**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**Факультет Программной инженерии и компьютерной техники**

**КУРСОВАЯ РАБОТА**

**Тема:** Разработать мобильное приложение, осуществляющее сбор данных о пульсе пользователя и его вариабельности, включающее игру, скорость которой зависит от индекса напряженности

**Работу выполнил студент** Кузьмичева Ксения Евгеньевна **группы** P3322

(фамилия, имя, отчество) (номер группы)

**Руководитель**  Штенников Дмитрий Геннадьевич

(фамилия, имя, отчество)

**Работа защищена** " " 2020 г.

**c оценкой**

**Подписи членов комиссии:**

**САНКТ-ПЕТЕРБУРГ, 2020**

Содержание

[Введение 3](#_Toc43403233)

[Глава 1. Обзорная 4](#_Toc43403234)

[1.1. Анализ требований к программной системе 4](#_Toc43403235)

[1.2. Доопределение задачи 4](#_Toc43403236)

[1.3. Описание технологий реализации 6](#_Toc43403237)

[1.4. Обзор аналогов 6](#_Toc43403238)

[Глава 2. Проектная 8](#_Toc43403239)

[2.1. Функциональная модель в стандарте IDEF0 8](#_Toc43403240)

[2.2. Сценарий использования 10](#_Toc43403241)

[2.3. Диаграмма классов 11](#_Toc43403242)

[2.4. Модель данных 12](#_Toc43403243)

[Глава 3. Реализация 13](#_Toc43403244)

[3.1. Пользовательский интерфейс 13](#_Toc43403245)

[3.2. Алгоритмы работы приложения 14](#_Toc43403246)

[3.3. Тестирование 18](#_Toc43403247)

[Заключение 20](#_Toc43403248)

[Библиографический список 21](#_Toc43403249)

# Введение

Многие ученые считают, что будущее за превентивной медициной: наши гаджеты будут собирать достаточное количество информации о показателях здоровья, чтобы можно было начать принимать меры еще до того, как появится реальная проблема. Однако гаджеты в настоящее время не могут производить столь же точные замеры, какие делаются в лабораториях и клиниках на специализированном оборудовании. Тем не менее некоторые показатели уже можно измерять и с помощью смартфона с достаточной точностью. Например – вариабельность сердечного ритма (ВСР).

ВСР – это показатель, отражающий неравномерность сердцебиения. Чем больше неравномерность, тем лучше функциональное состояние организма. В отличие от пульса, который в большей степени отражает реакцию организма на физическую активность, ВСР учитывает также ментальную и эмоциональную нагрузку. Использовать ВСР начали еще в 1960-х как неинвазивный способ измерения уровня стресса в организме, оценки функционального состояния, риска заболеть и других параметров. Изначально ВСР применяли, чтобы следить за самочувствием астронавтов. Но вариабельность оказалась таким всеобъемлющим показателем, что вскоре изучать ее стали и за пределами NASA.

Большим прорывом стало появление новых технологий – в частности смартфонов с камерой и вспышкой. Приложив палец к камере со включенной вспышкой можно фиксировать каждый удар сердца, так как приток крови затемняет кожу.

Целью данной курсовой работы является разработка игры для мобильных устройств на платформе Android, учитывающей текущее функциональное состояние пользователя и корректирующей его при необходимости.

# Обзорная

# Анализ требований к программной системе

В рамках настоящей курсовой работы необходимо разработать игру для мобильных устройств на платформе Android, учитывающую текущее функциональное состояние пользователя и корректирующую его при необходимости.

Приложение должно осуществлять сбор данных о пульсе пользователя и его вариабельности, используя вспышку и камеру мобильного устройства в качестве сенсора. Скорость игрового процесса должна варьироваться в зависимости от регистрируемого состояния пользователя с целью его нормализации.

Состояние пользователя оценивается по показателю ВСР - индексу напряжения регуляторных систем (ИН), характеризующему активность механизмов симпатической регуляции, состояние центрального контура регуляции. Этот показатель вычисляется на основании анализа распределения кардиоинтервалов. Активация центрального контура, усиление симпатической регуляции во время психических или физических нагрузок проявляется стабилизацией ритма, уменьшением разброса длительностей кардиоинтервалов, увеличением количества однотипных по длительности интервалов.

# Доопределение задачи

Исходя из требований к программному продукту мною был создан следующий концепт приложения под названием KCaKC, расшифровывающийся как Keep Calm and Kill Cockroaches.

При запуске приложения включается камера и вспышка телефона. Пользователю необходимо приложить палец к камере, чтобы начать считывать пульс. Как только собирается достаточно данных, на экране устройства должны отображаться числовые показатели пульса (уд. /мин) и ИН (усл. ед.). Для удобства восприятия числовых характеристик, исходя из показателя индекса напряженности, должен изменяться цвет фона экрана и появляться текстовое описание текущего состояния пользователя. Также на экране должны присутствовать индикатор ударов сердца в виде лампочки, мигающей при каждом ударе, и кнопка для начала игры.

По началу игры сверху вниз начинают ползти тараканы. Скорость их движения и генерации зависит от текущего индекса напряженности. По нажатии на таракана он умирает и, затем, исчезает. При отведении пальца от камеры игра останавливается и продолжается с того же места, как только пользователь вновь приложит палец к камере. Идея игры в том, чтобы убить как можно больше тараканов и не дать ни одному таракану доползти до нижней границы экрана. Если это все же произошло, игра завершается и пользователю выводится диалоговое окно с набранным рейтингом и поздравлением, если был побит рекорд. Далее пользователь может либо сразу начать новую игру, либо вернуться на начальный экран.

Во время игры пользователю должны отображаться текущий рейтинг и рекорд. Рейтинг равен количеству убитых тараканов за одну игру. Рекорд – максимальный рейтинг за все время, сохраняющийся при закрытии приложения.

В Таблице 1 приведена зависимость параметров игры от индекса напряженности (ИН). Цвета фона выбраны в соответствии со смысловыми значениями сигнальных цветов, чтобы пользователь интуитивно понимал, в каком состоянии находится, не читая описание. Скоростные параметры подобраны таким образом, чтобы:

* играть было наиболее комфортно в нормальном состоянии,
* при пониженном ИН игра ускорялась и, тем самым, требовала от пользователя более быстрой реакции с целью взбодрить его,
* при повышенном ИН игра замедлялась и расслабляла пользователя.

Таким образом будет происходить корректировка функционального состояния пользователя для приведения его в нормальное состояние.

Таблица 1 Зависимость параметров игры от ИН

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ИН, усл. ед.** | **Цвет фона** | **Описание состояния** | **Скорость передвижения тараканов, % от максимально значения** | **Период генерации тараканов, с** |
| от 1 до 10 | светло-красный | Резервы организма исчерпаны | 100% | 0.3 |
| от 10 до 20 | желтый | Вы на пределе | 90% | 0.35 |
| от 20 до 120 | зеленый | Норма | 80% | 0.4 |
| от 120 до 200 | желтый | Верхняя граница нормы | 70% | 0.45 |
| от 200 до 500 | оранжевый | Выше нормы | 60% | 0.5 |
| от 500 до 800 | светло-красный | Очень высокий | 50% | 0.55 |
| более 800 | темно-красный | Ваше сердце работает на износ | 30% | 0.6 |

# Описание технологий реализации

Для реализации мобильного приложения KCaKC под платформу Android мною был выбран язык Kotlin v.1.3.72 и SDK v.29. Для измерения ЧСС с помощью камеры и вспышки была использована библиотека Heart-Rate-Ometer v.1.3.2, выложенная в GitHub пользователем kibotu. В качестве системы сборки использовался Gradle v.5.6.4.

# Обзор аналогов

Показатели вариабельности рассчитывают различные приложения. Среди них, к примеру, Welltory – один из лучших российских сервисов в этой сфере. Бесплатный тариф позволяет измерять стресс, энергию и продуктивность. Замер пульса длится 2 минуты, в течение которых требуется не отрывать палец от камеры. После чего выводятся результаты измерений и сохраняются в истории. Приложение выдает различные рекомендации пользователю на основе рассчитанных показателей его состояния. Скриншоты программы после проведения измерений приведены на Рисунках 1 и 2.

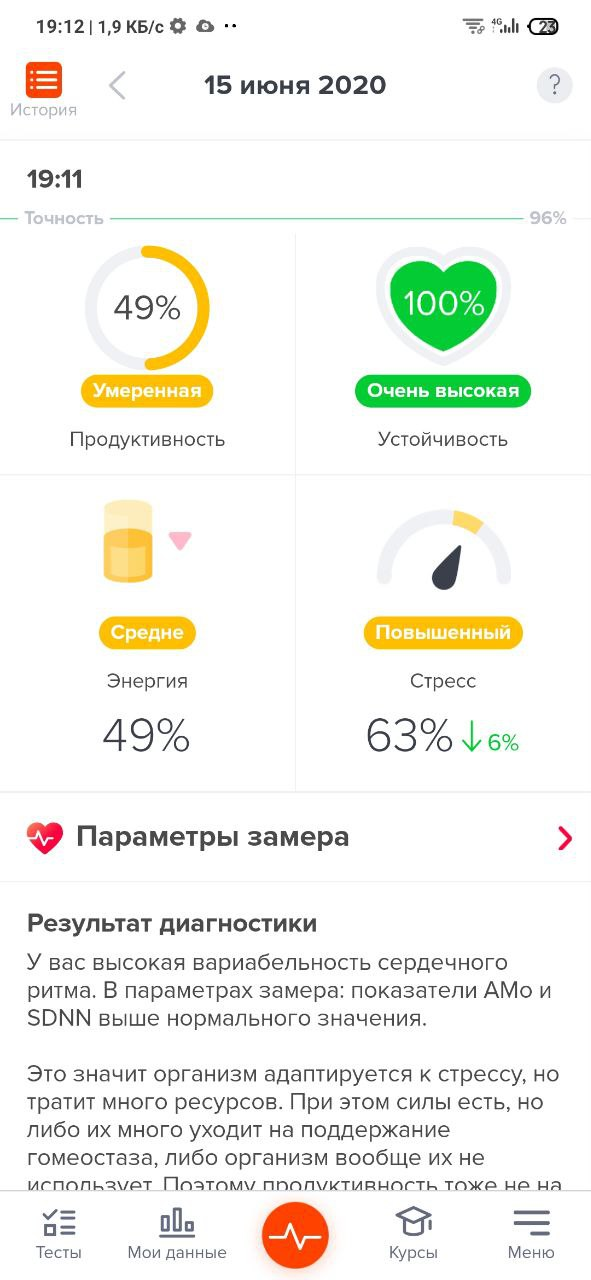
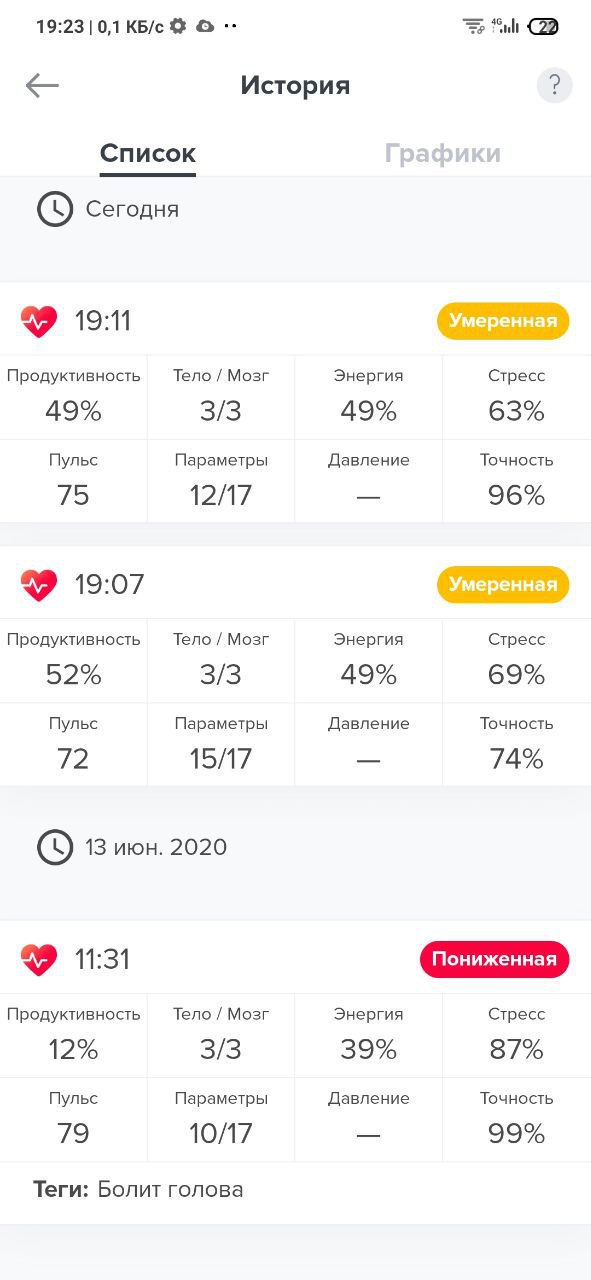
 

Рис. 1 Скриншот результата измерения ЧСС в приложении Welltory

Рис. 2 Скриншот истории измерений ЧСС в приложении Welltory

Приложением со схожим с Welltory функционалом является Engy. Оно создано совместно с научной командой Института ИМБП РАН, отвечающего за медико-биологическую поддержку полетов российских космонавтов. В его основе также лежит методика анализа ВСР. Однако данное приложение работает только в паре с пульсометром.

Приведенные аналоги позволяют быстро и достаточно точно оценить функциональное состояние пользователя. Но, в отличие от KCaKC они не имеют элемента игры и никак не влияют на состояние пользователя, а только дают рекомендации о том, как его улучшить.

Также аналогами будут являться игры аркадного жанра. Наиболее похожей на KCaKC оказалась игра Ant Smasher. В ней более разнообразный геймплей, но в бесплатной версии обилие рекламы сильно мешает игровому процессу и делает его практически невозможным при подключении к Интернету. Скриншот игры приведен на Рисунке 3.



Рис. 3 Скриншот игры Ant Smasher

Аналогов, совмещающих в себе, и замер ЧСС с его последующим анализом, и игровой процесс, стабилизирующий функциональное состояние пользователя, мною не было найдено.

# Проектная

# Функциональная модель в стандарте IDEF0

Для описания совокупности выполняемых системой функций используется IDEF0-диаграмма. Функциональная направленность означает, что функции системы исследуются независимо от механизмов, которые обеспечивают их выполнение. В целом такой подход направлен на изучение того, что делает исследуемая система, а не каким конкретно способом. Это позволяет более четко смоделировать логику и взаимодействие процессов системы.

Основу методологии IDEF0 составляет графический язык описания процессов. Модель в нотации IDEF0 представляет собой совокупность иерархически упорядоченных и взаимосвязанных диаграмм. Каждая диаграмма является единицей описания системы и располагается на отдельном листе. На Рисунке 4 представлена контекстная диаграмма игры с точки зрения пользователя и на Рисунке 5 ее декомпозиция.

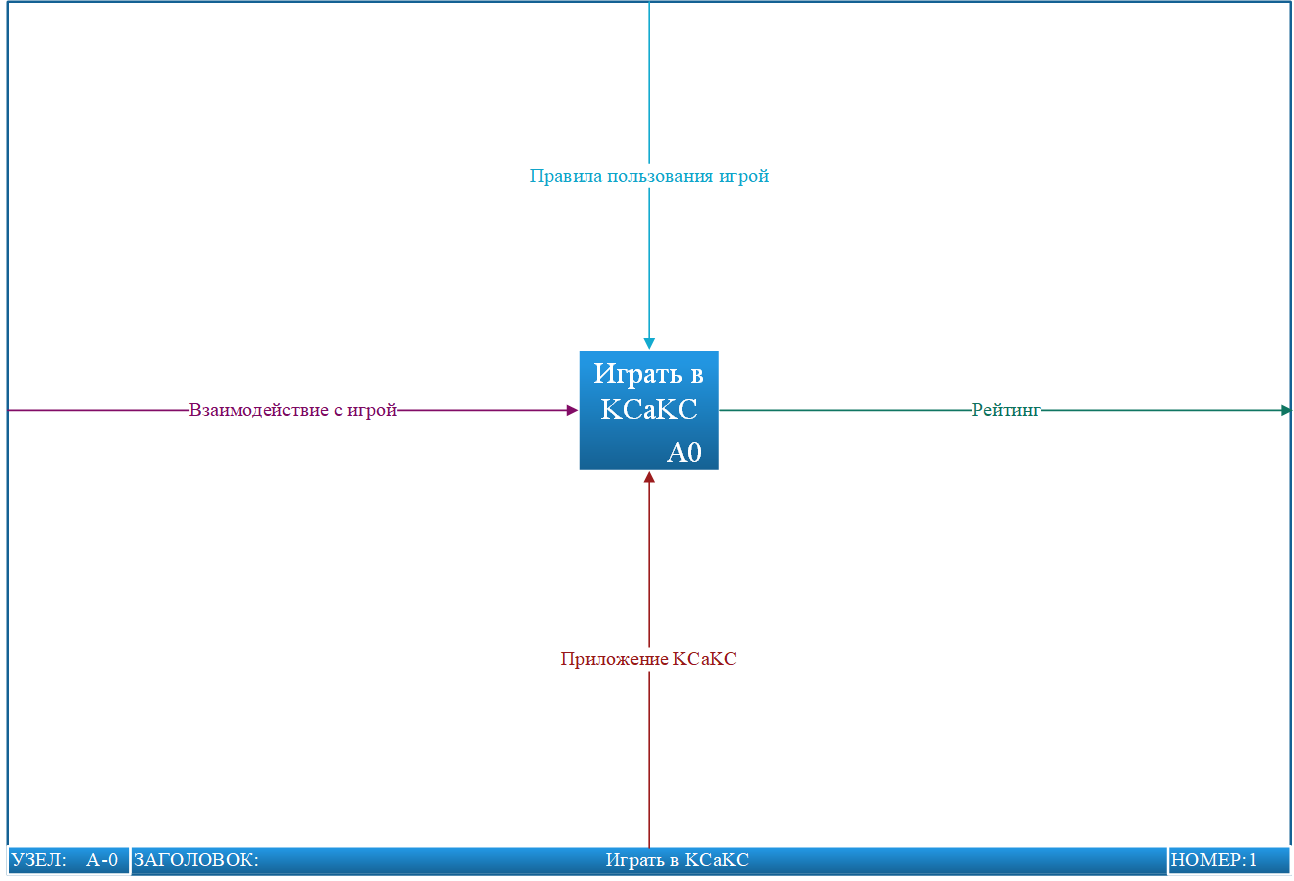


Рис. 4 Обобщенная модель IDEF0 игры KCaKC с точки зрения пользователя

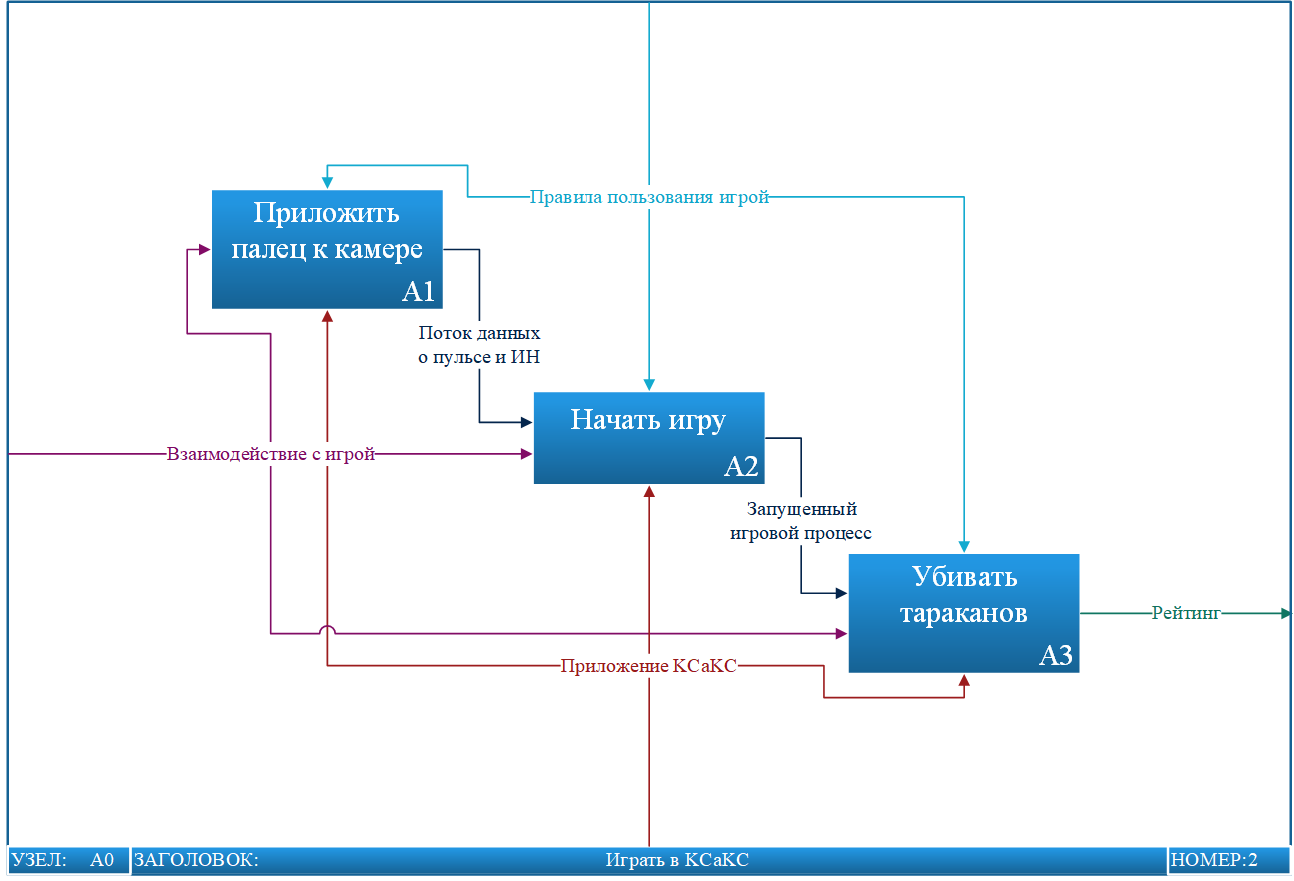


Рис. 5 Детализированная модель IDEF0 игры KCaKC с точки зрения пользователя

# Сценарий использования

Для наглядного представления требований к создаваемой системе используются диаграммы вариантов использования. Вариант использования представляет собой последовательность действий (транзакций), выполняемых системой в ответ на событие, инициируемое пользователем. Вариант использования описывает типичное взаимодействие между пользователем и системой и отражает представление о поведении системы с точки зрения пользователя.

Цель построения диаграмм вариантов использования – документирование функциональных требований к системе в самом общем виде. Диаграмма вариантов использования стандарта UML для системы KCaKC приведена на Рисунке 6.

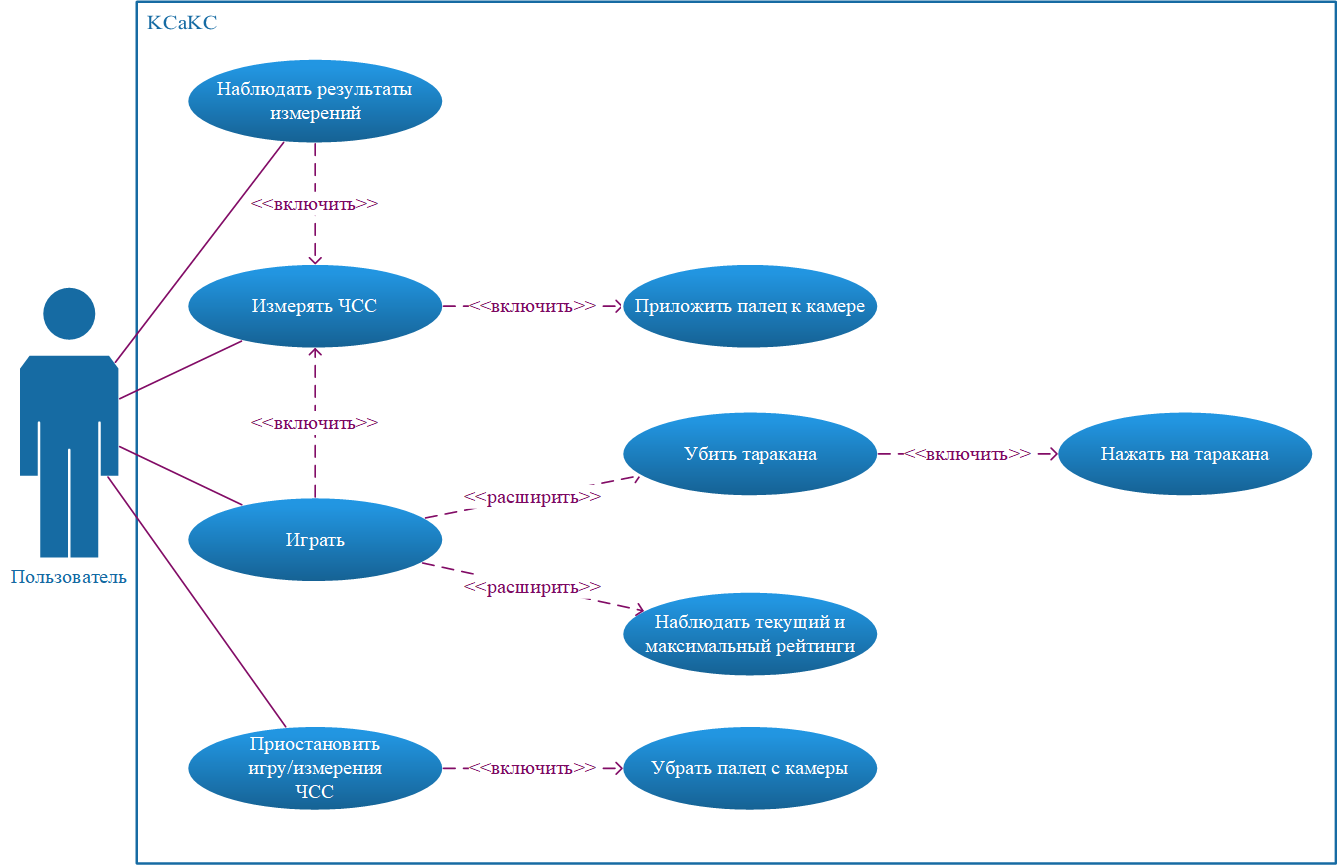


Рис. 6 Диаграмма вариантов использования системы KCaKC

# Диаграмма классов

Диаграмма классов описывает типы объектов системы и различного рода статические отношения, которые существуют между ними. На диаграммах классов отображаются также свойства классов, операции классов и ограничения, которые накладываются на связи между объектами.

Целью создания диаграммы классов является графическое представление статической структуры декларативных элементов системы и операций над ними. Для удобства восприятия дополним диаграмму классов представлением пакетов. На Рисунке 7 приведена диаграмма классов приложения KCaKC в нотации UML.

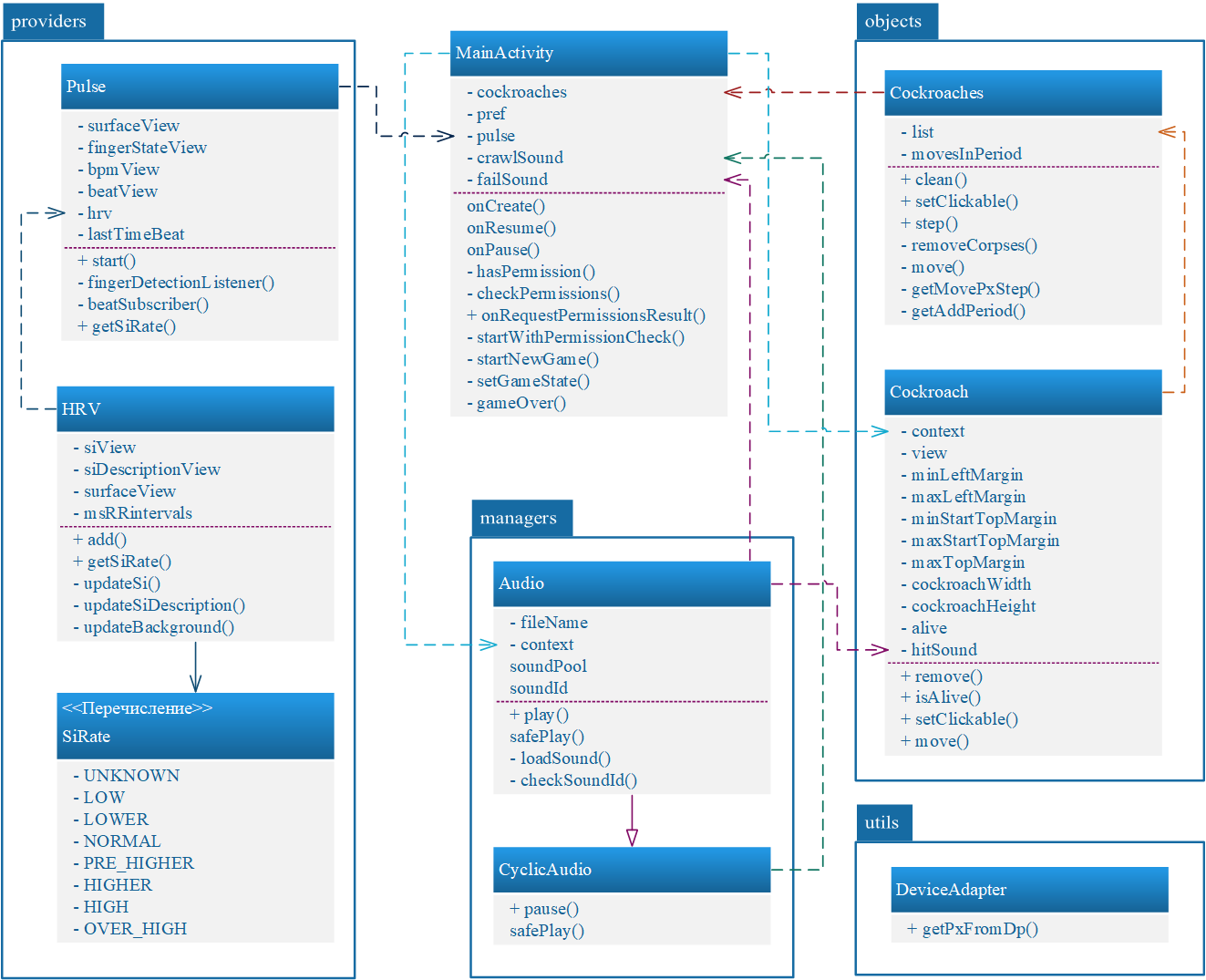


Рис. 7 Диаграмма классов приложения KCaKC

# Модель данных

Приложение KCaKC предполагает сохранение максимально набранного рейтинга при выходе из приложения. Никаких других данных хранить в памяти устройства не требуется. Исходя из этого, мною был сделан вывод о том, что хранить лишь одно значение в базе данных не будет оправдано. Для хранения малого количества простых данных, не требующих обработки и анализа можно использовать файлы. При запуске приложения достаточно будет считать необходимые данные из файла.

Для использования файловой структуры хранения данных в Android существует класс SharedPreferences (Общие настройки). Приложение автоматически создает файл в своей папке и хранит простые данные в виде «ключ – значение». Общие настройки поддерживают базовые типы: boolean, string, float, long и int. Будем хранить под ключом "max\_rate" значение типа int с рекордом пользователя по количеству убитых тараканов.

# Реализация

# Пользовательский интерфейс

Расположим элементы пользовательского интерфейса следующим образом. В верхней панели в первой строке будет выводиться значение пульса в уд./мин и там же будет расположена лампочка, загорающаяся при каждом ударе сердца. Во второй строке будет отображаться значение ИН в усл. ед. и краткое описание, соответствующее данному значению. Сразу под панелью будут расположены значения максимального и текущего рейтинга. Они будут отображаться только в режиме игры. Внизу экрана будет располагаться кнопка для начала игры. Также потребуется элемент с текстовым уведомлением о том, что требуется приложить палец к камере. Структура элементов интерфейса приведена на Рисунке 8, а на Рисунках 9 и 10 макеты интерфейса в режимах дизайна и чертежа соответственно.

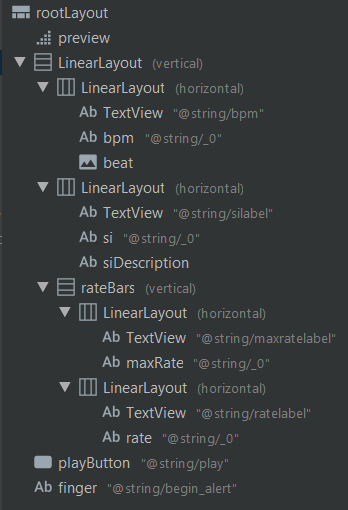


Рис. 8 Дерево компонентов пользовательского интерфейса приложения KCaKC

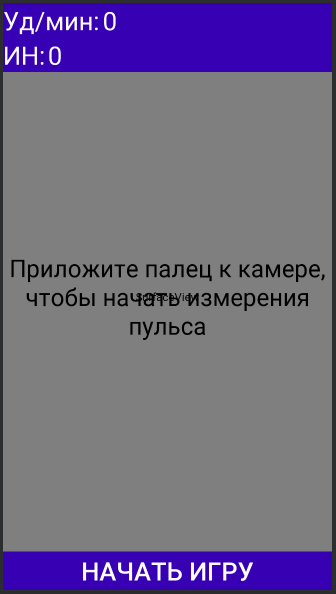
 

Рис. 9 Макет интерфейса приложения KCaKC в режиме дизайна

Рис. 10 Макет интерфейса приложения KCaKC в режиме чертежа

# Алгоритмы работы приложения

При запуске приложения проверяется наличие разрешения на доступ к камере устройства. Если его нет, пользователю выводится диалоговое окно, где он может выдать данное разрешение. Если разрешение есть, появляется окно основной активности приложения, значения показателей пульса и ИН инициализируются нулями, появляется сообщение о необходимости прикладывания пальца к камере, загружаются аудиофайлы в объекты менеджеров по работе со звуком, инициализируется объект пульса, из общих настроек берется значение рекорда. Далее программа ожидает воздействия пользователя.

Метод запуска объекта пульса выглядит следующим образом.

fun start() {  
 val kalman = JKalman(2, 1)  
  
 // measurement [x]  
 val m = Matrix(1, 1)  
  
 // transitions for x, dx  
 val tr = *arrayOf*(*doubleArrayOf*(1.0, 0.0), *doubleArrayOf*(0.0, 1.0))  
 kalman.*transition\_matrix* = Matrix(tr)  
  
 // 1s somewhere?  
 kalman.*error\_cov\_post* = kalman.*error\_cov\_post*.identity()  
  
 HeartRateOmeter()  
 .withAverageAfterSeconds(3)  
 .setFingerDetectionListener **{** fingerDetected **->** if (fingerDetected) {  
 fingerStateView.*text* = ""  
 } else {  
 lastTimeBeat = 0  
 fingerStateView.*text* = "Приложите палец к камере, чтобы продолжить"  
 bpmView.*text* = "0"  
 beatView.setImageResource(-1)  
 }  
 **}** .bpmUpdates(surfaceView)  
 .subscribe(**{** if (**it**.value == 0) {  
 bpmView.*text* = "0"  
 return@subscribe  
 }  
  
 m.set(0, 0, **it**.value.toDouble())  
  
 // state [x, dx]  
 val s = kalman.Predict()  
  
 // corrected state [x, dx]  
 val c = kalman.Correct(m)  
  
 bpmView.*text* = c.get(0, 0).toInt().toString()  
 beatView.setImageResource(if (**it**.type == HeartRateOmeter.PulseType.OFF) -1 else {  
 val timeBeat = System.currentTimeMillis()  
 if(lastTimeBeat != 0L) {  
 val diff = timeBeat - lastTimeBeat  
 if (diff in 400..1600 ) hrv.add(timeBeat - lastTimeBeat)  
 }  
 lastTimeBeat = timeBeat  
 R.drawable.*lamp* })  
  
 **}**, Throwable::printStackTrace)  
}

Данный код взят из примера к библиотеке HeartRateOmeter, о которой говорилось ранее, за исключением некоторых изменений в определении двух основных прослушивающих функции, фиксирующих: приложение пальца к камере и удары сердца соответственно. Первая отвечает за вывод сообщения о необходимости приложить палец. А во второй уже имеется значение пульса и в поле type определяется, совершается ли удар в настоящий момент времени. Так как библиотека уже предоставляет значения пульса, я не пересчитываю его самостоятельно, а рассчитываю только ВСР, а именно ИН. Для подсчета ИН я фиксирую время каждого удара и записываю значения RR-интервалов в мс в список объекта hrv, отбрасывая слишком маленькие и слишком большие значения для повышения точности.

При расчете ИН используется формула Р. М. Баевского, в которой стресс индекс (индекс напряжения регуляторных систем – SI) вычисляется путем деления амплитуды моды на удвоенное произведение моды на размах: . Ниже приведены код вычисления ИН и Таблица 2 с описанием используемых показателей ВСР.

val msMo = msRRintervals.*average*()  
val dx = 200  
val amo =  
 msRRintervals.*filter* **{ it** >= msMo - dx && **it** <= msMo + dx **}**.size.toDouble() \* 100 / msRRintervals.size  
val mxdmn = (msRRintervals.*max*()!! - msRRintervals.*min*()!!).toDouble() / 1000  
val si = (amo / (2 \* mxdmn \* msMo / 1000)).toInt()

Таблица 2 Перечень используемых показателей ВСР

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя переменной** | **Краткое обозначение** | **Наименование показателя** | **Физиологическая интерпретация** |
| mxdmn | MxDMn | Разность между максимальным и минимальным значениями кардиоинтервалов | Максимальная амплитуда регуляторных влияний |
| msMo | Mo | Мода | Наиболее вероятный уровень функционирования сердечно-сосудистой системы |
| amo | Amo | Амплитуда моды | Условный показатель активности симпатического звена регуляции |
| si | SI или ИН | Стресс индекс (индекс напряжения регуляторных систем) | Степень напряжения регуляторных систем (степень преобладания активности центральных механизмов регуляции над автономными) |

Как только прикладывается палец, начинается считывание ударов сердца и сбор статистики. Значения пульса и ИН выводятся с небольшим таймаутом, чтобы набрать необходимое минимальное количество данных для их расчета. Далее в процессе игры сбор статистики продолжается и значения, выводимые на экран, становятся точнее. Одновременно с изменениями значений ИН меняется цвет фона и текстовое описание состояния организма, а также скорость игры, если она запущена.

По нажатии на кнопку «Начать игру» вызывается метод startNewGame().

private fun startNewGame() {  
 playButton.*visibility* = View.*INVISIBLE* rateBars.*visibility* = View.*VISIBLE* val mainHandler = Handler(Looper.getMainLooper())  
 var prevState = false  
 mainHandler.post(object : Runnable {  
 override fun run() {  
 val gameIsOn = bpm.*text*.toString() != "0" && si.*text*.toString() != "0"  
 if (prevState != gameIsOn) setGameState(gameIsOn)  
 cockroaches.setClickable(gameIsOn)  
 if (gameIsOn) {  
 if (!cockroaches.step(  
 pulse.getSiRate(),  
 this@MainActivity  
 )  
 ) return gameOver()  
 }  
 prevState = gameIsOn  
 mainHandler.postDelayed(this, 10)  
 }  
 })  
}

В методе startNewGame() с периодом 10 мс в отдельном потоке запускается функция рендеринга игры. Объект cockroaches хранит список всех живых тараканов. На каждом шаге вызывается метод step этого объекта, который передвигает тараканов. Метод step принимает в параметрах величину ИН, по которой определяет, на какое расстояние за один шаг можно передвинуть тараканов и с каким периодом можно генерировать новых. Возвращается значение типа Boolean, говорящее о том, ни добрался ли хотя бы один таракан до нижней части экрана.

Тараканы появляются на экране как динамически создаваемые объекты ImageView. Их начальное положение и перемещения задаются за счет параметров внешних отступов (margin). При создании таракана для него задается обработчик события нажатия, в котором проигрывается звук удара, сменяется изображение с живого таракана на мертвого, инкрементируется текущий рейтинг и задается отложенная функция удаления ImageView с экрана, срабатывающая через 1с. Полный код класса таракана представлен ниже.

class Cockroach(private val context: Activity) {  
 private val view = ImageView(context)  
  
 private val minLeftMargin = DeviceAdapter.getPxfromDp(10f, context)  
 private val maxLeftMargin = DeviceAdapter.getPxfromDp(320f, context)  
 private val minStartTopMargin = DeviceAdapter.getPxfromDp(-50f, context)  
 private val maxStartTopMargin = DeviceAdapter.getPxfromDp(20f, context)  
 private val maxTopMargin = DeviceAdapter.getPxfromDp(750f, context)  
 private val cockroachWidth = 220  
 private val cockroachHeight = 240  
  
 private var alive = true  
 private var hitSound = Audio("hit.ogg", context)  
  
 init {  
 view.setImageResource(R.drawable.*cockroach*)  
  
 view.setOnClickListener **{** hitSound.play()  
 view.setImageResource(R.drawable.*dead\_cockroach*)  
 if (alive) context.rate.*text* = (context.rate.*text*.toString().*toInt*() + 1).toString()  
 alive = false  
 Handler().postDelayed(**{** remove() **}**, 1000)  
 **}** setClickable(false)  
  
 val params = FrameLayout.LayoutParams(cockroachWidth, cockroachHeight)  
 params.setMargins(  
 (minLeftMargin..maxLeftMargin).*random*(),  
 (minStartTopMargin..maxStartTopMargin).*random*(),  
 0,  
 0  
 )  
 context.rootLayout.addView(view, 1, params)  
 }  
  
 fun remove() {  
 context.rootLayout.removeView(view)  
 }  
  
 fun isAlive() = alive  
  
 fun setClickable(clickable: Boolean) {  
 view.*isClickable* = clickable  
 }  
  
 fun move(step: Int): Boolean {  
 val param = view.*layoutParams* as ViewGroup.MarginLayoutParams  
 if (param.topMargin > maxTopMargin) return false  
 val moveToLeft = if (param.leftMargin - step >= minLeftMargin) -step else 0  
 val moveToRight = if (param.leftMargin + step <= maxLeftMargin) step else 0  
 param.setMargins(  
 param.leftMargin + (moveToLeft..moveToRight).*random*(),  
 param.topMargin + (0..step).*random*(),  
 param.rightMargin,  
 param.bottomMargin  
 )  
 view.*layoutParams* = param  
 return true  
 }  
}

# Тестирование

В ходе тестирования приложения осуществлялась проверка на соответсвие заявленных требований фактическому результату. Проверки были успешно пройдены. Приложение содержит весь необходимый функционал согласно заданию и корректно отрабатывает во всех прецедентах использования. На Рисунках 11-13 представлены скриншоты приложения KCaKC в различных состояниях.

Рис. 11 Начальное окно приложения KCaKC

Рис. 12 Запущенная игра в приложении KCaKC

Рис. 13 Уведомление об окончании игры в приложении KCaKC

Также в ходе тестирования было установлено, что значение ИН стабилизируется, примерно, через 2 минуты считывания пульса.

# Заключение

В ходе выполнения данной курсовой работы была создана мобильная игра KCaKC, которая способна отслеживать функциональное состояние пользователя в реальном времени, используя данные с камеры, и корректировать его путем динамического изменения поведения игры.

# Библиографический список

1. Android Developers [Электронный ресурс]. – Режим доступа: https://developer.android.com
2. Engy [Электронный ресурс]. – Режим доступа: https://engy.app
3. Kotlin Programming Language [Электронный ресурс]. – Режим доступа: https://kotlinlang.org
4. Vasić, M. Mastering Android Development with Kotlin. – Birmingham: Packt Publishing Ltd., 2017. – 378 с.
5. Welltory [Электронный ресурс]. – Режим доступа: https://welltory.com/ru/
6. Баевский, Р. М. Анализ вариабельности сердечного ритма при использовании различных электрокардиографических систем. Методические рекомендации / Р. М. Баевский, Г. Г. Иванов, Л. В. Чирейкин и др. // Вест. аритмол. – 2001. – № 24. – 87 с.
7. Кулаичев, А. П. Компьютерная электрофизиология и функциональная диагностика. Изд. 4-е, перераб. и доп. – М.: ИНФРА-М, 2007. – 640 с.
8. Проектирование информационных систем : практикум для реализации содержания образовательных программ высшего образования I ступени / авт.-сост. : А. Н. Семенюта, Л. В. Ятченко. – Гомель : учреждения образования «Белорусский торгово-экономический университет потребительской кооперации», 2015. – 80 с.

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ**

**УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**ЗАДАНИЕ НА КУРСОВОЙ ПРОЕКТ (РАБОТУ)**

**Студент** Кузьмичева К. Е.

(Фамилия, И., О.)

**Факультет** Программной инженерии и компьютерной техники

**Группа** P3322

**Руководитель** Штенников Д.Г., ИТМО, факультет ПИиКТ, доцент

(Фамилия, И., О., место работы, должность)

**Дисциплина** Разработка мобильных приложений

**Наименование темы** Разработать мобильное приложение, осуществляющее сбор данных о пульсе пользователя и его вариабельности, включающее игру, скорость которой зависит от индекса напряженности

**Задание** Осуществить разработку мобильного приложения для платформы Android, осуществляющего сбор данных о пульсе пользователя и его вариабельности, включающего игру, скорость которой зависит от индекса напряженности. Осуществить обработку требований к системе. Разработать набор диаграмм: одна из функциональных (IDEF0, IDEF3, DFD), объектные UML (Use-Case, Collaboration, Classes, Sequences), описание структур данных. Привести макет интерфейса, подобрать или разработать алгоритмы для обработки данных в системе, подобрать библиотеки для реализации (или написать самостоятельно) выработать рекомендации по написанию программного кода для реализации системы. В рамках данного задания требуется получать данные о пульсе пользователя, используя вспышку и камеру мобильного телефона в качестве сенсора, и на основе этих данных определять скорость течения игрового процесса.

Студент Дата «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020г.

Подпись Дата

Руководитель Дата «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г. Подпись Дата

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ**

**УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**ОТЗЫВ РУКОВОДИТЕЛЯ О ВЫПОЛНЕНИИ**

**КУРСОВОЙ РАБОТЫ**

**Студент** Кузьмичева К. Е.

(Фамилия, И., О.)

**Факультет** Программной инженерии и компьютерной техники

**Группа** P3322

**Руководитель** Штенников Д.Г., ИТМО, факультет ПИиКТ, доцент

(Фамилия, И., О., место работы, должность)

**Дисциплина** Разработка мобильных приложений

**Наименование темы** Разработать мобильное приложение, осуществляющее сбор данных о пульсе пользователя и его вариабельности, включающее игру, скорость которой зависит от индекса напряженности

**ОЦЕНКА КУРСОВОГО ПРОЕКТА (РАБОТЫ)**

| **№**  **п/п** | **Показатели** | **Оценка** | | | |
| --- | --- | --- | --- | --- | --- |
| **5** | **4** | **3** | **0** |
| 1. | Соответствие отчетов требованиям задания |  |  |  |  |
| 2. | Ответы на вопросы при защите курсовой работы |  |  |  |  |
| 3. | Сдача этапов курсовой работы в срок |  |  |  |  |
| 4. | Полнота сдачи этапов курсовой работы |  |  |  |  |
| 5. | Соответствие оформлению по ГОСТ |  |  |  |  |
| 6. | Качество оформления пояснительной записки (общий уровень грамотности, стиль изложения, качество иллюстраций, корректность цитирования и пр.) |  |  |  |  |
| **Итоговая оценка** |  |  |  |  |  |

**Отмеченные достоинства:** Кузьмичева К. Е. проявила весьма высокую заинтересованность при изучении теоретических положений физиологических основ и основных методов проведения анализа вариабельности сердечного ритма, используемых в работе. Четко изложена концепция работы полученного мобильного приложения. Приложение полностью соответствует заявленным в задании требованиям. Обоснована практическая значимость и полезность приложения.

**Отмеченные недостатки:** В приложении наблюдаются выводы недостоверной информации о пульсе и индексе напряженности при малой (менее 20 сек) продолжительности замера ЧСС. Недостаточно освещен вопрос обработки данных, получаемых с камеры, и фиксирования по ним ударов сердца.

**Заключение:** В целом, курсовая работа Кузьмичевой К. Е. решает поставленные цели и задачи, отвечает предъявленным требованиям, может быть рекомендована к защите и заслуживает оценки «отлично».

Студент Дата «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г.

Подпись Дата

Руководитель Дата «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г.

Подпись Дата

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ**

**УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**АННОТАЦИЯ НА КУРСОВОЙ ПРОЕКТ (РАБОТУ)**

**Студент** Кузьмичева К. Е.

(Фамилия, И., О.)

**Факультет** Программной инженерии и компьютерной техники

**Группа** P3322

**Руководитель** Штенников Д.Г., ИТМО, факультет ПИиКТ, доцент

(Фамилия, И., О., место работы, должность)

**Дисциплина** Разработка мобильных приложений

**Наименование темы** Разработать мобильное приложение, осуществляющее сбор данных о пульсе пользователя и его вариабельности, включающее игру, скорость которой зависит от индекса напряженности

**ХАРАКТЕРИСТИКА КУРСОВОГО ПРОЕКТА (РАБОТЫ)**

**1. Цели и задачи работы**

Сформулированы при участии студента

Предложены студентом

Определены руководителем

Задачей данного курсового проекта является разработка мобильного приложения-игры. Приложение должно осуществлять сбор данных о пульсе пользователя, анализировать эти данные и использовать в игре.

**2. Характер работы** Расчет Конструирование

Моделирование Другое, разработка

Работа носит практический характер.

**3. Содержание работы** Работа содержит следующие части: Введение; Обзорная часть с анализом и доопределеним задачи, описанием технологий реализации и приведением аналогов; Проектная часть с приведением функциональных и объектных диаграмм; часть Реализации со скриншотами, фрагментами кода и результатами тестирования; Заключение и Библиографический список.

**4. Выводы**

По результатам выполнения курсовой работы удалось разработать мобильное приложение, включающее в себя игру и систему сбора информации о пульсе пользователя с расчетом необходимых параметров вариабельности сердечного ритма. В процессе игры в зависимости от рассчитанного индекса напряженности меняется скорость генерации требований действий пользователя.

Студент Дата «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г.

Подпись Дата

Руководитель Дата «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г

Подпись Дата

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ**

**УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**ГРАФИК ВЫПОЛНЕНИЯ КУРСОЙ РАБОТЫ**

**Студент** Кузьмичева К. Е.

(Фамилия, И., О.)

**Факультет** Программной инженерии и компьютерной техники

**Группа** P3322

**Руководитель** Штенников Д.Г., ИТМО, факультет ПИиКТ, доцент

(Фамилия, И., О., место работы, должность)

**Дисциплина** Разработка мобильных приложений

**Наименование темы** Разработать мобильное приложение, осуществляющее сбор данных о пульсе пользователя и его вариабельности, включающее игру, скорость которой зависит от индекса напряженности

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Наименование этапа** | **Дата завершения** | | **Оценка и подпись руководителя** |
| **Планируемая** | **Фактическая** |
| 1 | Анализ требований к программной системе |  |  |  |
| 2 | Проектирование мобильного приложения на основе функционального подхода |  |  |  |
| 3 | Проектирование мобильного приложения на основе объектного подхода |  |  |  |
| 4 | Конструирование мобильного приложения. Создание интерфейса |  |  |  |
| 5 | Конструирование мобильного приложения. Формирование структуры данных |  |  |  |
| 6 | Конструирование мобильного приложения. Обработка данных с датчиков |  |  |  |
| 7 | Тестирование программных систем |  |  |  |
| 8 | Оформление пояснительной записки |  |  |  |

Студент Дата «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г.

Подпись Дата

Руководитель Дата «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г.

Подпись Дата