**Федеральное государственное автономное образовательное учреждение   
высшего образования**

**КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ**

**ВЫСШАЯ ШКОЛА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И   
ИНФОРМАЦИОННЫХ СИСТЕМ**

Направление подготовки: 09.03.03 Прикладная информатика

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**Использование сенсорных технологий считывания движений для диагностики заболеваний нервной системы**

**Работа завершена:**

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2017 г.

Студент группы 11-305 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Л.Р. Абукаева

**Работа допущена к защите:**

Научный руководитель

Старший преподаватель

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2017 г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_В.В. Кугуракова

Директор Высшей школы ИТИС

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2017 г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.Ф. Хасьянов

Казань – 2017 г.

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 4](#_Toc485081433)

[1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 6](#_Toc485081434)

[1.1 Медицинские понятия и классификация заболеваний нервной системы 6](#_Toc485081435)

[1.1.1 Медицинские понятия 6](#_Toc485081436)

[1.1.2 Классификация заболеваний нервной системы 7](#_Toc485081437)

[1.2 Диагностика пациентов с заболеваниями нервной системы 8](#_Toc485081438)

[1.3 Выбор направления диагностирования 9](#_Toc485081439)

[2 СЕНСОРНЫЕ ТЕХНОЛОГИИ СЧИТЫВАНИЯ ДВИЖЕНИЙ 11](#_Toc485081440)

[2.1 Актуальность использования сенсорных технологий 11](#_Toc485081441)

[2.2 Обзор технических средств, способных считывать движения 12](#_Toc485081442)

[2.2.1 Браслет Myo 12](#_Toc485081443)

[2.2.2 Перчатка Control VR 12](#_Toc485081444)

[2.2.3 Технология Kinect 13](#_Toc485081445)

[2.2.4 Технология LeapMotion 15](#_Toc485081446)

[2.2.5 Технология iMotion 15](#_Toc485081447)

[2.3 Обоснование выбора технологии Kinect для диагностирования 16](#_Toc485081448)

[3 РЕАЛИЗАЦИЯ ПО ДЛЯ OS ANDROID 18](#_Toc485081449)

[3.1 Техническое задание 18](#_Toc485081450)

[3.2 Диаграмма классов 21](#_Toc485081451)

[4 ОПИСАНИЕ МЕТОДОВ ИССЛЕДОВАНИЯ 22](#_Toc485081452)

[4.1 Требования к оборудованию 22](#_Toc485081453)

[4.2 Необходимые условия для начала проведения исследования 24](#_Toc485081454)

[5 ПРОЦЕСС ДИАГНОСТИКИ 27](#_Toc485081455)

[5.1 Диагностика с помощью технологии Kinect 27](#_Toc485081456)

[5.2 Диагностика когнитивных отклонений 30](#_Toc485081457)

[5.3 Анализ результатов 33](#_Toc485081458)

[ЗАКЛЮЧЕНИЕ 36](#_Toc485081459)

[ГЛОССАРИЙ 38](#_Toc485081460)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 39](#_Toc485081461)

[ПРИЛОЖЕНИЕ А 41](#_Toc485081462)

**ВВЕДЕНИЕ**

Спектр неврологических заболеваний широк, и зачастую врачи сталкиваются с проблемами при диагностировании и лечении болезней нервной системы, в связи с обширной клинической картиной этих заболеваний.

По данным статистики до 30% населения мира старше 35 лет [1], подвержены развитию дегенеративных отклонений нервной системы. Общее количество людей, страдающее дегенеративными изменениями, составляет 80%.

Когнитивное расстройство является одним из самых глобальных симптомов развития у человека неврологических заболеваний. Одним из видов когнитивного расстройства является деменция, то есть заболевание, связанное с нарушениями познавательной деятельности. По данным Всемирной организации здравоохранения в мире насчитано более 46 миллионов людей с деменцией, к 2050 году число увеличится до 131,5 миллионов [2].

Таким образом, одной из главных задач, специалистов в области неврологии, является поиски новых методов обнаружения заболеваний нервной системы. Ранняя диагностика позволит определиться с путями наиболее эффективного лечения и обеспечит скорейшее выздоровление пациента.

Однако, в настоящее время, диагностирование заболеваний нервной системы осуществляется лишь специализированным медицинским оборудованием. Подобное оборудование является дорогостоящим и громоздким, а также нуждается в обученном персонале.

В рамках дипломной работы, совместно с отделением клинической неврологии и клинической лингвистики университетской клиники Казани, будут проведены исследования в области диагностирования заболеваний нервной системы, с помощью использования сенсорных технологий считывания движений. Исследование проводится для выявления возможности дальнейшего внедрения сенсорных технологий в медицинские учреждения, в связи со значительной дешевизной, в отличие от медицинского оборудования.

В качестве сенсорной технологии считывания движений, которая позволит проводить диагностирование заболеваний нервной системы, будет рассмотрена технология Kinect от компании Microsoft.

Проводимое исследование позволит определить то, что технология Kinect, захватывающая движения тела пациента и транслирующая их в виртуальной среде, может эффективно проводить диагностику возможных патологий нервной системы, поскольку использует высокочувствительный инфракрасный сенсор, способный улавливать малейшие движения.

Цели дипломной работы:

1. Проведение исследований в области диагностирования заболеваний нервной системы, с помощью сенсорных технологий считывания движения.
2. Проведение тестирований над испытуемыми, с помощью технологии Kinect.
3. Разработка мобильного приложения (опросника), на платформе OS Android, для диагностирования когнитивных отклонений.
4. Проведение тестирования над испытуемыми для выявления когнитивных отклонений, с помощью мобильного приложения (опросника).
5. Сбор и анализ полученных результатов тестирования.
6. Определение способов улучшения проекта.
7. Определение возможности применения в органах здравоохранения.

# 1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

* 1. **Медицинские понятия и классификация заболеваний нервной системы**

**1.1.1 Медицинские понятия**

Нервная система – это комплекс взаимосвязанных морфологических и функциональных нервных структур, работающая совместно с эндокринной системой и обеспечивающая регуляцию деятельности всех систем организма и реакцию на изменчивость условий внешней и внутренней среды. Нервная система представляет собой интегративную систему, способную объединять работу регуляторной системы, чувствительность и двигательную активность человека [3].

Нервные болезни – это отклонения от нормального состояния функциональности нервной системы. Возникновением этих заболеваний, а также их лечением занимается такая дисциплина, как неврология [4].

Неврология – это группа медико-биологических дисциплин, изучающая патологии и нормальное состояние нервной системы. Целью данной дисциплины является понимание вопроса о причинах возникновения заболеваний центральной и периферийной нервной системы. Неврология занимается изучением механизмов развития заболеваний, их симптомы, способы диагностирования, лечения и возможной профилактики [5].

При диагностировании и лечении болезней нервной системы врачи зачастую сталкиваются с проблемами, в связи с обширной клинической картиной этих заболеваний.

**1.1.2 Классификация заболеваний нервной системы**

Существует пять основных групп заболеваний нервной системы в зависимости от причин и условий их возникновения [6]:

1. Сосудистые заболевания нервной системы.

Отклонения от нормы возникают из-за нарушения работы в структурах мозга по причинам сосудисто-мозговой недостаточности и нарушения кровообращения.

2. Инфекционные заболевания нервной системы.

Возникают по причинам инфекционного (вирусного, грибкового, паразитарного, бактериального) поражения головного или спинного мозга, а также в редких случаях периферического отдела.

3. Наследственные заболевания нервной системы.

Различают геномные и хромосомные виды нервных болезней, при геномных отклонениях возникает поражение нервно-мышечной системы, что приводит к нарушениям двигательного аппарата и эндокринной системы.

4. Хронические заболевания нервной системы.

Отклонения вызваны специфическим строением нервной системы и инфекционного воздействия на нее, приводящего к нарушениям в обменных процессах организма.

5. Травматические заболевания нервной системы.

Отклонения возникают в результате травм, ушибов и механического воздействия на органы нервной системы.

**1.2 Диагностика пациентов с заболеваниями нервной системы**

Диагностика заболеваний нервной системы, как и других болезней, строится на определении неврологического статуса пациента, сбора анамнеза и тщательного осмотра.

Неврологический осмотр начинают с проверки сознания, снижения интеллекта, проверки ориентации во времени и пространстве, проверки нарушений речи у пациента. А также проверки двигательных функций, как пациент держит вытянутые вперед руки и движения в них, признаки нарушений чувствительности, возможное наличие патологических рефлексов, активные движения в суставах.

Клинические данные способны поставить диагноз, однако чаще всего для получения четкой картины заболевания необходимо проводить дополнительные исследования [7].

Инструментальная диагностика заболеваний нервной системы используется при подозрениях на нервные болезни. Главными инструментами для выявления отклонений являются:

* + Эхо-ЭГ (эхоэнцефалография головного мозга) – ультразвуковая диагностика, выявляющая отклонения в структуре головного мозга;
  + рентгенография позвоночника;
  + ЭЭГ (электроэнцефалография) – исследование головного мозга на основе биоэлектрической активности;
  + электромиография;
  + РЭГ (реоэнцефалография) – исследования сосудистой системы головного мозга;
  + нейросонография.

Наиболее точными методами для диагностирования нервных болезней являются магнитно-резонансная и компьютерная томографии, позитронная эмиссионная томография головного мозга, дуплексное сканирование, УЗДГ (ультразвуковая допплерография) сосудов головы.

**1.3 Выбор направления диагностирования**

В связи со значительной классификации болезней нервной системы, в качестве направления для диагностирования, была выбрана определенная группа неврологических расстройств: дегенеративные и когнитивные расстройства нервной системы.

Дегенерация нервной системы – это неподдающиеся восстановлению функциональные и органические изменения систем головного или спинного мозга [8]. Развивающиеся по нарастанию: память ослабляется, и в дальнейшем происходят необратимые органические нарушения структур мозга и психической деградации.

Когнитивные функции организма отвечают за понимание, познание, изучение, осознание, восприятие и переработку информации из внешнего мира. В отсутствие этих функций человек будет не в состоянии познавать окружающий его мир:

* Внимание. У человека отсутствует способность выделять важную информацию из общего потока, а также способность концентрироваться сводится к нулю.
* Восприятие. Блокируется возможность восприятия информации из внешней среды.
* Память. Способность человека сохранять и воспроизводить полученную информацию нарушается.
* Психомоторная функция. Дегенеративные отклонения человека, разрушающие способность выполнять какие-либо двигательные навыки.
* Интеллект. Способность человека анализировать информацию и делать умозаключения нарушается.
* Речь.

Выбор данного направления диагностирования был сделан не случайно, поскольку когнитивные расстройства являются одним из самых глобальных неврологических симптомов, который сигнализирует о том, что стандартная работа головного мозга нарушена. Рациональное познание мира нарушается из-за данного расстройства.

Поэтому возможность проведения исследований в области диагностирования заболеваний нервной системы, с помощью новейших информационных технологий, так актуально, поскольку в настоящий момент в данной области применяется только инструментальная диагностика. Данная диагностика требует огромных денежных средств, медицинское оборудование, специализированный персонал и подготовленное помещение.

Нет возможности диагностирования заболеваний нервной системы в домашних условия, в пределах офиса или небольшого помещения без специализированного медицинского оборудования.

Дальнейшие исследования дипломной работы будут основываться на диагностировании когнитивных и дегенеративных отклонений нервной системы, с помощью сенсорных технологий считывания движений.

1. **СЕНСОРНЫЕ ТЕХНОЛОГИИ СЧИТЫВАНИЯ ДВИЖЕНИЙ**

**2.1 Актуальность использования сенсорных технологий**

Использование современных устройств ввода, таких клавиатура или мышь, уже давно являются атрибутами повседневной жизни каждого человека. И данные устройства отлично справляются со своей работой. Также можно сказать и о сенсорном управлении, которое используется в различных мобильных компьютерах и гаджетах.

Не стоит забывать, что тактильный, осязательный методы далеко не являются единственными способами взаимодействия человека с компьютером. Поскольку для общения друг с другом люди могут использовать взгляд, голос или жесты. В современном мире все это может быть применимо и в отношении взаимодействия человека с компьютерами и бытовыми устройствами [9].

Дизайнеры и инженеры уже давно работают над тем, чтобы предоставить пользователям что-то новое, интересное в плане управления компьютером. Эксперты предполагают, что в недалеком будущем человек сможет общаться с компьютерными системами только посредством своих традиционных «средств коммуникации». На данный момент реализуются лишь первые шаги в продвижения к этой цели.

Системы отслеживания движений появились в повседневной жизни людей довольно давно и широко применяются, например, в видеоиграх. Взаимодействие человека с видеоигрой происходит благодаря отслеживанию положения человека в пространстве, с помощью специальных сенсоров или камер, реагирующих на свет. Для этого человеку необходимо держать пульты в поле зрения сенсоров и устройство считает движения [10].

Технологии, способные отслеживать движения, на данный момент имеют широкое распространение лишь в сфере видеоигровой индустрии. Однако со временем подобная технология расширит свои возможности не только в компьютерных играх, но и во многих других научно-технических приложениях, робототехнике, медицине, виртуальной реальности.

Объекты в трехмерном пространстве, объемные интерфейсы, высокочувствительные датчики считывания, в будущем все это даст возможность достижения эффективных результатов в здравоохранительных органах: в диагностировании заболеваний на ранних этапах и в хирургии, в качестве виртуального тренажера для хирургических оперативных вмешательств.

Дальнейшее развитие сенсорных технологий позволит применять их для реабилитации больных с неполноценной, нарушенной моторикой и неспособностью к движению [11].

### 2.2 Обзор технических средств, способных считывать движения

### 2.2.1 [Браслет Myo](https://www.thalmic.com/myo/)

Технологический браслет Муо надевается на руку немного ниже локтя и начинает распознавать сокращение мышц пользователя. При сжимании ладони в кулак, выпрямлении всех пальцев, или же, если пользователь направит ладонь к себе или от себя, данное устройство определяет любое движение и преобразует его в тот или иной сигнал.

### 2.2.2 [Перчатка Control VR](http://controlvr.com/)

На данный момент при использовании компьютера набор функций для жестов достаточно мал, однако при использовании совместно со шлемом виртуальной реальности, устройства для считывания движений могут иметь обширный диапазон применения [12].

Наиболее качественное считывание движений всего или большей части тела дает возможность для полного погружения пользователя в виртуальную реальность. В связи с этим перчатки Control VR для шлема виртуальной реальности Oculus представляют собой скорее целый костюм: гаджет надетый на шею, обвивается вокруг предплечий и заканчивается самими перчатками.

Диапазон использования данного устройства широк, его можно применять в процессе игр, изучении виртуальной реальности, где каждому пользователю предоставляется возможность тактильного взаимодействия с виртуальными предметами. Произведение действий в виртуальной среде выглядит вполне реалистичным, с помощью перчаток Control VR, особенно благодаря точному воспроизведению движений пальцев.

### 2.2.3 Технология Kinect

Самое широкое применение на рынке в области сенсорного считывания движений принадлежит технологии Kinect от компании Microsoft. Технология Kinect представляет собой дополнение для приставки Xbox 360, которая считывает все движения пользователя и позволяет в дополненной виртуальной реальности действовать без контроллеров [13].

Устройства способные отслеживать положения тела человека, начали разрабатывать уже достаточно давно, однако одной их главных проблем являлось то, что для начала отслеживания движений и жестов пользователю необходимо занять эталонную позу. В конце 2010 года компания Microsoft представила публике интерактивный контроллер Kinect, предоставляющий возможность взаимодействия пользователя с консолью без использования традиционного контроллера в виде джойстика. Расположение тела пользователя и его устная речь является главным управляющим элементом в данном устройстве.

В устройстве Kinect размещены два сенсора. Первый из них является инфракрасным, его функция заключается в «прощупывании» окружающего пространства. Второй сенсор обеспечивает преобразование полученных данных в 3D-проекцию и дальнейшего отслеживания движений человека внутри нее. По всему полю зрения сенсора встроенным процессором происходит измерение расстояния от камеры до сцены. Система определяет пользователя как набор движущихся точек, считывание данных системой происходит каждую десятую долю секунды. В процессе игры устройство Kinect определяет расположение рук, ног и головы пользователя, происходит автоматический подбор виртуального скелета. На основании полученной информации о движениях тела пользователя и подобранного скелета формируется определенный 3D персонаж, участвующий в игре.

Использование сенсорных технологий отслеживания движений и жестов не обходится и без ошибок. Система не всегда может корректно отследить движения человека, так как этого бы хотелось разработчикам технологии бесконтактного управления контентом. Несмотря на это, технология Kinect развивается и начала находить применение за пределами видеоигровой индустрии.

В частности, с помощью Kinect на сегодняшний момент можно проводить эффектные мультимедийные презентации без использования стандартных устройств ввода или контроллера. Технология дает возможность проводить взаимодействия с любыми проекционными системами, плазменными и ЖК-панелями. Пользователю стоит лишь использовать взмахи руки для управления любым контентом, тактильное взаимодействие с дисплеем отсутствует. Развитие технологии Kinect в дальнейшем позволить компании Microsoft использовать данную технологию в мобильных компьютерах и гаджетах.

### 2.2.4 Технология LeapMotion

Технология LeapMotion является альтернативой технологии Kinect в области управления жестами. Разработчики данной технологии утверждают, что она способна более качественно отслеживать движения пользователя.

Суть работы технологии LeapMotion заключается в том, что устройство образует перед собой виртуальный куб со стороной примерно 61 сантиметр, для начала работы пользователю необходимо лишь расположить руки в нем. В пределах куба происходит высокоточное отслеживание жестов и движений рук человека. Идет распознавание всех десяти пальцев, точность их перемещений считывается до долей миллиметра, задержки отсутствуют.

Технология LeapMotion представляет собой компактную коробочку размером с USB-накопитель. Она взаимодействует с компьютером посредством кабеля USB.

Технология LeapMotion только начала свое развитие, однако уже открывает для себя широкую область применения. В дальнейшем, представится возможность использовать эту технологию в мобильных компьютерах или гаджетах.

На данный момент LeapMotion зарекомендовала себя как передовую технологию по отслеживанию движений и управления компьютером без прикосновений.

### 2.2.5 Технология iMotion

Технология iMotion является еще одной инновационной разработкой в области считывания движения. Данная технология дает возможность управлять компьютерами и телевизорами, в которых присутствует интегрированная камера, без прикосновений, с помощью нескольких жестов.

iMotion представляет собой многофункциональное устройство, способное поддерживать различные приложения, игры на ПК, мобильные приложения на разных платформах, таких как iOS и Android. Появляется возможность превращения обычного телевизора в «умный» TV, управление в котором будет осуществляться с помощью жестов.

Перед пользователем проецируется виртуальный сенсорный дисплей, который осуществляет тактильную обратную связь, с помощью которой можно получить реальное физическое ощущение прикосновения к виртуальному объекту.

Устройство представляет собой перчатку, надетую на кисть. На перчатке располагаются в виде треугольника три светодиода. Веб-камера отслеживает движения светодиодов. iMotion распознает команды перемещения, вращения руки. Дистанция, на которой происходит распознавание, составляет до 5 метров от камеры.

Первоначально технология iMotion создавалась как тренажер, способный обучать людей, например, стрельбе. Однако позже концепция проекта была изменена, и теперь это многофункциональное устройство применяется для дополнения к игровой консоли, для управления компьютером и как замена стандартному пульту.

## 2.3 Обоснование выбора технологии Kinect для диагностирования

Технология Kinect является передовой в области сенсорного считывания движений и жестов, на фоне конкурирующих продуктов имеет целый ряд преимуществ [14]:

* дополнительные устройства не требуются;
* руки свободны;
* распознает не только руки, но и все тело в целом;
* имеет функцию распознавания голоса;
* не нуждается в зарядке или замене батареек;
* дает возможность определять жесты и движения любой сложности;
* значительное снижение уровня шума и увеличенная точность глубины датчика предоставляет наиболее качественную трехмерную визуализацию, четкое отображение всех объектов, даже мелких, и более устойчивое отслеживание положения тела;
* предоставляется единый режим для сидячих и стоящих пользователей/игроков;
* повышенное качество камеры глубины позволяет реализовывать различные сценарии с отслеживанием положения тела. Доступна возможность отслеживать 6 человеческих скелетов и 25 суставов на каждого пользователя. Анатомическое отслеживание положения тела является наиболее правильным и устойчивым. Расширился диапазон считывания движений. Система имеет способность распознавать сжимание и разжимание кисти в кулак;
* представлена возможность работы Kinect в практически полной темноте.

Технология Kinect идеально подходит для осуществления исследования в области диагностирования заболеваний нервной системы, с помощью сенсорных технологий считывания движений, поскольку данный контроллер анатомически верно может считать положение тела пациента и выстроить скелет. Благодаря этим возможностям можно легко и четко выявлять развитие отклонений в нервной системе.

**3 РЕАЛИЗАЦИЯ ПО ДЛЯ OS ANDROID**

**3.1 Техническое задание**

1. Наименование системы (см. Приложение А).

Мобильное приложение «Diagnostics» для операционной системы Android.

1. Назначение и цели создания системы.
   1. Назначение. Приложение «Diagnostics» предназначено для проведения диагностики когнитивных отклонений.
   2. Цель. Возможность проведения тестов для выявления нарушений умственной деятельности; получение полного отчета тестирования.
2. Требования к системе.
   1. Требования к программному обеспечению.

* Персональный компьютер – ОС Microsoft Windows 7/8 (х86).
* Интегрированная среда разработки (IDE) – Android Studio.
* Язык программирования – Java.
  1. Требования к мобильному приложению.
* Платформа работы приложения – Android.
* Версия поддерживаемых Android устройств – Android 4.0 или выше.
* Ориентация экрана – портретная.
* Разрешения экранов Android – mdpi (320x480 px), hdpi (480x800 px).
  1. Требования к функциональным характеристикам.
* Возможность выбора направления тестирования.
* Прохождение одного из вариантов тестирования.
* Возможность преждевременного завершения тестирования.
* Получение отчета об ошибках.

1. Интерфейс приложения.

* Локализация приложения предусматривает русскую версию пользовательского интерфейса.
* Главная страница приложения – страница выбора направления тестирования (см. Рисунок 1).

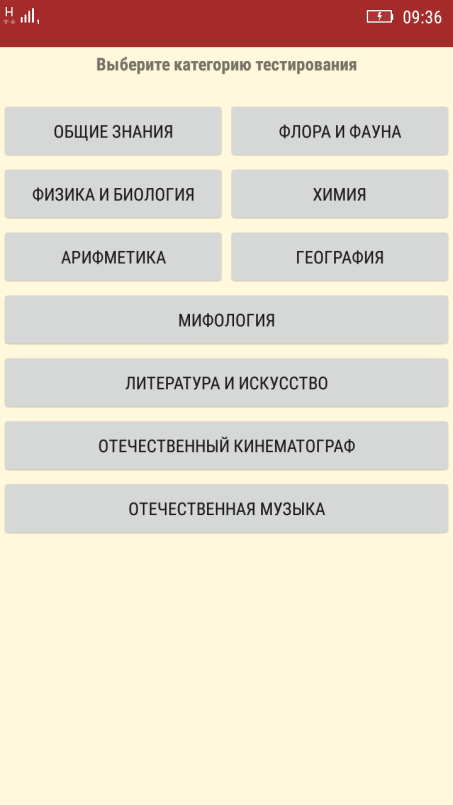


Рисунок 1. Главная страница с выбором направления тестирования

* Вторая страница приложения – страница выбранного тестирования, на которой указывается вопрос и варианты ответа на него. Существует возможность завершения тестирования после прохождения всех вопросов или преждевременное завершение при нажатии на кнопку «Завершить тестирование» (см. Рисунок 2).

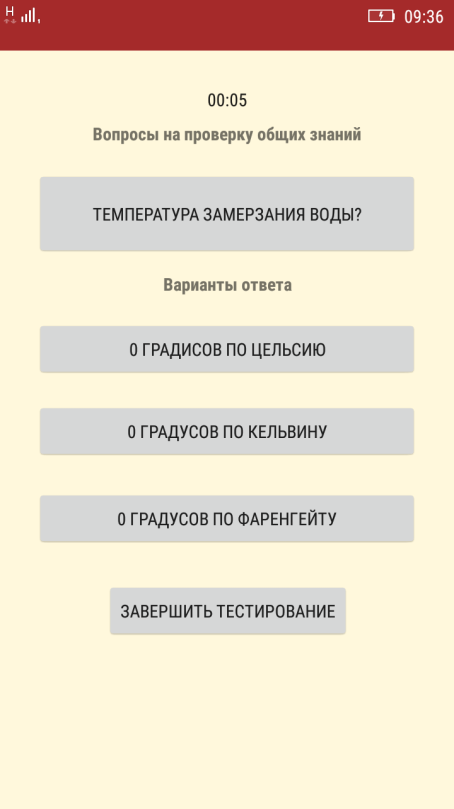


Рисунок 2. Страница с выбранным тестом

* Третья страница приложения – вывод полного отчета тестирования, то есть количество правильных ответов, общее время прохождения тестирования, вывод ошибок (см. Рисунок 3).

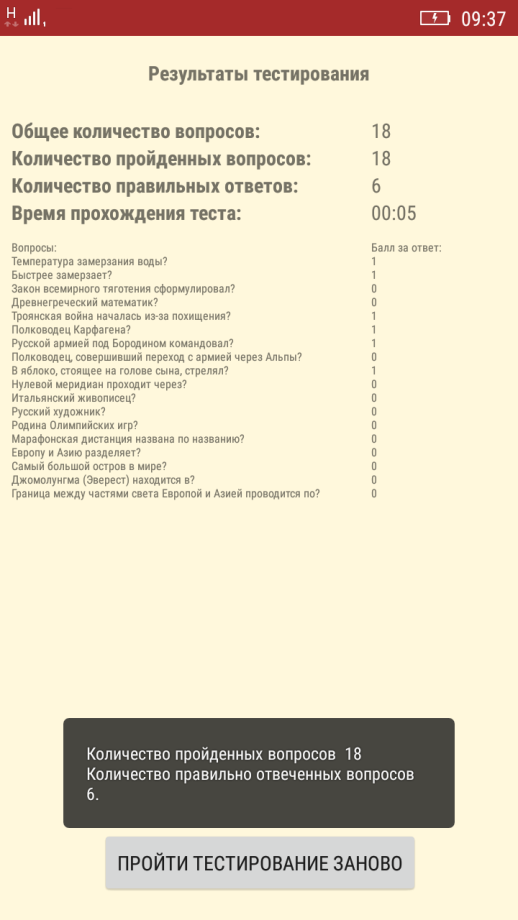


Рисунок 3. Страница с результатами тестирования

**3.2 Диаграмма классов**

Диаграмма классов демонстрирует присутствующие в системе классы, их атрибуты и взаимосвязи (см. Рисунок 4).

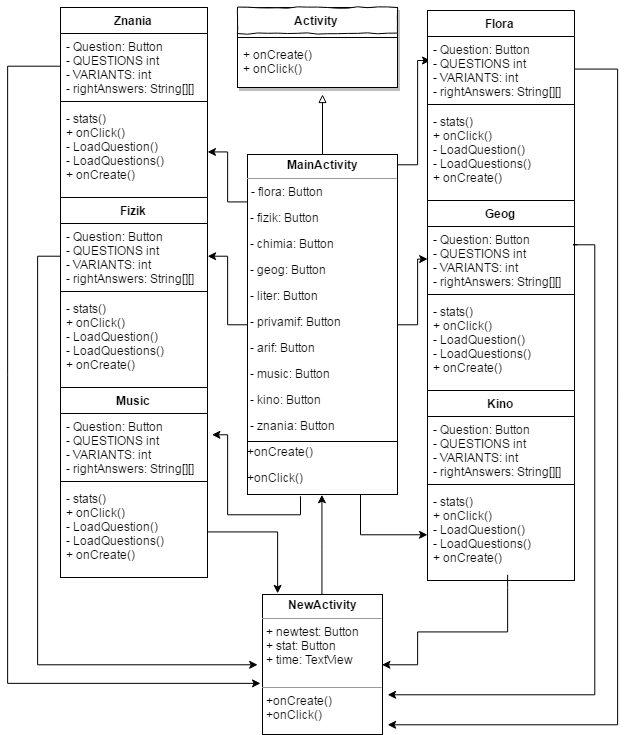
****

Рисунок 4. UML-диаграмма классов



**4 ОПИСАНИЕ МЕТОДОВ ИССЛЕДОВАНИЯ**

### 4.1 Требования к оборудованию

Для проведения исследования необходимы следующие компоненты:

* контроллер Kinect;
* специальный набор драйверов (Microsoft’s Kinect SDK);
* мобильное приложение на OS Android;
* смартфон на платформе OS Android;
* среда разработки Visual Studio 2010;
* персональный компьютер с операционной системой Microsoft Windows 8.

Для обеспечения высочайшей производительности в условиях сложных сценариев отслеживания пользователя были использованы фиксированные конфигурации компьютера, поддерживающие дополнительные вычислительные операции при оптимальной частоте кадров.

Требования к персональному компьютеру:

* 64-битный (x64) процессор;
* двухъядерный процессор с частотой 2,66 ГГц или выше;
* порт USB 2.0;
* 2 ГБ памяти (ОЗУ);
* Сенсор Kinect для Windows.

Для проверки соответствия совместимости к персональному компьютеру, использовались средства конфигурации Kinect, проверяющие систему на предмет известных неполадок и наличия последних версий драйвера для GPU.

Microsoft Kinect SDK – начальный набор инструментов для начала работы с Kinect. Подходит для изучения возможностей NUI (Natural User Interface) с помощью Kinect и связанных технологий [15].

Natural User Interface (NUI) – естественный пользовательский интерфейс, позволяющий пользователю визуально работать с системой, при помощи жестов, а также при помощи голосовых команд.

SDK включает:

* драйверы для использования сенсоров Kinect на ОС Windows;
* API и интерфейсы для работы с устройством;
* примеры с исходными кодами;
* справка по установке и настройки.

SDK предоставляет следующий перечень возможностей:

* Данные от сенсоров (сырой поток данных). Предоставляется возможность самостоятельного анализа данных от сенсора глубины, цветной камеры и данных от микрофонов, генерируемые сенсорами Kinect.
* Отслеживание скелета. Возможность отслеживания движений скелета на основе 20 опорных точек у одного или двух людей, расположенных перед Kinect.
* Аудиовозможности. Возможность определения источников звука и интеграции Windows API для распознавания речи.

OpenNI – программная платформа с открытым исходным кодом, предназначенная для взаимодействия с оборудованием и более высокоуровневыми программами. NITE – промежуточное программное обеспечение, решающее задачи по определению жестов, отслеживания скелета для управления компьютером и играми, работает совместно с OpenNI.

В исследовании будет использоваться Microsoft Kinect SDK.

### 4.2 Необходимые условия для начала проведения исследования

Исследование строилось на поиске способа диагностирования заболеваний нервной системы, с помощью технологии Kinect, способного отслеживать движения и жесты. Проектор Kinect фиксирует мельчайшие движения и мимику пользователя посредством накладывания на пространство перед устройством невидимой глазу сетки из точек. Считывание датчиком происходит 30 раз в секунду, и данные передаются на консоль.

Для получения наиболее точных результатов при проведении исследования в области диагностирования заболеваний нервной системы, необходим ряд условий:

* 1. Фиксированное расстояние. Для получения наибольшего количества пикселей на изображение, расстояние до датчика Kinect должно составлять 2-3 метра, свободное пространство по бокам в каждую сторону должно составлять один метр.
  2. Расположение сенсора. Фиксирование сенсора должно быть на высоте 50-180 метров от пола, угол наклона подстраивается автоматически, датчики способны изменять угол в рамках 27°.
  3. Освещенность и лучи света. Помещение должно быть достаточно освещенным для правильного распознавания пациента и его движений, попадание прямых солнечных лучей должно отсутствовать.
  4. Одежда испытуемых. При выборе одежды должны отсутствовать темные ткани, поскольку они способны поглощать инфракрасное излучение сенсора и мешать корректному отслеживанию скелета.

Данные от каждого найденного скелета преобразуются в координаты 20 точек (см. Рисунок 5).

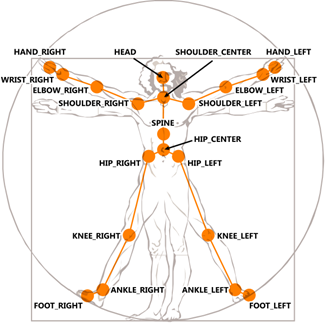


Рисунок 5. Представление скелета пользователя на основе 20 точек – SkeletonFrame

Плагин поддержки Kinect способен продолжать работать, при условии, что отслеживается ровно один скелет, иначе происходит неоднозначное распознавание пользователя и скелета (см. Рисунок 6).

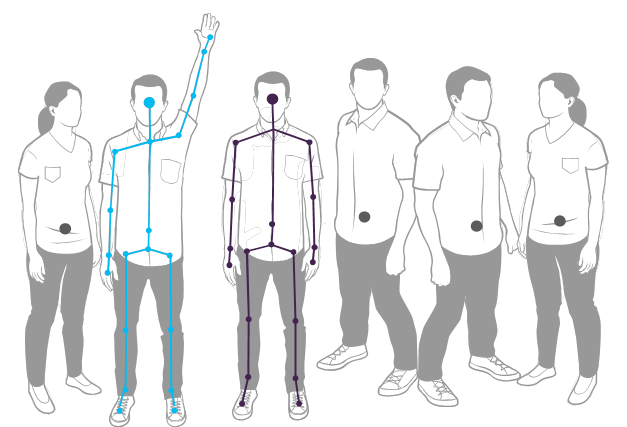


Рисунок 6. Захват скелета одного пользователя

Стандартные методы Kinect SDK сглаживают и обрабатывают координаты. Однако Kinect SDK не предоставляет возможность отслеживать точки скелета пользователя, вместо этого есть возможность управлять коэффициентами усреднения координат, получать данные о корректности точек скелета.

При исследовании было принято ограничить расстояние до датчиков отслеживания Kinect до 180 см, а расстояние до пола – 50 см, на таком расстоянии считывание происходит без искажения (см. Рисунок 7).

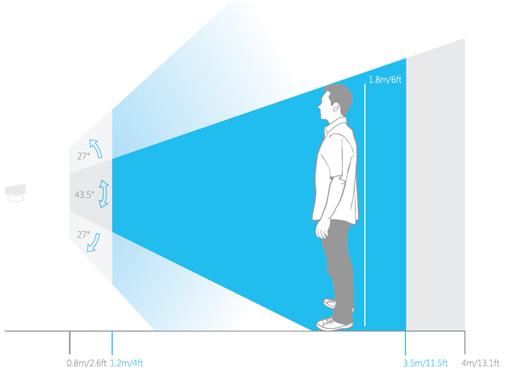


Рисунок 7. Необходимое положение для эффективного распознавания

Используемый в исследовании Kinect SDK, с помощью встроенной функции определяет пользователя в пространстве, благодаря этому есть возможность сосредоточиться на преобразовании полученных данных с сенсора.

В результате, по завершении исследования имеются структурные данные, включающие разнообразную информацию о «скелете», положение точек, формирующие скелет, а также векторные углы.

Сглаживание скелета строится на следующих параметрах:

* Smoothing (сглаживание) - default 0,5;
* Correction (коррекция) - default 0,5;
* Prediction (прогнозирование) - default 0,5;
* JitterRadius (радиус дрожания) - default 0,05;
* MaxDeviationRadius (радиус максимального отклонения) - default 0,05.

### 5 ПРОЦЕСС ДИАГНОСТИКИ

### 5.1 Диагностика с помощью технологии Kinect

В проведении исследования принимали участие десять человек в возрасте от 20 до 25 лет, не страдающие какими-либо заболеваниями. С помощью данной контрольной группы необходимо было выяснить следующие аспекты:

* определить шаблоны двигательных отклонений;
* зафиксировать, где и как отображаются нарушения;
* проанализировать корректность полученных данных;
* сопоставить движения испытуемого с движениями скелета;
* сопоставить зафиксированные отклонения с самочувствием испытуемого в тот момент;
* протестировать испытуемых на двух программах, подстроенных под технологию Kinect;
* сопоставить двигательную активность каждого испытуемого на двух программах;
* проанализировать время выполнения каждого действия;
* определиться с дальнейшими корректировками для возможности проведения наиболее эффективного диагностирования.

Проведение первого тестирования осуществлялось на готовом продукте от Microsoft – Kinect SDK.

После установки данной программы, из списка доступных проектов на C# был выбран Kinect Explorer (см. Рисунок 8).

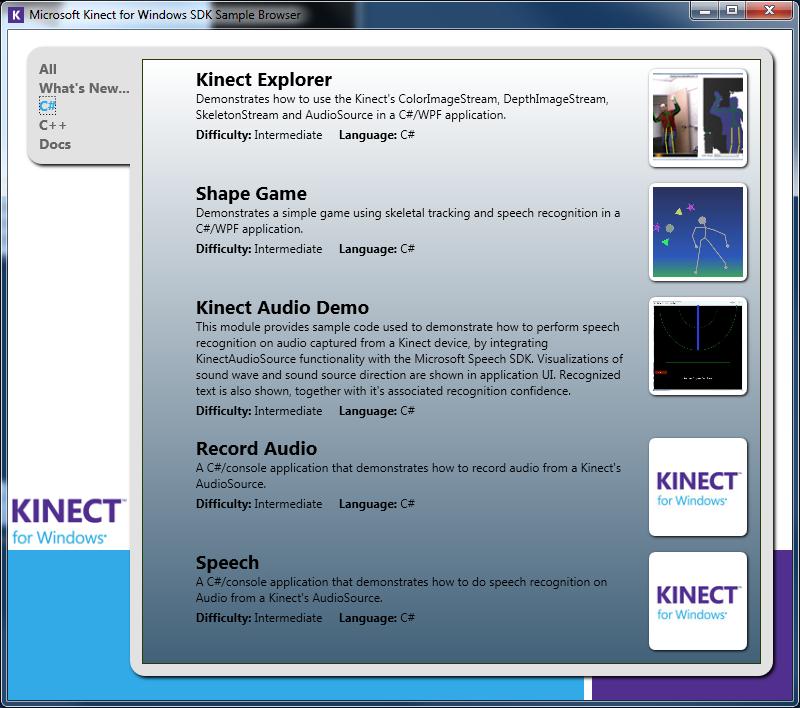


Рисунок 8. Проект Kinect Explorer

Данный проект позволяет отслеживать движения пользователя и выстраивать для него скелет.

При начале проведения тестирования на компьютере запускалась программа записи видео, которая позволяла после тестирования проверять и сверять результаты. Время, отведенное для каждого испытуемого, составляло две минуты.

За время теста испытуемому необходимо было совершить ряд фиксированных действий:

* поочередное поднятие рук;
* движение руками по сторонам;
* статическое удержание рук по сторонам;
* прорисовывание круга в воздухе;
* удержание равновесия на одной ноги с выполнением движений в руках.

При выполнении каждого действия отслеживалось и фиксировалось время. Казалось бы, все действия являются простыми и повседневными, однако при развитии нарушений нервной системы, например, при треморе, выполнение даже легких движений нарушается.

Проведение второго тестирования осуществлялось на базе готового решения Kinect DTW (Kinect SDK Dynamic Time Warping) с открытым исходным кодом, доступного в свободном доступе в сети Интернет (см. Рисунок 9).



Рисунок 9. Проект на основе Skeleton Tracking

При проведении данного тестирования, испытуемым было необходимо выполнить все те же действия, что и при первом тестировании, это было осуществлено в рамках сравнения полученных результатов, и поиска оптимальных решений.

После выполнения всех тестирований осуществлялась сравнительная оценка. Вначале на видеозаписях были сопоставлены движения человека с движениями скелета, в обоих случаях считывание движений было реализовано корректно.

После сопоставлялись скелеты каждого испытуемого с двух тестирований. В связи с тем, что тестирования выполнялись не одновременно, необходимо было учитывать погрешности, но при сравнительном анализе особых отличий выявлено не было.

Поскольку в исследовании проходили участие люди без каких-либо неврологических заболеваний, с уверенностью говорить о том, что были найдены отклонения, связанные с неврологическими заболеваниями, нельзя.

Но при выполнении определенных действий, таких как, статическое удержание рук, стояние на одной ноге, и прорисовывание круга в воздухе, были выявлены небольшие нарушения. Испытуемые зачастую теряли равновесие или прорисовывали изломанный круг. По утверждениям каждого испытуемого, при выполнении действий они не испытывали затруднений, кроме того, что не могли удержать равновесие.

По полученным сравнительным результатам, можно утверждать, что особых различий в применении двух программ выявлено не было. Отличие заключалось лишь в том, что официальный проект Kinect Explorer, быстрее отображал действия пользователя на экране.

Проведенное исследование зафиксировало места, на которых испытуемые совершали ошибки, колебались и тратили больше времени. Данные места были отмечены как шаблоны отклонения и, при проведении дальнейших исследований, будет являться базой, с которой можно проводить сравнения.

## 5.2 Диагностика когнитивных отклонений

При проведении исследований в области использования сенсорных технологий считывания движений для диагностики заболеваний нервной системы, необходимо было составить полную картину возможных отклонений испытуемого.

В связи с этим необходимо было определить не только двигательные отклонения, связанные с заболеваниями нервной системы, полученные с помощью использования технологии Kinect, но также провести у испытуемых диагностику когнитивных нарушений.

Проведение данной диагностики, совместно с полученными результатами уже проведенного ранее исследования, определит наиболее четкий неврологический статус испытуемого. Получение результатов поможет определиться в постановке диагноза и дальнейших методах лечения заболевания.

Диагностика когнитивных отклонений проводилась с помощью мобильного приложения «Diagnostics» на платформе OS Android, которое представляет собой группу связанных тематических тестов, ориентированных на выявление умственных отклонений у пациентов.

Совместно с отделением клинической неврологии и клинической лингвистики были составлены вопросы, сгруппированные по темам, которые ориентированы на определенную область знаний, и определяющие умственные способности испытуемых.

Таким образом, было составлено десять тестов, включающие в себя от 10 до 40 вопросов на одну тему, вопросы были составлены с учетом того, чтобы на них мог ответить человек любой возрастной категории.

Список тематических тестов для проведения оценки когнитивных отклонений:

1. Вопросы на проверку общих знаний.
2. Вопросы на знание флоры и фауны.
3. Вопросы на знание физики и биологии.
4. Вопросы на знание химии.
5. Вопросы на знание арифметики.
6. Вопросы на знание географии.
7. Вопросы на знание мифологии.
8. Вопросы на знание литературы и искусства.
9. Вопросы на знание Отечественной музыки.
10. Вопросы на знание Отечественного кинематографа.

Проведение данного когнитивного теста позволит определить, на чем происходит наибольшее затруднение, выявить у испытуемых возможное начало развития нарушений в познавательном процессе.

В исследовании диагностики когнитивных отклонений принимали участие те же испытуемые, что и в диагностики заболеваний нервной системы, с помощью технологии Kinect. Однако были добавлены дополнительные испытуемые для получения наилучшей статистики. Суммарное число испытуемых составило 30 человек.

Для проверки когнитивных нарушений каждый испытуемый был вынужден пройти все тематически связанные тесты через мобильное приложение «Diagnostics». В нем производился отсчет времени, который испытуемый тратил на ответ, подсчитывалось количество пройденных вопросов, правильных ответов, процент погрешностей. В результате каждому испытуемому был предоставлен полный отчет проходимого тестирования.

По отчетам всех испытуемых был проведен сравнительный анализ, определяющий области знаний с наибольшими затруднениями, и выявлен фиксированный процент погрешностей, при котором может проявляться когнитивное нарушение.

**5.3 Анализ результатов**

Проведение диагностики заболеваний нервной системы, с помощью технологии Kinect, дало следующие результаты, описанные в Таблице 1.

Таблица 1.

Процент выявления нарушений рук и ног у испытуемых (П – правая конечность, Л – левая конечность)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Проект | Поднятие рук | | Движение руками по сторонам | | Удержание рук | | Прорисовывание круга | | Удержание равновесия | |
| П | Л | П | Л | П | Л | П | Л | П | Л |
| Kinect Explorer | 20 | 10 | 10 | 10 | 20 | 20 | 20 | 30 | 30 | 20 |
| Kinect DTW | 10 | 10 | 10 | 10 | 10 | 20 | 20 | 20 | 20 | 20 |

У 80% испытуемых не было выявлено явных двигательных нарушений, в связи с этим было предпринято в качестве отклонений принимать нечеткое выполнение, задержку движений, потерю координации.

Для осуществления фиксации в самой программе, необходимо было получить информацию о фигурах в кадре через включение нужного потока:

* Подписаться на событие с координатами найденных фигур в кадре:

sensor.SkeletonFrameReady += SkeletonsReady;

* Включить поток:

sensor.SkeletonStream.Enable();

Одна фигура – это один объект класса Skeleton.

Joints – это информация о точках скелета.

JointType – это перечисление точек скелета.

Для получения местоположения определенной точки скелета (например, левое запястье) необходимо выполнить следующую операцию:

Joint wristLeft = skeleton.Joints[JointTypeWristLeft];

int x = wristLeft.Position.X;

int y = wristLeft.Position.Y;

int z = wristLeft.Position.Z;

Таким образом, осуществляется получение всех шаблонов отклонения.

Шаблоны отклонения фиксировали такие точки скелета, как:

* Правый локоть, правое запястье, правое плечо;
* Левый локоть, левое запястье, левое плечо;
* Позвоночник.

Проведение диагностики когнитивных отклонений позволило определить процентный уровень, на котором можно фиксировать возможное проявление нарушения умственной деятельности. Так из 127 вопросов, на которые должен ответить испытуемый, при наличии 20% ошибок (приблизительно 25 вопросов) можно предполагать возможное проявление нарушений умственной деятельности.

При рассмотрении результатов всех испытуемых, средний показатель ошибок составил 24,7%. Количество испытуемых, с процентным соотношением ошибок больше 20%, составило 14 человек. Таким образом, можно утверждать, что у половины всех испытуемых было выявлено проявление умственных нарушений. Поскольку разработкой вопросов для тестирования занималось отделение клинической неврологии и клинической лингвистики, которые на практике постоянно проводят подобное тестирование, можно утверждать, что полученные в ходе исследования результаты являются корректными и точными.

При сопоставлении результатов диагностики двигательных нарушений и когнитивных отклонений, у четверых испытуемых были выявлены ошибки в обоих тестированиях. Можно утверждать, что у этих испытуемых, возможно, начинают проявляться заболевания нервной системы. Однако с точностью утверждать подобное нельзя, поскольку диагностирование заболеваний нервной системы является сложным процессом, а проводимое в ходе дипломной работы исследование является лишь пробным шагом для развития полноценной системы диагностирования.

Исследование позволило определиться с дальнейшими путями развития проекта и выявило ошибки, нуждающиеся в доработке:

* определены значительные недостатки, связанные с ручной обработкой результатов;
* шаблоны отклонений являются не полными, поскольку проводились на здоровых людях, а не на пациентах;
* отсутствует автоматическое сопоставление двигательных нарушений с когнитивными отклонениями;
* отсутствует программная реализация, способная сопоставлять результаты с базой данных.

Таким образом, дальнейшее развитие исследования возможно при проведении тестирований над реальными пациентами, это позволит собрать обширную базу шаблонов отклонений, и в дальнейшем реализовать целую систему, адаптированную под технологию Kinect, способную проводить диагностики без участия врача.

# ЗАКЛЮЧЕНИЕ

В результате выполнения дипломной работы были проведены исследования в области диагностирования заболеваний нервной системы, с помощью сенсорных технологий считывания движений.

В процессе выполнения дипломной работы были осуществлены следующие задачи:

* Рассмотрены существующие технологии, способные отслеживать движения и жесты, проанализирован их функционал;
* Проведено тестирование над испытуемыми в области диагностирования двигательных нарушений, с помощью технологии Kinect;
* Разработано мобильное приложение «Diagnostics» под OS Android для выявления когнитивных отклонений;
* Проведено тестирование над испытуемыми, с помощью мобильного приложения, на выявление когнитивных отклонений;
* Получены и проанализированы результаты тестирований;
* Определены способы дальнейшего улучшения проекта, для наиболее эффективного диагностирования испытуемых.

По результатам проведенных исследований было выявлено, что возможность использования сенсорных технологий считывания движений для диагностирования заболеваний нервной системы является реальной. Однако к данной технологии необходимо прилагать еще и методы диагностики когнитивных отклонений, поскольку выявление только двигательных нарушений, не может составить полную картину неврологического заболевания.

В ходе выполнения дипломной работы экспериментально были выявлены недостатки исследования, связанные с проведением тестирования на здоровых испытуемых. Однако данные недостатки в исследовании можно устранить, в случае проведения тестирований в медицинских учреждениях.

Прямой доступ к медицинским учреждениям позволит на основе реальных пациентов создать обширную базу шаблонов отклонений. Осуществить попытки реализации автоматического сопоставления тестирования пациента с созданной ранее базой шаблонов отклонений.

Таким образом, возможно создание целой системы под технологию Kinect, способной проводить точные диагностики заболеваний нервной системы, что обеспечит условия для дальнейшего сотрудничества с медицинскими организациями, и возможность внедрения проекта в лечебно-диагностический процесс.

**ГЛОССАРИЙ**

**Android Studio** – интегрированная среда разработки (IDE) для работы с платформой Android [16].

**DTW (Dynamic Time Warping)** – алгоритм динамической трансформации временной шкалы. Алгоритм, позволяющий находить оптимальное соответствие между временными последовательностями [17].

**IDE (Integrated development environment)** – комплекс программных средств, используемый программистами для разработки программного обеспечения [18].

**Нейросонография** – ультразвуковая диагностика головного мозга новорожденного ребенка [8].

**NITE -** промежуточное программное обеспечение, решающее задачи по определению жестов, отслеживанию скелета [15].

**NUI (Natural User Interface)** –естественный пользовательский интерфейс, позволяющий пользователю визуально работать с системой, при помощи жестов, а также при помощи голосовых команд [15].

**OpenNI (Open Natural Interaction)** –программное обеспечение с открытым исходным кодом, ориентированное на улучшение взаимодействия с пользовательскими интерфейсами и высокоуровневыми программами [15].

**SDK (Software Development Kit)** –комплект средств разработки, который позволяет специалистам по программному обеспечению создавать приложения для определённого пакета программ и прочих платформ [19].

**Windows API****(Application Programming Interfaces)** – набор базовых функций интерфейсов программирования приложений ОС семейств Microsoft Windows [20].

**Электромиография** –метод исследования биоэлектрических потенциалов, регистрация электрической активности мышц [21].

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Дистрофические изменения [Электронный ресурс] / Здоровье суставов. Позвоночник – Режим доступа: http://systawy.ru/pozvonochnik/distroficheskie-izmenenija-pozvonochnika.html, свободный.
2. Деменция [Электронный ресурс] / Википедия. – Режим доступа: https://ru.wikipedia.org/wiki/Деменция, свободный.
3. Нервная система [Электронный ресурс] / Википедия. – Режим доступа: https://ru.wikipedia.org/wiki/Нервная\_система, свободный.
4. Нервные болезни [Электронный ресурс] / РТФ.рф. – Режим доступа: http://ru-transferfactor.ru/nervnyie-bolezni, свободный.
5. Неврология [Электронный ресурс] / Википедия. – Режим доступа: https://ru.wikipedia.org/wiki/Неврология, свободный.
6. Суслина З.А., Иллариошкин С.Н., Пирадов М.А. Неврология и нейронауки – прогноз развития [Текст]: Анн. клинич. и эксперим. неврол. – 2007. – 1(1). – С 5–9.
7. Болезни нервной системы [Электронный ресурс] / Medical Journal. – Режим доступа: http://www.medicalj.ru/diseases/neurology, свободный.
8. Дегенерации нервов: группа наследственных и приобретенных заболеваний [Электронный ресурс] / Медицинский портал «Твой Айболит». – Режим доступа: http://tvoyaybolit.ru/degeneracii-nervov-gruppa-nasledstvennyx-i-priobretennyx-zabolevanij.html, свободный.
9. Технологии управления устройствами взглядом и жестами [Электронный ресурс] / Fotokomok. – Режим доступа: http://www.fotokomok.ru/texnologii-upravleniya-ustrojstvami-vzglyadom-i-zhestami/, свободный.
10. Ларионов В. Технологии отслеживания движений [Электронный ресурс] / HiNews.ru – новости высоких технологий. – Режим доступа: https://hi-news.ru/gadgets/texnologiya-hauoli-pozvolyaet-otslezhivat-dvizheniya-po-zvukam.html, свободный.
11. Хижникова А.Е., Клочков А.С., Котов-Смоленский А.М., Супонева Н.А., Черникова Л.А. Виртуальная реальность как метод восстановления двигательной функции руки [Текст]: Клиническая неврология. – 2016. – С 4-5.
12. Чернецова М. Пять гаджетов для управления жестами [Электронный ресурс] / the Runet beta – средства массовой коммуникации. – Режим доступа: https://therunet.com/articles/4514, свободный.
13. Kinect [Электронный ресурс] / Википедия. – Режим доступа: https://ru.wikipedia.org/wiki/Kinect, свободный.
14. Оборудование Kinect [Электронный ресурс] / Microsoft. Центр разработки для Windows. – Режим доступа: https://developer.microsoft.com/ru-ru/windows/kinect/hardware, свободный.
15. Кичинский К. Kinect for Windows SDK вышел в плавание [Электронный ресурс] / Microsoft. Developer. – Режим доступа: https://blogs.msdn.microsoft.com/kichinsky/2011/06/19/kinect-for-windows-sdk/, свободный.
16. Android Studio [Электронный ресурс] / Википедия. – Режим доступа: https://ru.wikipedia.org/wiki/Android\_Studio, свободный.
17. DTW [Электронный ресурс] / Википедия. – Режим доступа: https://ru.wikipedia.org/wiki/Алгоритм\_динамической\_трансформации\_временной \_шкалы, свободный.
18. Интегрированная среда разработки [Электронный ресурс] / Википедия. – Режим доступа: https://ru.wikipedia.org/wiki/IDE, свободный.
19. SDK [Электронный ресурс] / Википедия. – Режим доступа: https://ru.wikipedia.org/wiki/SDK, свободный.
20. Windows API [Электронный ресурс] / Википедия. – Режим доступа: https://ru.wikipedia.org/wiki/Windows\_API, свободный.
21. Электромиография [Электронный ресурс] // Википедия. – Режим доступа: https://ru.wikipedia.org/wiki/Электромиография, свободный.

**ПРИЛОЖЕНИЕ А**

Класс начального запуска MainActivity.java

package com.example.diagnostics;

import android.content.Intent;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.Button;

import android.widget.Toast;

public class MainActivity extends AppCompatActivity implements OnClickListener{

private Button flora;

private Button fizik;

private Button chimia;

private Button geog;

private Button liter;

private Button mif;

private Button arif;

private Button music;

private Button kino;

private Button znania;

@Override

protected void onCreate(Bundle savedInstanceState) {

try{

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_test);

flora=(Button)findViewById(R.id.button22);

flora.setOnClickListener(this);

fizik=(Button)findViewById(R.id.button21);

fizik.setOnClickListener(this);

chimia=(Button)findViewById(R.id.button23);

chimia.setOnClickListener(this);

geog=(Button)findViewById(R.id.button19);

geog.setOnClickListener(this);

liter=(Button)findViewById(R.id.button16);

liter.setOnClickListener(this);

mif=(Button)findViewById(R.id.button17);

mif.setOnClickListener(this);

music=(Button)findViewById(R.id.button14);

music.setOnClickListener(this);

kino=(Button)findViewById(R.id.button15);

kino.setOnClickListener(this);

znania=(Button)findViewById(R.id.button20);

znania.setOnClickListener(this);

arif=(Button)findViewById(R.id.button18);

arif.setOnClickListener(this);

}

catch(Exception e){

Toast.makeText(this, e+"", Toast.LENGTH\_LONG).show();

}

}

public void onClick(View v) {

switch (v.getId()){

case R.id.button20:

Intent intent=new Intent(this, Znania.class);

startActivity(intent);

break;

case R.id.button22:

Intent intent1=new Intent(this, Flora.class);

startActivity(intent1);

break;

case R.id.button21:

Intent intent2=new Intent(this, Fizik.class);

startActivity(intent2);

break;

case R.id.button23:

Intent intent3=new Intent(this, Chimia.class);

startActivity(intent3);

break;

case R.id.button14:

Intent intent4=new Intent(this, Music.class);

startActivity(intent4);

break;

case R.id.button15:

Intent intent5=new Intent(this, Kino.class);

startActivity(intent5);

break;

case R.id.button18:

Intent intent6=new Intent(this, Arif.class);

startActivity(intent6);

break;

case R.id.button19:

Intent intent7=new Intent(this, Geog.class);

startActivity(intent7);

break;

case R.id.button16:

Intent intent8=new Intent(this, Liter.class);

startActivity(intent8);

break;

case R.id.button17:

Intent intent9=new Intent(this, Mif.class);

startActivity(intent9);

break;

}

}

}

Класс выбора направления тестирования NewActivity1.java

package com.example.diagnostics;

import android.content.Intent;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.Button;

import android.widget.TextView;

import android.widget.Toast;

public class NewActivity1 extends AppCompatActivity implements OnClickListener{

private Button newtest;

private Button stat;

private TextView time1;

private TextView n;

private TextView k;

private TextView m;

private TextView ques;

private TextView ans;

@Override

protected void onCreate(Bundle savedInstanceState) {

try{

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_new1);

newtest=(Button)findViewById(R.id.button7);

newtest.setOnClickListener(this);

time1=(TextView)findViewById(R.id.textView12);

n=(TextView)findViewById(R.id.textView5);

k=(TextView)findViewById(R.id.textView10);

m=(TextView)findViewById(R.id.textView11);

ques=(TextView)findViewById(R.id.textView14);

ans=(TextView)findViewById(R.id.textView13);

k.setText(getIntent().getStringExtra("k"));

n.setText(getIntent().getStringExtra("n"));

m.setText(getIntent().getStringExtra("m"));

time1.setText(getIntent().getStringExtra("time1"));

ques.setText(getIntent().getStringExtra("que"));

ans.setText(getIntent().getStringExtra("ans"));

}

catch(Exception e){

Toast.makeText(this, e+"", Toast.LENGTH\_LONG).show();

}

}

public void onClick(View v) {

switch (v.getId()){

case R.id.button7:

Intent intent=new Intent(this, MainActivity.class);

startActivity(intent);

break;

}

}

}

Один из классов тестирования Arif.java

package com.example.diagnostics;

import android.app.Activity;

import android.content.Intent;

import android.content.res.Resources;

import android.content.res.TypedArray;

import android.os.Bundle;

import android.view.Display;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.AbsoluteLayout;

import android.widget.Button;

import android.widget.Chronometer;

import android.widget.EditText;

import android.widget.RadioGroup;

import android.widget.TextView;

import android.widget.Toast;

public class Arif extends Activity implements OnClickListener {

private Button Question;

private Button exit;

private EditText edit;

private Button[] Answers=new Button[3];

private double width;

private double height;

private static int QUESTIONS=32;

private static final int VARIANTS=3;

private static final char DELIMITER='/';

private String[][] AnsMatrix=new String[VARIANTS][QUESTIONS];

private int[] RightAnswers=new int[QUESTIONS];

private String[] Ques=new String[QUESTIONS];

private TypedArray Base;

private int time=0;

private int right=0;

private int current\_right=0;

private TextView time1;

private TextView n;

private TextView k;

private TextView m;

private TextView ok;

private String ques1;

private int number=-1;

private Chronometer mChronometer;

private TextView ques;

private TextView ans;

private int z=0;

String que="\nВопросы:"+"\n";

String answ="\nБалл за ответ:"+"\n";

private AbsoluteLayout.LayoutParams[] params=new AbsoluteLayout.LayoutParams[VARIANTS+1];

@Override

public void onCreate(Bundle savedInstanceState) {

try{

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_arif);

Question=(Button)findViewById(R.id.button2);

Answers[0]=(Button)findViewById(R.id.button);

Answers[1]=(Button)findViewById(R.id.button3);

Answers[2]=(Button)findViewById(R.id.button4);

exit=(Button)findViewById(R.id.button5);

for (int i=0;i<VARIANTS;i++){

Answers[i].setOnClickListener(this);

}

exit.setOnClickListener(this);

Question.setOnClickListener(this);

Display display = getWindowManager().getDefaultDisplay();

width=display.getWidth();

height=display.getHeight();

time1=(TextView)findViewById(R.id.textView12);

n=(TextView)findViewById(R.id.textView5);

k=(TextView)findViewById(R.id.textView10);

m=(TextView)findViewById(R.id.textView11);

ques=(TextView)findViewById(R.id.textView14);

ans=(TextView)findViewById(R.id.textView13);

mChronometer=(Chronometer)findViewById(R.id.chronometer4);

mChronometer.start();

LoadQuestions();

LoadQuestion();

}

catch(Exception e){

Toast.makeText(this, e+"", Toast.LENGTH\_LONG).show();

}

}

private void LoadQuestion() {

number++;

if (number < 32) {

Question.setText(Ques[number]);

for (int i = 0; i < VARIANTS; i++) {

Answers[i].setText(AnsMatrix[i][number]);

}

current\_right = RightAnswers[number] - 1;

}

}

private void LoadQuestions(){

Resources res=getResources();

Base=res.obtainTypedArray(R.array.Arif);

for (int i=0;i<QUESTIONS;i++){

Ques[i]=getSubstringBetweenDelimiters(0,1,Base.getString(i));

que+=Ques[i]+"\n";

for (int j=0;j<VARIANTS;j++){

AnsMatrix[j][i]=getSubstringBetweenDelimiters(j+1,j+2,Base.getString(i));

}

RightAnswers[i]=Integer.parseInt(getSubstringBetweenDelimiters(4,5,Base.getString(i)));

}

}

private String getSubstringBetweenDelimiters(int k, int m, String str){

int index1=0;

int index2=0;

int len=str.length();

int dels=0;

for (int i=0;i<len;i++){

if (str.charAt(i)==DELIMITER){

dels++;

}

if (dels==k){

index1=i;

}

if (dels==m){

index2=i;

}

}

return str.substring(index1+2, index2+1);

}

@Override

public void onClick(View arg0) {

if (arg0==Question){

wrong++;

answ+=0+"\n";

}

else{

wrong++;

for (int i=0;i<VARIANTS;i++){

if (arg0==Answers[i]){

if (current\_right==i){

wrong--;

right++;

answ+=1+"\n";

}

else {

answ+=0+"\n";

}

}

}

}

time++;

LoadQuestion();

int t=time;

int r=right;

if (number==32){

Intent intent=new Intent(this, NewActivity1.class);

int N=32;

intent.putExtra("k",String.valueOf(t).toString());

intent.putExtra("n",String.valueOf(N).toString());

intent.putExtra("m",String.valueOf(r).toString());

intent.putExtra("time1",mChronometer.getText().toString());

intent.putExtra("que",que.toString());

intent.putExtra("ans",answ.toString());

startActivity(intent);

}

switch (arg0.getId()){

case R.id.button5:

Intent intent=new Intent(this, NewActivity1.class);

time--;

int N=32;

String stat="";

stat+=getString(R.string.note1);

stat+=" "+time+" ";

stat+=getString(R.string.note2);

stat+=" "+right+". ";

intent.putExtra("k",String.valueOf(time).toString());

intent.putExtra("n",String.valueOf(N).toString());

intent.putExtra("m",String.valueOf(right).toString());

intent.putExtra("time1",mChronometer.getText().toString());

intent.putExtra("que",que.toString());

intent.putExtra("ans",answ.toString());

Toast.makeText(this, stat, Toast.LENGTH\_LONG).show();

startActivity(intent);

break;

}

}

private void Stats() {

String stat="";

stat+=getString(R.string.note1);

stat+=" "+time+" ";

stat+=getString(R.string.note2);

stat+=" "+right+". ";

Toast.makeText(this, stat, Toast.LENGTH\_LONG).show();

}

}