

**Содержание**

Введение…………………….……………………………………………………………………….3

1. Теоретические основы разработки проекта «Заряди мозг»…………………………………………………………………………………………………3

1.1 **Анализ предметной области**………………………………………….................................…...3

1.2 Анализ существующих решений …………………………….………………………………..6

2 Моделирование проекта «Заряди мозг»…………………………..……………………………10

2.1 Разработка концепции проекта…………………….........…………………………………….10

2.2 Определение требований с ИС........................………………………………………………..14

2.3 Модели реализации проекта.....……………………………………………………………….22

Заключение…………………………………………………………………………………………23

Приложение………………………………………………………………………………………...26

Источники…………………………………………………………………………………………..44

**Введение**

Когнитивные нарушения представляют собой одну из наиболее актуальных и сложных проблем современного общества. Эти нарушения могут возникать в результате различных факторов, таких как травмы головы, инсульты, нейродегенеративные заболевания и возрастные изменения. В данной работе мы рассмотрим актуальность проблемы, цели и задачи проекта, а также ожидаемые результаты и методы реализации.

**1. Теоретические основы разработки проекта «Заряди мозг»**

**1.1 Анализ предметной области**

Когнитивная реабилитация представляет собой процесс, направленный на восстановление или компенсацию когнитивных функций, таких как память, внимание, мышление, речь, восприятие и исполнительные функции. Эти функции могут быть нарушены в результате различных факторов, включая травмы головы, инсульты, нейродегенеративные заболевания (например, болезнь Альцгеймера) и возрастные изменения. Когнитивная реабилитация включает в себя использование различных методов и подходов, таких как тренировки памяти, упражнения на внимание и логическое мышление, а также применение современных технологий для улучшения результатов.  
  
Современные исследования показывают, что когнитивная реабилитация может значительно улучшить качество жизни пациентов, а также замедлить прогрессирование заболеваний. Важно отметить, что реабилитация должна быть индивидуализированной и учитывать уникальные потребности каждого пациента.  
  
**Мобильное приложение**  
  
Мобильное приложение — это программное обеспечение, разработанное для работы на мобильных устройствах, таких как смартфоны и планшеты. С ростом распространенности мобильных технологий и увеличением числа пользователей мобильных устройств, приложения становятся все более популярными в различных областях, включая здравоохранение и реабилитацию.   
  
Использование мобильных приложений в когнитивной реабилитации открывает новые возможности для пациентов. Они могут заниматься реабилитацией в удобное для них время и месте, что делает процесс более гибким и доступным. Кроме того, мобильные приложения могут обеспечить постоянный мониторинг прогресса и предоставлять пользователям обратную связь.  
  
**Игры**  
  
Игры представляют собой интерактивные приложения, в которых пользователю предлагаются различные задачи или испытания, имеющие развлекательный или обучающий характер. Игровая форма обучения и тренировки становится все более популярной благодаря своей способности удерживать внимание пользователей и создавать увлекательный опыт.  
  
В контексте когнитивной реабилитации игры могут быть использованы для стимуляции и тренировки когнитивных функций. Они могут включать в себя различные механики — от простых задач на запоминание до сложных логических головоломок. Игры могут быть адаптированы под уровень пользователя, что делает их подходящими как для людей с серьезными когнитивными нарушениями, так и для тех, кто просто хочет улучшить свои навыки.  
  
**Сочетание**  
  
Сочетание мобильного приложения с играми для когнитивной реабилитации представляет собой мощный инструмент, который использует игровые механики для стимуляции и тренировки когнитивных функций у людей, нуждающихся в реабилитации. Это решение позволяет не только развивать необходимые навыки, но и делает процесс обучения более увлекательным и интересным.

**Цели и задачи приложения**  
  
**Основные цели:**  
  
Основные цели нашего приложения заключаются в следующем:  
  
• **Улучшение когнитивных функций:** Приложение будет направлено на повышение уровня внимания, памяти, мышления, восприятия и исполнительных функций у пользователей.  
  
• **Восстановление нарушенных функций:** Основной задачей является компенсация когнитивных нарушений, вызванных травмами или заболеваниями.  
  
• **Мотивация и вовлечение:** Обеспечение увлекательного и мотивирующего процесса реабилитации через игровые механики и систему вознаграждений.  
  
**Задачи приложения:**  
  
Для достижения поставленных целей необходимо решить следующие задачи:  
  
• **Предоставление набора игр:** Разработка разнообразных игр, направленных на тренировку различных когнитивных функций. Игры должны охватывать широкий спектр навыков и уровней сложности.  
  
• **Адаптация уровня сложности:** Настройка сложности игр в соответствии с уровнем пользователя для обеспечения оптимального процесса обучения.  
  
• **Отслеживание прогресса:** Внедрение системы мониторинга успехов пользователя с предоставлением обратной связи о достигнутых результатах.

• **Удобный интерфейс:** Разработка простого и интуитивно понятного интерфейса, чтобы пользователи могли легко взаимодействовать с приложением.  
  
• **Персонализация:** Возможность настройки приложения под индивидуальные нужды пользователя для повышения эффективности реабилитации.  
  
• **Доступность:** Обеспечение доступности приложения для разных пользователей, включая людей с особыми потребностями.  
  
• **Развитие:** Постоянное добавление новых игр и функциональных возможностей для поддержания интереса пользователей.  
  
**Участники предметной области**  
  
**Пользователи:**  
  
Пользователи приложения могут быть разнообразными:  
  
• **Люди с когнитивными нарушениями:** Это могут быть пациенты после травм, инсультов или страдающие нейродегенеративными заболеваниями. Они нуждаются в специализированной реабилитации для восстановления утраченных функций.  
  
• **Люди, желающие улучшить свои когнитивные функции:** Это может включать студентов или профессионалов, стремящихся повысить свою продуктивность и эффективность.  
  
• **Дети с задержками развития:** Данная категория пользователей также может извлечь пользу из игр, направленных на развитие когнитивных навыков. ,

**1.2 Анализ существующих решений (достоинства, недостатки и отличительные особенности разрабатываемой ИС)**

**Достоинства существующих решений**  
  
1. **Эффективность в тренировке когнитивных функций**  
Существующие приложения для когнитивной реабилитации доказали свою эффективность в улучшении различных когнитивных функций, таких как память, внимание, логическое мышление и исполнительные функции. Многочисленные исследования подтверждают, что регулярные тренировки с использованием этих приложений могут привести к значительным улучшениям в повседневной жизни пользователей. Это делает такие решения привлекательными для пациентов и специалистов.  
  
2. **Персонализированный подход к тренировкам**  
Многие из существующих приложений предлагают персонализированные планы тренировок, которые адаптируются к индивидуальным потребностям пользователя. Это позволяет учитывать уровень когнитивных нарушений, предпочтения и цели пользователя. Таким образом, пользователи могут получать более целенаправленные и эффективные тренировки, что способствует лучшим результатам.  
  
3. **Удобный интерфейс и понятный геймплей**  
Большинство приложений разрабатываются с акцентом на удобство использования и интуитивно понятный интерфейс. Это особенно важно для пользователей старшего возраста или людей с ограниченными возможностями. Простота навигации и ясные инструкции помогают пользователям легко ориентироваться в приложении и сосредоточиться на тренировках.  
  
4. **Широкое распространение на рынке приложений**  
Существующие решения имеют широкую доступность, поскольку они доступны на популярных платформах (iOS, Android) и имеют большое количество загрузок. Это создает сообщество пользователей, которые могут делиться опытом и поддерживать друг друга, что дополнительно повышает мотивацию.  
  
**Недостатки существующих решений**  
  
1. **Ограниченность бесплатного функционала**  
Многие приложения предлагают ограниченный доступ к функционалу без оплаты. Пользователи часто сталкиваются с тем, что для получения доступа к более сложным играм или полному набору функций требуется подписка или разовая плата. Это может стать барьером для пользователей, желающих начать тренировки.  
  
2. **Высокая стоимость премиум-подписки**  
Высокая стоимость премиум-подписки может оттолкнуть потенциальных пользователей, особенно тех, кто не может позволить себе регулярные расходы на приложения. Это ограничивает доступность качественных реабилитационных ресурсов для людей с низким доходом или ограниченными финансовыми возможностями.  
  
3. **Недостаточная адаптация к специфическим потребностям пользователей**  
Несмотря на наличие персонализированных подходов, многие приложения не всегда учитывают индивидуальные особенности пользователей, такие как уровень прогресса, предпочтения в типах упражнений или конкретные когнитивные нарушения. Это может привести к недостаточной эффективности тренировок.  
  
4. **Не всегда очевидный прогресс пользователя**  
Некоторые приложения не предоставляют пользователям достаточной информации о их прогрессе или достигнутых результатах. Это может снижать мотивацию пользователей продолжать тренировки, так как они не видят конкретных улучшений своих когнитивных функций.  
  
5. **Отсутствие связи с медицинскими специалистами**  
В большинстве случаев существующие решения не обеспечивают интеграцию с медицинскими специалистами, что ограничивает возможности для комплексного подхода к реабилитации. Пользователи не могут получать рекомендации от врачей на основе данных о прогрессе и состоянии их когнитивных функций.  
  
**Отличительные особенности разрабатываемой ИС**  
  
1. **Интеграция с врачом**  
Одной из ключевых отличительных особенностей разрабатываемого приложения является возможность интеграции с медицинскими специалистами. Это позволит врачам отслеживать прогресс своих пациентов в режиме реального времени, корректировать план тренировок на основе полученных данных и предоставлять рекомендации по дальнейшей реабилитации. Такая связь между пользователями и специалистами обеспечит более комплексный подход к лечению и повысит эффективность реабилитации.

2. **Адаптивность**

Разрабатываемое приложение будет использовать улучшенный алгоритм адаптации сложности игр к потребностям пользователя. Это означает, что по мере прогресса пользователя уровень сложности будет автоматически подстраиваться под его способности и достижения. Такой подход позволит избежать скуки от слишком простых заданий или фрустрации от слишком сложных, что сделает процесс тренировок более комфортным и продуктивным.  
  
3. **Разнообразие игр**  
В отличие от многих существующих решений, наше приложение предложит широкий спектр тренировок, нацеленных на различные когнитивные функции. Это будут не только традиционные задачи на память и внимание, но и игры, развивающие логическое мышление, пространственное восприятие и другие навыки. Такое разнообразие поможет удерживать интерес пользователей и обеспечит всестороннее развитие когнитивных функций.  
  
4. **Бесплатный доступ**  
Мы планируем предоставить базовый функционал приложения бесплатно или по более доступной цене по сравнению с конкурентами. Это сделает приложение более доступным для широкой аудитории пользователей, включая тех, кто может испытывать финансовые трудности. Дополнительные функции могут быть доступны через разумную подписку, что позволит пользователям выбирать необходимые им опции без значительных затрат.  
  
5. **Мотивация**  
Разрабатываемое приложение будет включать улучшенную систему мотивации через достижения и вознаграждения. Пользователи смогут получать баллы за выполнение заданий, достигать новых уровней и открывать новые игры или функции по мере прогресса. Этот элемент геймификации поможет поддерживать интерес пользователей и побуждать их продолжать тренировки.  
  
6. **Интуитивно понятный интерфейс**  
Мы уделим особое внимание разработке простого и понятного интерфейса, который будет удобен для пользователей всех возрастов и уровней подготовки. Приложение будет включать четкие инструкции, визуальные подсказки и легкую навигацию, чтобы пользователи могли сосредоточиться на тренировках без лишних трудностей.

**2. Моделирование проекта «Заряди мозг»**

**2.1 Разработка Концепции Проекта**

**Целевая аудитория**  
Проект «Заряди мозг» нацелен на несколько ключевых групп пользователей:

1. **Пациенты с когнитивными нарушениями**:

Люди, страдающие от различных форм когнитивных нарушений, таких как деменция, болезнь Альцгеймера, последствия инсульта и другие неврологические расстройства. Эта группа нуждается в специализированных инструментах для реабилитации и тренировки когнитивных функций.  
  
2. **Родственники и опекуны пациентов**:

• Люди, ухаживающие за пациентами с когнитивными нарушениями, которые ищут эффективные способы поддержки и улучшения состояния своих близких. Они могут использовать приложение для мониторинга прогресса и выбора соответствующих упражнений.  
  
3. **Медицинские учреждения**:

• Больницы, реабилитационные центры и частные практики, которые могут интегрировать приложение в свои программы лечения. Врачи и терапевты смогут использовать приложение для отслеживания состояния пациентов и адаптации реабилитационных планов.  
  
**Описание функциональных возможностей приложения**  
Приложение «Заряди мозг» будет обладать множеством функциональных возможностей, направленных на улучшение когнитивных функций пользователей. Основные функции включают:  
  
**1. Простой и интуитивно понятный интерфейс**

**Простая навигация**:  
Интерфейс приложения будет разработан с акцентом на удобство использования. Пользователи смогут легко перемещаться между различными разделами, такими как тренировки, статистика и настройки. Главное меню будет содержать четко обозначенные категории, что позволит пользователям быстро находить нужные им игры и упражнения.

**Привлекательный визуальный стиль**:  
Дизайн приложения будет современным и эстетически приятным, что создаст положительный пользовательский опыт. Использование ярких цветов, четкой типографики и интуитивно понятных иконок будет способствовать тому, чтобы пользователи чувствовали себя комфортно во время использования приложения.  
  
**2. Персонализированные тренировки**  
  
**Адаптация к уровню пользователя**:  
Приложение будет предлагать персонализированные тренировки на основе уровня когнитивных нарушений пользователя. В начале использования пользователи пройдут короткий тест для определения их текущего уровня навыков, после чего приложение предложит индивидуально подобранный план тренировок.  
  
**Разнообразие упражнений**:  
Приложение будет содержать широкий ассортимент игр и задач, направленных на развитие различных когнитивных функций: памяти, внимания, логического мышления и пространственного восприятия. Это позволит пользователям выбирать наиболее интересные для них упражнения и избегать скуки.  
  
**3. Мониторинг прогресса**  
  
**Статистика и отчеты**:  
Пользователи смогут отслеживать свой прогресс через встроенные статистические отчеты. Приложение будет отображать данные о времени, проведенном за тренировками, уровне сложности пройденных заданий и достигнутых результатах. Эти данные помогут как пользователям, так и их опекунам видеть динамику улучшений.  
  
**Цели и достижения**:  
В приложении будет реализована система целей и достижений, которая будет мотивировать пользователей продолжать тренировки. Пользователи смогут устанавливать свои собственные цели и получать награды за их достижение, что повысит уровень вовлеченности.  
  
**4. Интеграция с медицинскими специалистами**  
  
**Обмен данными с врачами**:  
Приложение позволит пользователям делиться своими данными о прогрессе с медицинскими специалистами. Врачи смогут получать доступ к статистике пациента, что поможет им адаптировать реабилитационный план и давать рекомендации на основе фактических данных.  
  
**Напоминания о тренировках**:  
Встроенная функция напоминаний поможет пользователям не забывать о регулярных тренировках. Пользователи смогут настраивать уведомления о времени занятий, что будет способствовать соблюдению режима.  
  
**5. Сообщество пользователей**  
  
**Функция общения**:

Приложение будет включать возможность общения между пользователями, что позволит им делиться опытом и советами. Это создаст поддержку среди пользователей и поможет им чувствовать себя менее одинокими в процессе реабилитации.  
  
**Форумы и обсуждения**:  
В рамках приложения можно будет организовать форумы для обсуждения различных тем, связанных с когнитивной реабилитацией. Это предоставит пользователям возможность обмениваться полезной информацией и находить единомышленников.

**Технологическая платформа: Android Studio**  
Приложение «Заряди мозг» будет разрабатываться на платформе Android Studio, что обеспечит его доступность для широкой аудитории пользователей мобильных устройств на базе Android. Основные преимущества использования Android Studio:  
  
1. **Широкая аудитория**:  
Android является одной из самых распространенных мобильных платформ в мире, что обеспечит доступность приложения для большого числа пользователей.  
  
2. **Инструменты разработки**:  
Android Studio предлагает мощные инструменты для разработки, включая встроенные средства тестирования, отладки и профилирования производительности приложения. Это позволит разработчикам создавать качественное приложение с высокой производительностью.  
  
3. **Поддержка различных устройств**:  
Платформа позволяет адаптировать приложение под различные устройства с разными размерами экранов и характеристиками, что обеспечит удобство использования на смартфонах и планшетах.  
  
4. **Гибкость разработки**:  
Платформа поддерживает множество языков программирования (Java, Kotlin), что позволяет разработчикам выбирать наиболее подходящий инструмент для реализации функционала приложения.

**2.2 Определение требований к ИС**

**SWOT-Анализ**

**Силы**  
  
1. **Уникальные игровые механики, фокусирующиеся на конкретных когнитивных функциях**:  
Приложение будет включать специализированные игры и упражнения, направленные на тренировку различных когнитивных функций, таких как память, внимание и логическое мышление. Это позволит пользователям сосредоточиться на своих слабых местах и достичь более высоких результатов.  
  
2. **Персонализированные тренировки, адаптирующиеся к прогрессу пользователя**:  
Используя алгоритмы машинного обучения, приложение сможет адаптировать уровень сложности заданий в зависимости от прогресса пользователя. Это обеспечит постоянный вызов и мотивацию для пользователей.  
  
3. **Интеграция с медицинскими системами**:  
 Возможность передачи данных о прогрессе пользователя медицинским специалистам обеспечит более тесное сотрудничество между пациентами и врачами, что повысит качество лечения и реабилитации.  
  
4. **Привлекательный и интуитивно понятный интерфейс**:  
Дизайн приложения будет ориентирован на удобство использования, что особенно важно для пожилых пользователей или людей с низким уровнем технологической грамотности. Простота навигации и доступность функций сделают приложение более привлекательным.  
  
5. **Доступная цена**:  
Конкурентоспособная ценовая политика позволит привлечь большее количество пользователей, особенно среди тех, кто ищет доступные решения для когнитивной реабилитации.  
  
**Слабости**  
  
1. **Ограниченный охват рынка на начальном этапе**:  
 На старте проекта может быть сложно привлечь достаточное количество пользователей из-за недостатка известности приложения. Это потребует значительных усилий в маркетинге и продвижении.  
  
2. **Зависимость от качества и актуальности игрового контента**:  
Для обеспечения интереса пользователей необходимо постоянно обновлять контент и добавлять новые игры и упражнения. Это потребует ресурсов и времени на разработку.  
  
3. **Необходимость в постоянном обновлении и поддержке приложения**:  
Техническая поддержка, исправление ошибок и обновления функционала потребуют постоянного внимания со стороны команды разработчиков.  
  
4. **Конкуренция со стороны уже существующих приложений**:  
На рынке уже присутствуют множество приложений для когнитивной тренировки, что создает значительную конкуренцию и может затруднить привлечение пользователей.  
  
**Возможности**  
  
1. **Растущий спрос на технологии для когнитивной реабилитации**:

С увеличением числа людей с когнитивными нарушениями растет интерес к технологиям, которые могут помочь в их реабилитации. Это создает возможности для успешного внедрения приложения.  
  
2. **Расширение рынка мобильных приложений для здравоохранения**:  
Рынок мобильных приложений для здоровья активно развивается, что открывает новые возможности для внедрения инновационных решений.  
  
3. **Возможность сотрудничества с медицинскими учреждениями и специалистами**:

Партнерство с медицинскими учреждениями позволит не только протестировать приложение, но и получить рекомендации от врачей, что повысит доверие пользователей к продукту.  
  
4. **Возможность расширения функционала приложения (новые игры, функции, языки)**:  
Разработка новых функций и расширение языковой поддержки сделает приложение более привлекательным для международной аудитории и поможет охватить большее количество пользователей.  
  
**Угрозы**  
  
1. **Изменение законодательства в сфере здравоохранения и мобильных приложений**:  
Изменения в законодательстве могут повлиять на функционирование приложения и его соответствие требованиям, что создаст дополнительные риски для бизнеса.  
  
2. **Негативные отзывы пользователей или медицинских специалистов**:  
Плохие отзывы могут негативно сказаться на репутации приложения и снизить уровень доверия со стороны потенциальных пользователей.  
  
3. **Появление новых конкурентов с более совершенными приложениями**:

Конкуренция на рынке мобильных приложений для когнитивной реабилитации может привести к появлению новых игроков, которые предложат более инновационные решения.  
  
**Бизнес-модель:**  
  
**1. Сегменты клиентов**  
Проект «Заряди мозг» ориентирован на несколько ключевых сегментов клиентов:  
  
• **Пациенты с когнитивными нарушениями**:  
Разделение на подгруппы по типу когнитивного нарушения (легкая когнитивная недостаточность, деменция различной этиологии, последствия черепно-мозговых травм, рассеянный склероз и др.), возрасту (молодые, среднего возраста, пожилые), уровню технологической грамотности (высокий, средний, низкий). Это позволит таргетировать маркетинговые усилия и адаптировать интерфейс приложения.  
  
• **Медицинские учреждения**:  
Клиники, больницы, реабилитационные центры, частные медицинские практики, специализирующиеся на нейрореабилитации и лечении когнитивных расстройств. Разделение по размеру учреждения (крупные сети, небольшие клиники) и типу собственности (государственные, частные).  
  
• **Родственники и опекуны пациентов**:  
Подсегменты по степени вовлеченности в уход за пациентом, уровню технологической грамотности, географическому расположению (близость к пациенту).  
  
• **Некоммерческие организации**:  
Фонды, занимающиеся поддержкой пациентов с когнитивными нарушениями и их семей.  
  
**2. Ценностное предложение:**  
  
«Заряди мозг» — это персонализированное мобильное приложение для когнитивных тренировок, основанное на научно обоснованных методах и адаптированное к индивидуальным потребностям пациентов. В отличие от конкурентов наше приложение также предлагает чат с врачом и функцию напоминания о приеме таблеток и записи к врачу.  
  
• **Индивидуальный план тренировок**: Разработанный врачом с учетом диагноза, когнитивных способностей и индивидуальных целей пациента. Система автоматически корректирует сложность заданий в зависимости от прогресса.  
  
• **Широкий спектр когнитивных игр**: Направленных на тренировку памяти, внимания, речи, пространственного мышления и исполнительных функций. Игры разработаны с учетом принципов нейропластичности и доказали свою эффективность в исследованиях.  
  
• **Интеграция с медицинскими системами**: Возможность передачи данных о прогрессе пациента лечащему врачу (с согласия пациента).  
  
• **Функция напоминаний**: Напоминания о приеме лекарств, тренировках и консультациях.  
  
• **Мониторинг прогресса**: Визуализация прогресса пациента с помощью интерактивных графиков и отчетов.  
  
**3. Каналы сбыта**  
  
• **Прямые продажи**: Сотрудничество с медицинскими учреждениями, продажа подписок через приложение.  
  
• **Партнерские программы**: Сотрудничество с некоммерческими организациями, реклама в медицинских журналах и на специализированных веб-сайтах.  
  
• **Онлайн-маркетинг**: SEO-оптимизация сайта, контекстная реклама OpenAI Ads, таргетированная реклама в социальных сетях.  
  
**4. Ключевые партнеры:**  
  
• **Медицинские учреждения**: Для тестирования приложения и его дальнейшего внедрения.  
  
• **Научные институты и исследовательские центры**: Для проведения независимых исследований эффективности приложения и подтверждения его научной обоснованности.  
  
• **Разработчики контента**: Партнерство с экспертами в области нейропсихологии для создания актуального и качественного контента.

**Календарный план**   
  
**Фаза 1: Планирование и анализ (4 недели)**  
  
• **Определение требований к приложению**  
На первом этапе команда проекта сосредоточится на сборе и формулировании требований к приложению. Это включает в себя определение ключевых функций, которые должны быть реализованы, таких как типы игр, механика взаимодействия и уровень сложности. Важно учитывать как функциональные, так и нефункциональные требования, такие как производительность, безопасность и доступность.  
  
• **Анализ целевой аудитории**  
Понимание целевой аудитории является критически важным для успешной разработки приложения. Команда проведет исследования, чтобы определить возрастные группы, интересы и предпочтения пользователей. Это может включать опросы, фокус-группы и анализ существующих приложений на рынке. Результаты этого анализа помогут сформировать стратегию разработки контента и интерфейса приложения, чтобы оно соответствовало ожиданиям пользователей.

• **Анализ конкурентов**  
На этом этапе будет проведен детальный анализ конкурирующих приложений. Команда изучит их функционал, пользовательские отзывы, сильные и слабые стороны. Это позволит выявить уникальные предложения и возможности для улучшения, которые могут быть использованы в нашем приложении. Также важно оценить рыночные тренды и технологии, которые могут повлиять на успех проекта.

• **Разработка концепции приложения**  
Основываясь на собранной информации, команда разработает концепцию приложения. Это будет включать в себя описание основных функций, уникальных особенностей и общего видения продукта. Концепция должна быть согласована с основными целями проекта и ожиданиями целевой аудитории.  
  
• **Проектирование структуры и архитектуры**  
На этом этапе будет разработана структура приложения, включая определение его основных компонентов и их взаимодействия. Команда создаст диаграммы потоков пользователей и архитектурные схемы, которые помогут визуализировать структуру приложения и упростят дальнейшую разработку.  
  
**Фаза 2: Проектирование и дизайн (2 недели)**  
  
• **Разработка прототипов пользовательского интерфейса и пользовательского опыта**  
Команда создаст интерактивные прототипы пользовательского интерфейса (UI) и пользовательского опыта (UX). Эти прототипы позволят протестировать навигацию и взаимодействие с приложением до начала разработки. Пользователи смогут оценить первоначальные идеи, что поможет выявить возможные проблемы на ранних стадиях.  
  
• **Разработка дизайна элементов управления и визуального стиля**  
  
После утверждения прототипов команда разработает дизайн элементов управления, таких как кнопки, меню и другие интерактивные элементы. Визуальный стиль приложения будет согласован с концепцией, включая цветовую палитру, шрифты и графические элементы. Это обеспечит целостность дизайна и улучшит восприятие пользователями.  
  
• **Создание макетов экранов приложения**  
  
Команда создаст детализированные макеты всех экранов приложения. Эти макеты будут служить основой для разработки и помогут команде лучше понять, как будет выглядеть конечный продукт. Макеты также могут быть использованы для презентации заинтересованным сторонам и получения обратной связи.  
  
• **Выбор подходящих графических элементов**  
  
Важным аспектом дизайна является выбор графических элементов, таких как иконки, изображения и анимации. Команда будет работать над созданием или выбором графики, которая соответствует общей концепции приложения и привлекает внимание пользователей.  
  
**Фаза 3: Разработка (3 недели)**  
  
• **Разработка базовой структуры приложения**  
  
На этом этапе команда начнет реализацию программного обеспечения. Будет разработана базовая структура приложения с использованием выбранных технологий и инструментов. Это включает в себя настройку серверной части, базы данных и клиентской части приложения.  
  
• **Реализация основных игр (первый набор)**

Команда сосредоточится на разработке первого набора игр для приложения. Каждая игра будет реализована согласно заранее определенным требованиям, включая механику игры, графику и взаимодействие с пользователем. Важно обеспечить высокое качество кода и следовать лучшим практикам разработки.  
  
• **Доработка дизайна приложения**  
  
Во время разработки команда будет продолжать работу над дизайном приложения, чтобы убедиться, что он соответствует первоначальным макетам. Доработка может включать изменения в интерфейсе на основе обратной связи от тестирования или изменений в функциональности.

**Фаза 4: Тестирование и отладка (4 недели)**  
  
• **Модульное тестирование отдельных функций и игр**  
  
На этом этапе команда проведет модульное тестирование всех функций приложения. Каждая игра будет протестирована на наличие ошибок и проблем с производительностью. Это поможет выявить недочеты на ранних стадиях и избежать более серьезных проблем в будущем.  
  
• **Альфа-тестирование приложения (внутри команды)**  
  
После завершения модульного тестирования команда проведет альфа-тестирование приложения внутри своей группы. Это позволит выявить проблемы с интеграцией различных компонентов и оценить общее качество продукта. Члены команды будут использовать приложение, предоставляя обратную связь о функциональности, удобстве использования и возможных улучшениях.  
  
• **Подготовка к бета-тестированию**  
  
По завершении альфа-тестирования команда подготовит приложение к бета-тестированию с участием внешних пользователей. Это позволит собрать более широкую обратную связь и выявить проблемы, которые могли быть пропущены во время внутреннего тестирования.

**2.3 Модели реализации проекта**

**Основное меню:**

* **Список разделов:** отображается список разделов приложения.
* **Переход в выбранный раздел:** Пользователь выбирает нужный раздел.

**Раздел “Игры”:**

* **Каталог игр:** отображается список доступных игр с иконками.
* **Выбор игры:** Пользователь выбирает интересующую его игру.
* **Запуск игры:** Пользователь нажимает кнопку “Начать игру”.

**Игровой процесс:**

* **Интерактивное взаимодействие:** пользователь взаимодействует с элементами игры (касается экрана, перетаскивает объекты и т. д.).
* **Cчётчик:** во время игры может отображаться счётчик очков.
* **Завершение игры:** игра заканчивается, когда пользователь выполняет условие завершения.

**Результаты игры:**

* **Отображение результатов:** отображается экран с результатами игры.
* **Сохранение результатов:** Результаты игры сохраняются в базу данных пользователя.
* **Переход в главное меню или повтор игры:** пользователю предлагается вернуться в меню или повторить текущую игру.

**Заключение**

В ходе учебной практики был разработан прототип мобильного приложения «Заряди мозг», предназначенного для когнитивной реабилитации. Целью этого проекта было создание инструмента, который предоставлял бы пользователям набор интерактивных игр для тренировки когнитивных функций, таких как память, внимание, мышление и восприятие.

**Основные результаты работы:**

В рамках данного этапа были достигнуты следующие ключевые результаты:

* **Разработана концепция приложения:** были определены основные цели, задачи и функциональные требования, соответствующие предметной области.
* **Спроектирована структура приложения:** создана архитектура приложения и прототипы пользовательского интерфейса, обеспечивающие удобство и интуитивность использования.
* **Реализован базовый функционал:** разработана интерактивная игра «Найди пару», направленная на тренировку различных когнитивных функций.
* **Проведено тестирование:** проведено альфа-тестирование с целью выявления и исправления первоначальных ошибок.

**Выводы:**

В ходе работы над проектом было приобретено ценное понимание процесса разработки мобильных приложений для когнитивной реабилитации. Были приобретены навыки командной работы, а также практический опыт применения различных инструментов и технологий разработки. Работа над проектом показала важность тщательного планирования, проектирования и тестирования для создания эффективного и удобного продукта.

**Планы по доработке:**

В дальнейшем планируется продолжить работу над приложением, уделив особое внимание следующим направлениям:

* **Расширение набора игр:** добавление новых игр, направленных на тренировку более широкого спектра когнитивных функций.
* **Улучшение игровых механик:** доработка существующих игр для обеспечения большей увлекательности и эффективности.
* **Улучшение дизайна:** обновление пользовательского интерфейса и визуального стиля приложения на основе полученной обратной связи.
* **Адаптация уровня сложности:** разработка более гибкой системы адаптации сложности игр под индивидуальные потребности пользователя.
* **Интеграция дополнительных функций:** добавление функционала для персональной настройки приложения и получения рекомендаций на основе анализа данных пользователя.
* **Улучшение системы отслеживания прогресса:** более наглядное отображение статистики и результатов, а также сравнение с другими пользователями.
* **Проведение бета-тестирования:** привлечение реальных пользователей для оценки приложения и получения более точной обратной связи.
* **Публикация приложения:** Публикация приложения в магазинах приложений.

**Приложение**

**Техническое задание**

1. Введение

1.1 Наименование продукта

Мобильное приложение для когнитивной реабилитации "Заряди мозг".

1.2 Краткая характеристика области применения

Продукт предназначен для восстановления и тренировки когнитивных функций (памяти, внимания, логического мышления) у людей, страдающих от последствий инсультов, черепно-мозговых травм, возрастных изменений, а также для предотвращения когнитивных нарушений.

2. Основания для разработки

Приложение создается в ответ на отсутствие доступных, персонализированных решений для когнитивной реабилитации, которые могли бы использоваться в домашних условиях. В условиях роста численности пожилого населения и увеличения частоты инсультов и травм головы необходимость таких решений особенно актуальна.

3. Назначение разработки

Продукт предназначен для использования:

* частными лицами, нуждающимися в реабилитации когнитивных функций;
* врачами-реабилитологами и специалистами по нейропсихологии;
* больницами и реабилитационными центрами для мониторинга и поддержки пациентов.

4 Требования к цифровому продукту

4.1 Требования к функциональным характеристикам

4.1.1 Требования к составу выполняемых функций

Игровой модуль. Предоставление игр для тренировки памяти, внимания и логического мышления. Игры должны адаптироваться к уровню пользователя с учетом прогресса.

Персонализация. Алгоритмы настройки программ реабилитации на основе рекомендаций врача, состояния пациента и его целей.

Система напоминаний. Автоматическое напоминание о тренировках, приеме лекарств и выполнении рекомендаций врача.

Аналитический модуль. Отображение прогресса пользователя в удобной форме (графики, отчеты).

Связь с врачом. Возможность отправки отчетов врачу, получение рекомендаций и консультаций.

Контентная база. Советы по реабилитации, инструкции и полезные материалы для пациентов и их родственников.

Интеграция с медицинскими устройствами. Возможность подключения трекеров здоровья (при необходимости).

4.1.2 Требования к организации входных и выходных данных

Входные данные:

* Информация о пользователе (возраст, диагноз, уровень исходных когнитивных способностей).
* Ввод данных вручную (результаты тестов) или автоматический (синхронизация с трекерами здоровья).

Выходные данные:

* Графики и аналитические отчеты по результатам тренировок.
* Уведомления и рекомендации в текстовом формате.

4.1.3 Требования к временным характеристикам.

Обновление данных должно происходить в реальном времени.

Напоминания о тренировках — с точностью до минуты.

Синхронизация с внешними устройствами — не реже одного раза в минуту.

4.2 Требования к надежности

Приложение должно обеспечивать высокую надежность работы, включая возможность восстановления данных в случае сбоя. Важно, чтобы данные регулярно сохранялись и имели резервные копии, что позволит избежать их потери при отказе системы. Время восстановления после сбоя не должно превышать 30 минут, чтобы минимизировать время простоя и обеспечить бесперебойную работу приложения. Также необходимо предусмотреть защиту от некорректных действий пользователей, таких как отмена операций, которые могут привести к ошибкам в функционировании системы.

4.2.1 Требования к обеспечению надежного (устойчивого) функционирования программы.

* Обеспечение сохранности данных через резервное копирование в облачное хранилище (ежедневно).
* Автосохранение прогресса после каждой завершенной сессии.

4.2.2 Время восстановления после отказа.

В случае отказа системы, время восстановления не должно превышать 30 минут. Это обеспечит минимизацию простоя и бесперебойную работу приложения.

4.2.3 Отказы из-за некорректных действий оператора

Для предотвращения сбоев в работе системы необходимо исключить действия пользователей, которые могут вызвать сбои. Например, пользователь не должен иметь доступ к функциям, которые могут повлиять на целостность данных или привести к сбоям в функционале.

4.3 Условия эксплуатации.

Климатические условия не влияют на работу приложения.

4.3.1 Климатические условия эксплуатации.

Приложение не имеет особых климатических требований, так как оно является программным продуктом, не зависящим от окружающей среды.

4.3.2 Требования к видам обслуживания.

* Обновление приложения через магазины приложений.
* Поддержка пользователей через встроенный чат.

4.3.3 Требования к численности и квалификации персонала.

Для обслуживания требуется 1-2 технических специалиста со знанием мобильной разработки и базовых навыков аналитики.

4.4 Требования к составу и параметрам технических средств.

* Смартфоны или планшеты с ОС Android 8.0 или выше.
* Наличие подключения к интернету для обновлений и синхронизации.

4.5 Требования к информационной и программной совместимости.

Приложение должно быть совместимо с последними версиями мобильных операционных систем Android

4.6 Требование к маркировке и упаковке.

Не требуется

4.7 Требования к транспортированию и хранению.

Не требуется

4.8 Специальные требования.

Не требуется

5 Требования к программной документации.

Для успешного использования приложения потребуется руководство пользователя и методики по тестированию и внедрению. Также необходимо предоставить документацию по администрированию системы для администраторов, а также инструкцию по установке и настройке приложения.

6 Технико-экономические показатели.

В рамках текущей разработки расчёт экономической эффективности не предусмотрен.

7 Стадии и этапы разработки.

1. Разработка технического задания.
2. Создание прототипа интерфейса.
3. Разработка приложения.
4. Тестирование функционала и безопасности.
5. Выпуск продукта и тестирование на целевой группе пользователей.
6. Регулярное обновление и техническая поддержка.

8 Порядок контроля и приемки.

* Проверка функциональности всех модулей.
* Тестирование на отказоустойчивость.
* Аудит безопасности данных.
* Оценка удовлетворенности пользователей (пилотное внедрение).

**Листинг кода ПО**

MainActivity.java

package com.example.chargeyourbrain;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.widget.Button;  
import android.widget.Toast;  
import androidx.appcompat.app.AppCompatActivity;  
  
public class MainActivity extends AppCompatActivity {  
  
 private Button gamesButton;  
 private Button remindersButton;  
 private Button chatButton;  
 private Button settingsButton;  
 private Button loginButton;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
   
 gamesButton = findViewById(R.id.*gamesButton*);  
 remindersButton = findViewById(R.id.*remindersButton*);  
 chatButton = findViewById(R.id.*chatButton*);  
 settingsButton = findViewById(R.id.*settingsButton*);  
 loginButton = findViewById(R.id.*loginButton*);  
  
   
 gamesButton.setOnClickListener(view -> {   
 startActivity(new Intent(MainActivity.this, GamesActivity.class));  
 });  
  
   
 remindersButton.setOnClickListener(view -> {   
 showWinDialog();  
 });  
  
   
 chatButton.setOnClickListener(view -> {   
 showWinDialog();  
 });

settingsButton.setOnClickListener(view -> {   
 showWinDialog();  
 });  
   
 loginButton.setOnClickListener(view -> {   
 showWinDialog();  
 });  
 }  
 private void showWinDialog() {   
 android.app.AlertDialog.Builder builder = new android.app.AlertDialog.Builder(this);  
 builder.setTitle("Кнопка пока не работает. Будет реализованно в будущем!");  
  
 builder.setNegativeButton("Выйти", (dialog, which) -> dialog.dismiss());  
  
 android.app.AlertDialog dialog = builder.create();  
 dialog.show();  
 }  
}

GamesActivity.java

package com.example.chargeyourbrain;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import androidx.appcompat.app.AppCompatActivity;  
  
public class GamesActivity extends AppCompatActivity {  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_games*);  
   
 Button gameButton = findViewById(R.id.*gameButton*);  
 gameButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Intent intent = new Intent(GamesActivity.this, GameDetailsActivity.class);  
 startActivity(intent);  
 }  
 });  
  
   
 Button backButton = findViewById(R.id.*backButton*);  
 backButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 finish();   
 }  
 });  
 }  
}

GameDetailsActivity.java

package com.example.chargeyourbrain;  
  
import android.content.SharedPreferences;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.GridLayout;  
import android.widget.TextView;  
import androidx.appcompat.app.AppCompatActivity;  
import android.animation.ObjectAnimator;  
import android.animation.AnimatorSet;  
  
public class GameDetailsActivity extends AppCompatActivity {  
  
 private static final String *PREFS\_NAME* = "MatchCardGamePrefs";  
 private MatchCardGame game;  
 private Button[] buttons;  
 private TextView flipsText;  
 private int size = 24;   
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_game\_details*);  
  
 flipsText = findViewById(R.id.*flipsText*);  
 GridLayout gameGrid = findViewById(R.id.*gameGrid*);  
  
   
 Button restartButton = findViewById(R.id.*restartButton*);  
 Button backButton = findViewById(R.id.*backButton*);  
  
 restartButton.setOnClickListener(v -> restartGame());  
 backButton.setOnClickListener(v -> finish());  
  
   
 initializeGame(gameGrid);  
 flipsText.setText("Перевороты: " + game.getFlips());  
 }  
  
 private void initializeGame(GridLayout gameGrid) {   
 SharedPreferences prefs = getSharedPreferences(*PREFS\_NAME*, *MODE\_PRIVATE*);  
  
 if (prefs.contains("cards")) {  
 char[] savedCards = prefs.getString("cards", "").toCharArray();  
 boolean[] savedFaceUp = getBooleanArray(prefs, "faceUp", size);  
 int savedFlips = prefs.getInt("flips", 0);  
 game = new MatchCardGame(size, savedCards, savedFaceUp, savedFlips);  
 } else {  
 game = new MatchCardGame(size);   
 }

buttons = new Button[size];  
 for (int i = 0; i < size; i++) {  
 Button button = new Button(this);  
 button.setText("#");  
 button.setTag(i);   
 button.setOnClickListener(this::onCardClick);  
  
   
 GridLayout.LayoutParams params = new GridLayout.LayoutParams();  
 params.width = 0;  
 params.height = 0;  
 params.columnSpec = GridLayout.*spec*(GridLayout.*UNDEFINED*, 1f);  
 params.rowSpec = GridLayout.*spec*(GridLayout.*UNDEFINED*, 1f);  
 button.setLayoutParams(params);  
  
 buttons[i] = button;  
 gameGrid.addView(button);  
  
   
 if (prefs.contains("cards") && game.getFaceUpArray()[i]) {  
 button.setText(String.*valueOf*(game.getCards()[i]));  
 button.setEnabled(false);  
 if (game.isMatched(i)) {  
 button.setVisibility(View.*INVISIBLE*);  
 }  
 }  
 }  
  
 if (game.gameOver()) {  
 closeAllCards();  
 }  
 }  
  
 private void onCardClick(View view) {  
 int index = (int) view.getTag();  
   
 if (!game.flip(index)) {  
 return;  
 }  
  
   
 char flippedCard = game.previousFlipIdentity();  
 flipCardAnimation(buttons[index], true, flippedCard);  
  
   
 if (game.getFlips() % 2 == 0) {  
 if (game.wasMatch()) {  
 int firstIndex = game.getPreviousFlippedIndex();  
 int secondIndex = game.getLastFlippedIndex();  
  
 view.postDelayed(() -> handleMatchedCards(firstIndex, secondIndex), 500);  
 } else {  
 view.postDelayed(this::resetMismatch, 1000);  
 }  
 }  
  
   
 flipsText.setText("Перевороты: " + game.getFlips());  
  
   
 if (game.gameOver()) {  
 int currentScore = game.getFlips();   
 int bestScore = getBestScore();   
  
   
 showWinDialog(currentScore, bestScore);  
  
 closeAllCards();   
 }  
 }  
  
 private void closeAllCards() {   
 game = new MatchCardGame(size);  
  
   
 for (int i = 0; i < buttons.length; i++) {  
 buttons[i].setText("#");  
 buttons[i].setEnabled(true);  
 buttons[i].setVisibility(View.*VISIBLE*);  
 }  
  
   
 flipsText.setText("Перевороты: 0");  
  
   
 SharedPreferences prefs = getSharedPreferences(*PREFS\_NAME*, *MODE\_PRIVATE*);  
 SharedPreferences.Editor editor = prefs.edit();  
 editor.clear();  
 editor.apply();  
 }  
  
 private void handleMatchedCards(int firstIndex, int secondIndex) {  
 ObjectAnimator fadeOutFirst = ObjectAnimator.*ofFloat*(buttons[firstIndex], "alpha", 1f, 0f);  
 ObjectAnimator fadeOutSecond = ObjectAnimator.*ofFloat*(buttons[secondIndex], "alpha", 1f, 0f);  
  
 fadeOutFirst.setDuration(500);  
 fadeOutSecond.setDuration(500);  
  
 fadeOutFirst.addListener(new android.animation.AnimatorListenerAdapter() {  
 @Override  
 public void onAnimationEnd(android.animation.Animator animation) {  
 buttons[firstIndex].setVisibility(View.*INVISIBLE*);  
 }  
 });  
  
 fadeOutSecond.addListener(new android.animation.AnimatorListenerAdapter() {  
 @Override  
 public void onAnimationEnd(android.animation.Animator animation) {  
 buttons[secondIndex].setVisibility(View.*INVISIBLE*);  
 }  
 });  
  
 AnimatorSet animatorSet = new AnimatorSet();  
 animatorSet.playTogether(fadeOutFirst, fadeOutSecond);  
 animatorSet.start();  
 }  
  
 private void resetMismatch() {  
 int firstIndex = game.getPreviousFlippedIndex();  
 int secondIndex = game.getLastFlippedIndex();  
  
 if (firstIndex != -1) {  
 flipCardAnimation(buttons[firstIndex], false, '#');  
 }  
 if (secondIndex != -1) {  
 flipCardAnimation(buttons[secondIndex], false, '#');  
 }  
  
 game.flipMismatch();  
 }  
  
 private void restartGame() {  
 game = new MatchCardGame(size);  
  
 for (Button button : buttons) {  
 button.setText("#");  
 button.setEnabled(true);  
 button.setVisibility(View.*VISIBLE*);  
 }  
  
 flipsText.setText("Перевороты: 0");  
  
 SharedPreferences prefs = getSharedPreferences(*PREFS\_NAME*, *MODE\_PRIVATE*);  
 SharedPreferences.Editor editor = prefs.edit();  
 editor.clear();  
 editor.apply();  
  
 recreate();   
 }  
  
 private int getBestScore() {  
 SharedPreferences prefs = getSharedPreferences("game\_prefs", *MODE\_PRIVATE*);  
 return prefs.getInt("best\_score", 0); }  
  
 private void updateBestScore(int newBestScore) {  
 SharedPreferences prefs = getSharedPreferences("game\_prefs", *MODE\_PRIVATE*);  
 SharedPreferences.Editor editor = prefs.edit();  
 editor.putInt("best\_score", newBestScore);  
 editor.apply();  
 }  
  
 private void showWinDialog(int currentScore, int bestScore) {   
 if (currentScore < bestScore) {  
 bestScore = currentScore;   
 SharedPreferences prefs = getSharedPreferences("game\_prefs", *MODE\_PRIVATE*);  
 SharedPreferences.Editor editor = prefs.edit();  
 editor.putInt("best\_score", bestScore);  
 editor.apply();  
 }  
  
 android.app.AlertDialog.Builder builder = new android.app.AlertDialog.Builder(this);  
 builder.setTitle("Поздравляем с победой!");  
 builder.setMessage("Ваш счёт: " + currentScore + "\nЛучший счёт: " + bestScore);  
 builder.setPositiveButton("Начать заново", (dialog, which) -> restartGame());  
 builder.setNegativeButton("Выйти", (dialog, which) -> finish());  
 builder.create().show();  
 }  
  
 private void flipCardAnimation(Button button, boolean showFace, char faceChar) {  
 ObjectAnimator flipOut = ObjectAnimator.*ofFloat*(button, "scaleX", 1f, 0f);  
 ObjectAnimator flipIn = ObjectAnimator.*ofFloat*(button, "scaleX", 0f, 1f);  
  
 flipOut.setDuration(150);  
 flipIn.setDuration(150);  
  
 flipOut.addListener(new android.animation.AnimatorListenerAdapter() {  
 @Override  
 public void onAnimationEnd(android.animation.Animator animation) {  
 if (showFace) {  
 button.setText(String.*valueOf*(faceChar));  
 button.setEnabled(false);  
 } else {  
 button.setText("#");  
 button.setEnabled(true);  
 }  
 }  
 });  
  
 AnimatorSet flip = new AnimatorSet();  
 flip.playSequentially(flipOut, flipIn);  
 flip.start();  
 }  
  
 @Override  
 protected void onPause() {  
 super.onPause();  
  
 SharedPreferences prefs = getSharedPreferences(*PREFS\_NAME*, *MODE\_PRIVATE*);  
 SharedPreferences.Editor editor = prefs.edit();  
  
 editor.putString("cards", new String(game.getCards()));  
 saveBooleanArray(editor, "faceUp", game.getFaceUpArray());  
 editor.putInt("flips", game.getFlips());   
  
 editor.apply();  
 }  
  
 private boolean[] getBooleanArray(SharedPreferences prefs, String key, int size) {  
 String saved = prefs.getString(key, "");  
 boolean[] array = new boolean[size];  
 for (int i = 0; i < size && i < saved.length(); i++) {  
 array[i] = saved.charAt(i) == '1';  
 }  
 return array;  
 }  
  
 private void saveBooleanArray(SharedPreferences.Editor editor, String key, boolean[] array) {  
 StringBuilder sb = new StringBuilder();  
 for (boolean b : array) {  
 sb.append(b ? '1' : '0');  
 }  
 editor.putString(key, sb.toString());  
 }  
}

MatchCardGame.java

package com.example.chargeyourbrain;  
  
import java.util.ArrayList;  
import java.util.Collections;  
  
public class MatchCardGame {  
 private final char[] cards;   
 private final boolean[] faceUp;   
 private int flips;   
 private int lastFlippedIndex;   
 private boolean lastWasMatch;   
  
 public final int size;   
  
 private int previousFlippedIndex = -1;   
  
   
 public MatchCardGame(int n) {  
 assert n % 4 == 0 && n >= 4 && n <= 4 \* 26;  
 size = n;  
 cards = new char[n];  
 faceUp = new boolean[n];  
 flips = 0;  
 lastFlippedIndex = -1;  
 lastWasMatch = false;  
  
   
 for (int i = 0; i < n; i++) {  
 cards[i] = (char) ('a' + i / 4);  
 }  
  
   
 shuffleCards();  
 }  
  
   
 public MatchCardGame(int n, char[] savedCards, boolean[] savedFaceUp, int savedFlips) {  
 size = n;  
 cards = savedCards;  
 faceUp = savedFaceUp;  
 flips = savedFlips;  
 lastFlippedIndex = -1;  
 previousFlippedIndex = -1;  
 lastWasMatch = false;  
 }  
  
   
 private void shuffleCards() {  
 ArrayList<Character> cardList = new ArrayList<>();  
 for (char c : cards) {  
 cardList.add(c);  
 }  
 Collections.*shuffle*(cardList);  
 for (int i = 0; i < cards.length; i++) {  
 cards[i] = cardList.get(i);  
 }  
 }  
  
   
 public boolean flip(int i) {  
 if (i < 0 || i >= cards.length || faceUp[i]) {  
 return false;   
 }  
  
 faceUp[i] = true;  
 flips++;   
 if (lastFlippedIndex == -1) {  
 lastFlippedIndex = i;  
 } else {  
 previousFlippedIndex = lastFlippedIndex;  
 lastFlippedIndex = i;  
 }  
 return true;  
 }  
  
   
 public boolean wasMatch() {  
 if (previousFlippedIndex == -1 || lastFlippedIndex == -1) {  
 return false;  
 }  
 return cards[lastFlippedIndex] == cards[previousFlippedIndex];  
 }  
  
   
 public char previousFlipIdentity() {  
 if (lastFlippedIndex == -1 || lastFlippedIndex >= cards.length) {  
 throw new IllegalStateException("Нет перевернутой карты.");  
 }  
 return cards[lastFlippedIndex];  
 }

public boolean gameOver() {  
 for (boolean card : faceUp) {  
 if (!card) {  
 return false;   
 }  
 }  
 return true;   
 }  
  
   
 public int getFlips() {  
 return flips;  
 }

public char[] getCards() {  
 return cards.clone();  
 }  
  
   
 public boolean[] getFaceUpArray() {  
 return faceUp.clone();  
 }

public boolean isMatched(int index) {  
 if (!faceUp[index]) {  
 return false; }  
  
   
 for (int i = 0; i < size; i++) {  
 if (i != index && faceUp[i] && cards[i] == cards[index]) {  
 return true;   
 }  
 }  
 return false;   
 }  
  
 public void flipMismatch() {  
 if (previousFlippedIndex != -1 && lastFlippedIndex != -1) {  
 faceUp[previousFlippedIndex] = false;   
 faceUp[lastFlippedIndex] = false;   
 previousFlippedIndex = -1;   
 lastFlippedIndex = -1;   
 }  
 }  
  
 public int getPreviousFlippedIndex() {  
 return previousFlippedIndex;

}  
  
 public int getLastFlippedIndex() {  
 return lastFlippedIndex;

}  
}

activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@drawable/\_im\_69b91f20\_2670\_4e19\_a980\_1386e11d5828"  
 android:gravity="center\_horizontal"  
 android:orientation="vertical"  
 android:padding="16dp">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="16dp"  
 android:gravity="end"  
 android:orientation="horizontal">  
  
 <Button  
 android:id="@+id/settingsButton"  
 style="@style/Widget.Material3.ExtendedFloatingActionButton.Icon.Primary"  
 android:layout\_width="55dp"  
 android:layout\_height="60dp"  
 android:layout\_marginEnd="210dp"  
 android:backgroundTint="#6205A4"  
 android:text="."  
 android:textColor="@color/white"  
 android:textSize="18sp"  
 app:icon="@drawable/icons8\_\_50"  
 app:iconTint="@color/white" />  
  
 <Button  
 android:id="@+id/loginButton"  
 style="@style/Widget.Material3.ExtendedFloatingActionButton.Icon.Primary"  
 android:layout\_width="95dp"  
 android:layout\_height="60dp"  
 android:backgroundTint="#6205A4"  
 android:text="Вход"  
 android:textColor="@color/white"  
 android:textSize="18sp" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:gravity="center"  
 android:orientation="vertical">  
  
 <ImageView  
 android:id="@+id/imageView3"  
 android:layout\_width="218dp"  
 android:layout\_height="164dp"  
 app:srcCompat="@drawable/\_21" />  
  
 <Button  
 android:id="@+id/gamesButton"  
 style="@style/Widget.Material3.ExtendedFloatingActionButton.Icon.Primary"  
 android:layout\_width="200dp"  
 android:layout\_height="70dp"  
 android:layout\_margin="20dp"  
 android:backgroundTint="#6205A4"  
 android:text="Игры"  
 android:textColor="#FFFFFF"  
 android:textSize="20sp" />  
  
 <Button  
 android:id="@+id/chatButton"  
 style="@style/Widget.Material3.ExtendedFloatingActionButton.Icon.Primary"  
 android:layout\_width="202dp"  
 android:layout\_height="70dp"  
 android:layout\_margin="20dp"  
 android:backgroundTint="#6205A4"  
 android:text="Чат"  
 android:textColor="@color/white"  
 android:textSize="20sp" />  
  
 <Button  
 android:id="@+id/remindersButton"  
 style="@style/Widget.Material3.ExtendedFloatingActionButton.Icon.Primary"  
 android:layout\_width="200dp"  
 android:layout\_height="70dp"  
 android:layout\_margin="20dp"  
 android:backgroundTint="#6205A4"  
 android:text="Напоминания"  
 android:textColor="@color/white"  
 android:textSize="20sp"  
 app:strokeColor="#01000000" />  
  
 </LinearLayout>  
  
</LinearLayout>

activity\_games.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@drawable/\_im\_69b91f20\_2670\_4e19\_a980\_1386e11d5828"  
 android:gravity="center"  
 android:orientation="vertical"  
 android:padding="16dp">  
   
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="32dp"  
 android:gravity="center"  
 android:text="Игры"  
 android:textColor="#000000"  
 android:textSize="40sp"  
 android:textStyle="bold" />  
   
 <LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:gravity="center"  
 android:orientation="vertical">  
  
 <Button  
 android:id="@+id/gameButton"  
 style="@style/Widget.AppCompat.Button.Colored"  
 android:layout\_width="200dp"  
 android:layout\_height="200dp"  
 android:background="@android:drawable/divider\_horizontal\_dark"  
 android:foreground="@drawable/icni"  
 app:cornerRadius="0dp"  
 app:iconSize="100dp" />  
  
 <TextView  
 style="@android:style/Widget.Material.Light.TextView.SpinnerItem"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="8dp"  
 android:background="#4E6205A4"  
 android:gravity="center"  
 android:text="Найди пару"  
 android:textColor="#040000"  
 android:textSize="30sp" />  
 </LinearLayout>  
   
 <Button  
 android:id="@+id/backButton"  
 style="@style/Widget.Material3.ExtendedFloatingActionButton.Icon.Primary"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="16dp"  
 android:backgroundTint="#6205A4"  
 android:text="Назад"  
 android:textColor="@color/white" />  
</LinearLayout>

activity\_game\_details.xml

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@drawable/\_im\_69b91f20\_2670\_4e19\_a980\_1386e11d5828"  
 android:orientation="vertical"  
 android:padding="16dp"  
 tools:context=".MainActivity">  
  
 <TextView  
 android:id="@+id/flipsText"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Перевороты: 0"  
 android:textColor="#000000"  
 android:textSize="18sp" />  
  
 <Button  
 android:id="@+id/restartButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:backgroundTint="#6205A4"  
 android:text="Начать заново" />  
  
 <Button  
 android:id="@+id/backButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:backgroundTint="#6205A4"  
 android:text="Вернуться назад" />  
  
 <GridLayout  
 android:id="@+id/gameGrid"  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 android:layout\_weight="1"  
 android:columnCount="4"  
 android:padding="8dp"  
 android:rowCount="6" />  
</LinearLayout>

**Источники**

1. Разработчики Android [Электронный ресурс] // URL: <https://developer.android.com/> (дата обращения: 01.11.2024).

2. Material Design [Электронный ресурс] // URL: <https://material.io/> (дата обращения: 07.11.2024).

3.Android Studio [Электронный ресурс] // URL: <https://developer.android.com/studio> (дата обращения: 16.11.2024)

4. Android Arsenal [Электронный ресурс] // URL: <https://android-arsenal.com/> (дата обращения: 17.11.2024).

5. **IncRussia** [Электронный ресурс] // URL: <https://incrussia-ru.turbopages.org/incrussia.ru/s/news/okazalos-chto-mobilnye-igry-i-golovolomki-uluchshayut-kognitivnye-sposobnosti-i-uluchshayut-sostoyanie-pri-dementsii/> (дата обращения: 20.10.2024)

6. Компоненты архитектуры Android [Электронный ресурс] // URL: <https://developer.android.com/topic/libraries/architecture/index.html> (дата обращения: 18.11.2024).

7. Учебники по Java [Электронный ресурс] // URL: <https://docs.oracle.com/javase/tutorial/> (дата обращения: 30.11.2024).

8. Java Code Geeks [Электронный ресурс] // URL: <https://www.javacodegeeks.com/> (дата обращения: 03.12.2024).

9.  Baeldung [Электронный ресурс] // URL: <https://www.baeldung.com/> (дата обращения: 05.11.2024).

10. Stack Overflow [Электронный ресурс] // URL: <https://stackoverflow.com/questions/tagged/java> (дата обращения: 07.12.2024).