет интерфейсом, обработкой выделения слов, переходом между уровнями, сохранением игры, загрузкой и сетевым взаимодействием.
2. Game — Основная логика игры. Управляет уровнями, запуском и сетевой игрой. Отвечает за сохранение прогресса и взаимодействие в многопользовательском режиме.
3. FrontEnd — Отображение интерфейса. Управляет отрисовкой матрицы, списка слов, подсказками и реакцией на действия пользователя (наведение, выделение слов).

4. FileManager — Управление файлами. Загружает/сохраняет слова и состояние игры. Реализует сериализацию для сохранений.
5. Level — Логика уровня. Управляет подсказками и проверяет правильность выделенных слов.
6. Matrix — Создание и размещение слов в матрице. Отвечает за генерацию игрового поля и расстановку букв.
7. Word — Представляет каждое слово в игре и хранит его местоположение на игровом поле.
8. Server и Client — Логика сетевой игры. Организуют взаимодействие между игроками через TCP/IP. Обмен данными происходит с помощью JSON сериализации.
9. Helper — Вспомогательные методы: генерация случайных чисел, перемешивание слов и др.

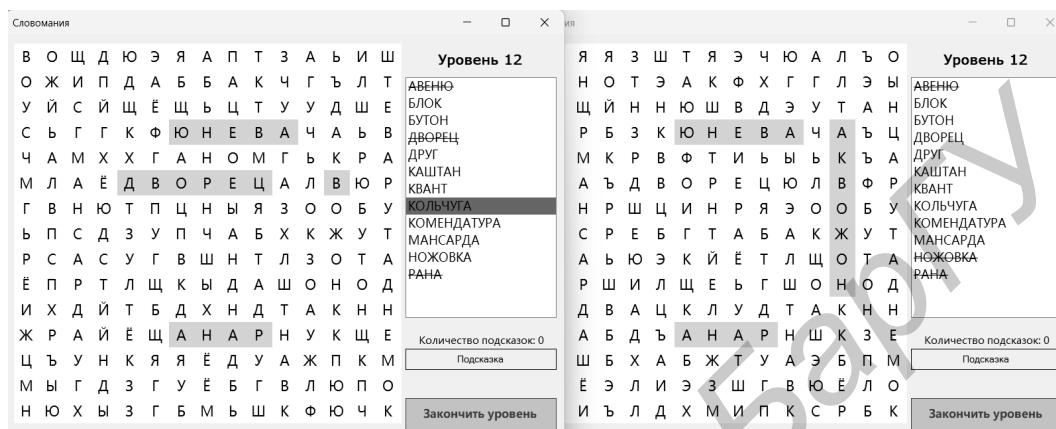


Рисунок 5 — Пример игры друг против друга по локальной сети на одном компьютере

Заключение. Разработка игры «Поиск слов» с применением объектно-ориентированного подхода демонстрирует преимущества модульного программирования. Благодаря инкапсуляции и абстракции, каждый аспект игрового процесса — от генерации матрицы букв до сетевого взаимодействия — реализован как отдельный функциональный блок. Это позволило сделать игру гибкой, масштабируемой и легкой для поддержки.

Использование TCP/IP для мультиплеерного режима и сериализация данных для сохранения игрового процесса обеспечивают пользователю возможность непрерывного игрового опыта, даже в случае сетевых игр. Результаты тестирования показали стабильность работы приложения при разных уровнях сложности и в мультиплеерном режиме.

В перспективе данную игру можно расширить, добавив больше режимов игры, улучшив сетевую архитектуру, а также реализовав поддержку других языков для создания словарей.

Список цитируемых источников

1. Создание приложения Windows Forms на C# в Visual Studio. : [сайт]. — URL: <https://learn.microsoft.com/ru-ru/visualstudio/ide/create-csharp-winform-visual-studio?view=vs-2022> (дата обращения: 10.09.2024).
2. Максимович, М. В. Алгоритм приложения «Кофемашина» с помощью JAVA / М. В. Максимович, А. И. Калько // Наука – практика : материалы II Междунар. науч.-практ. конф., Барановичи, 13 мая 2021 г. : в 3 ч. / М-во образования Респ. Беларусь, Баранович. гос. ун-т, редкол.: В. В. Климух (гл. ред.) [и др.]. — Барановичи : БарГУ, 2021. — Ч. 2. — С. 64–65.

УДК 004.94

Д. А. Литвинко

Учреждение образования «Барановичский государственный университет», Барановичи, Республика Беларусь

Научный руководитель
А. В. Шах

РАЗРАБОТКА 3-D МАКЕТА УНИВЕРСИТЕТА

Введение. Разработка 3D макета университета является важным этапом в современном проектировании. Такой макет позволяет визуализировать архитектурные и дизайнерские концепции, представить будущий вид здания. 3D модель университета может быть использована как инструмент для принятия решений,