**2 Проектирование задачи**

**2.1 Организация данных**

Приложение будет представлять собой интерактивную обучающую программу для детей, включающую несколько мини-игр и функций. В основе кода будет лежать объектно-ориентированный подход (ООП), что обеспечит четкое разделение логики и интерфейса.

Основные классы программы включат MainApp, который будет управлять навигацией между окнами, LoadingScreen с анимированным прогресс-баром, FindMatchesGame для игры на поиск парных карточек, WordGame для угадывания слов по изображениям, и Skazki, отвечающий за воспроизведение аудиосказок.

Необходимым условием при разработке данного приложения является описание организации данных, т.е. логическая и физическая структура данных в контексте среды разработки. Данные в приложении будут организованы следующим образом: изображения и звуки загружаются из файлов форматов .png, .jpg и .ogg и хранятся в объектах sf::Texture, sf::Sprite и sf::SoundBuffer. Тексты сказок и слова для игр будут загружаться из .txt-файлов и храниться в структурах std::vector. Игровые состояния, такие как текущий уровень, количество найденных пар и настройки, сохраняются в переменных соответствующих классов.

Для динамического управления данными будут использоваться стандартные контейнеры C++: std::vector для хранения карточек в FindMatchesGame и слов в WordGame, а также std::string для работы с текстовой информацией.

Также в приложении активно используются структуры и вспомогательные классы для хранения состояния интерфейса: позиции кнопок, их состояние (наведение, нажатие), обработчики событий реализованы через std::function. Для хранения текущих игровых параметров применяются логические флаги (например, isFlipped, isMatched, showCongratulations) и счётчики (winCounter, matchedPairs, attempts), что позволяет гибко управлять внутренней логикой. Программа не требует подключения внешних баз данных — все данные динамически загружаются из локальных ресурсов. Для генерации случайных элементов в играх используются генераторы псевдослучайных чисел (std::random\_device, std::mt19937) и алгоритмы перемешивания (std::shuffle), что обеспечивает уникальный опыт в каждой сессии.

**2.2 Процессы (основные функции приложения)**

Согласно всем перечисленным требованиям и указаниям, которые были рассмотрены в разделе «Анализ задачи», было определено, чем конкретно должна заниматься разрабатываемая курсовая программа. При запуске программы пользователь увидит экран загрузки. Он включит в себя прогресс-бар с числами, отображающими процесс загрузки ресурсов и фоновое изображение. После завершения загрузки автоматически будет открываться главный экран.

Главное меню будет предоставлять доступ ко всем разделам приложения. Пользователь сможет перейти в раздел "Сказки" для прослушивания сказок, запустить игру "Найди совпадения" или "Словарик". Дополнительно доступны настройки, где можно изменить фоновое изображение или ознакомиться со справкой и выключить фоновую музыку.

Игровой процесс в игре «Найди совпадения» будет построен на поиске парных карточек. В начале каждого уровня карточки будут перемешиваться, и пользователь будет открывать их по две. Если карточки совпадают, они остаются открытыми, в противном случае — переворачиваются обратно. При успешном завершении уровня игра будет предлагать перейти на следующий с помощью кнопки Enter или завершить игру с помощью закрытия игрового окна. Первый уровень представляет собой 10 карточек, из которых надо найти 5 пар, а второй уровень 12 карт, где уже необходимо найти 6 пар.

В игре «Словарик» пользователю будет показываться изображение объекта, и он должен будет составить его название из предложенных букв. При правильном ответе слово меняет цвет текста на зеленый и загружается следующее слово, а при неправильном – буквы подсвечиваются красным до тех пор, пока не будет введено верное слово. После пяти и десяти угаданных слов выводятся анимации победы, мотивирующие ребенка продолжать игру. Также на окне с игрой будет присутствовать кнопка «Пропустить», нажав на которую можно пропустить слово, перейдя к следующему.

Раздел «Сказки» включает в себя кнопки для проигрывания аудио-сказок. Пользователь сможет прослушать сказку из файла, который открывается при нажатии на кнопку с нужной сказкой.

Дополнительно предусмотрены процедуры для отображения справочной информации и сведений об авторе. При нажатии кнопки «Справка» запускается внешний .chm-файл или, при его отсутствии, открывается встроенное окно с текстовым содержимым. В процессе работы возможно управление фоновым музыкальным сопровождением: пользователь может включить или приостановить музыку. Также реализована система настройки внешнего вида главного окна: пользователь выбирает фон из заранее определённого набора, каждый из которых оформлен в стиле, подходящем для детской аудитории (пчела, осьминог, овца, лягушка и др.).

Отдельно стоит отметить, что структура кода учитывает обработку пользовательских событий в реальном времени, включая движения мыши, клики, нажатия клавиш (в том числе R, Enter, Escape), что обеспечивает интуитивное управление без дополнительных обучающих шагов.

**2.3 Описание внешнего пользовательского интерфейса**

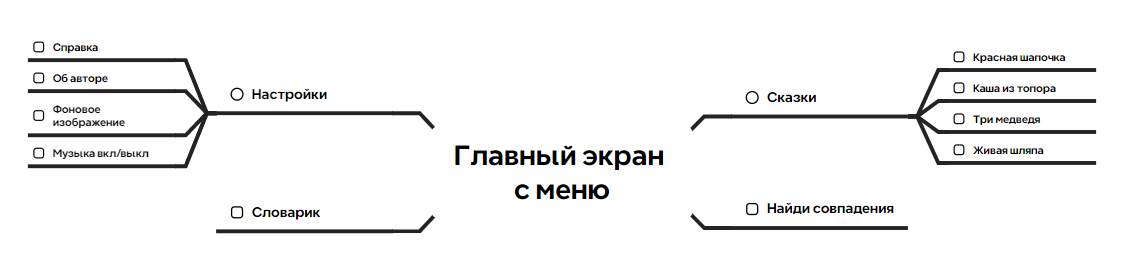
Важнейшим аспектом разработки данного курсового проекта является организация эффективного взаимодействия между пользователем и программой. Качество этого взаимодействия напрямую зависит от продуманности интерфейсных решений, выбора компонентов и степени автоматизации ключевых процессов. Поскольку программа разрабатывается в основном для детей, значит интерфейс должен быть интуитивно понятным для данного сегмента пользователей. Дизайн будет выполнен в ярких, привлекательных для детей тонах. Интерфейс главного меню будет представлять собой централизованный узел управления, откуда осуществляется доступ ко всем разделам приложения с помощью кнопок с соответствующими названиями.

рис. 2.1 − Система меню в графическом виде

Каждая мини-игра имеет четко структурированный интерфейс с логичным расположением управляющих элементов и информационных блоков.

Компоненты интерфейса размещаются с учётом правил визуальной иерархии — наиболее важные элементы (картинка и кнопки выбора режима) должны находиться в центре экрана, а вспомогательные элементы (настройки, пропустить слово, счётчик успехов) — на краях, но должны оставаться легко доступными. Каждая кнопка оформлена с помощью класса Button, который реагирует на наведение (анимация «подпрыгивания») и нажатие. При этом доступность управляемых элементов сохраняется даже при небольшом экране. Особое внимание уделено анимациям и визуальной обратной связи: плавные переходы в прогресс-баре загрузки, мигание победных надписей, затемнение фона при достижении цели — всё это делает взаимодействие эмоционально насыщенным и понятным даже для маленького ребёнка.

Визуальное оформление программы будет разрабатываться с учетом особенностей целевой аудитории - детей дошкольного и младшего школьного возраста. Будет подобрана яркая, но не раздражающая цветовая гамма, способствующая концентрации внимания. Все текстовые элементы используют крупные, хорошо читаемые шрифты с закругленными формами, что делает интерфейс дружелюбным и доступным для восприятия.