

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

**Факультет компьютерных наук
Департамент программной инженерии**

Согласовано

Академический руководитель ОП
Системная и программная инженерия
профессор ДПИ

_____ Д.В. Александров

" ____ " _____ 2018 г

Утверждаю

Академический руководитель
образовательной программы
«Программная Инженерия»
профессор, канд. техн. наук
В. В. Шилов

" ____ " _____ 2018 г

**Клиент-Серверное Android-Приложение для Управления Скидками в
Розничных Сетях**

Пояснительная записка

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.506900 81 01-1

Студент группы БПИ 151 НИУ-ВШЭ

_____ Куприянов К. И.

" ____ " _____ 2018 г

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл.	

2018

УТВЕРЖДЕНО
RU.17701729.506900 81 01-1

Клиент-Серверное Android-Приложение для Управления Скидками в Розничных Сетях

Пояснительная записка

RU.17701729.506900 81 01-1

Листов 18

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

2018

1. Аннотация

В данном программном документе приведена пояснительная записка к программе «Клиент-Серверное Android-Приложение для Управления Скидками в Розничных Сетях». В данном программном документе, в разделе «Введение» указано наименование программы, краткое наименование программы и документы, на основании которых ведется разработка. В разделе «Назначение и область применения» указано функциональное назначение программы и краткая характеристика области применения программы. В данном программном документе, в разделе «Технические характеристики» содержатся следующие подразделы:

- Постановка задачи на разработку программы;
- Описание алгоритма и функционирования программы с обоснованием выбора схемы алгоритма решения задачи и возможные взаимодействия программы с другими программами;
- Описание и обоснование выбора состава технических и программных средств

Так же к документу прикреплены приложения. Настоящий документ разработан в соответствии с требованиями:

- 1) ГОСТ 19.101-77 Виды программ и программных документов [**gost_types_of_software**];
- 2) ГОСТ 19.102-77 Стадии разработки [**gost_stages_of_devel**];
- 3) ГОСТ 19.103-77 Обозначения программ и программных документов [**gost_marking_software**];
- 4) ГОСТ 19.104-78 Основные надписи [**gost_main_signs**];
- 5) ГОСТ 19.105-78 Требования к программным документам [**gost_demands_for_docs**];
- 6) ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению [**gost_pz**].

Изменения к данной Пояснительной Записке оформляются согласно ГОСТ 19.603-78 [**gost_main_rules_change**]. Перед прочтением данного документа рекомендуется ознакомиться с терминологией, приведенной в Приложении 1 настоящей Пояснительной Записки.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.506900 81 01-1				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

Содержание

1	Аннотация	1
2	Введение	3
2.1	Наименование программы	3
2.2	Краткая характеристика	3
3	Назначение разработки	4
3.1	Функциональное назначение	4
3.2	Эксплуатационное назначение	4
4	Технические характеристики	5
4.1	Постановка задачи на разработку программы	5
4.2	Описание алгоритма и функционирования программы	6
4.2.1	Описание построения Андроид приложения	6
4.2.2	Описание построения crawler'a	8
4.3	Ежедневное обновление акций	9
5	Технико-экономические показатели	11
5.1	Предполагаемая потребность	11
5.2	Экономические преимущества разработки	11
6	Источники, используемые при разработке	12
7	Приложение 1. Терминология	13
7.1	Терминология	13
8	Приложение 2. Описание классов	14
8.1	WalkByLook	14
8.2	BellBehaviour	14
8.3	BellPuzzleLauncher	14
8.4	KeyDoorOpener	15
8.5	LastPuzzleLogic	15
8.6	MainMenu	15
8.7	ModChanger	15

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.506900 81 01-1				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

8.8 NumberChanger	16
8.9 OpenDoorAndLoadScene	16
8.10 PickUpObject	16
8.11 StoryLogic	17
8.12 Unscrew	17
8.13 VRSlider	17

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.506900 81 01-1				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

2. Введение

2.1. Наименование программы

Наименование программы на русском: “Клиент-Серверное Android-Приложение для Управления Скидками в Розничных Сетях”.

Наименование на английском: “The Client-Server Android Application for Managing the Products’ Discounts in Retail Networks”.

2.2. Краткая характеристика

Программа предназначена для пользователей смартфонов на базе платформы Android. Цель работы - создать удобное приложения для составления списков покупок, добавления в них любых товаров (даже тех, которых нет в магазине), отслеживания акций, а так же просмотра всех акций в нескольких магазинах. Это позволит пользователям экономить свои средства на ежедневных покупках и быть в курсе актуальных акций магазинов. Серверная часть приложения должна представлять из себя web-crawler для сбора данных с сайтов розничных сетей. Должен быть реализован механизм взаимодействия с клиентами посредством обменивания файлами в формате JSON. Главной чертой данного приложения является его лёгкая, быстрая масштабируемость и модульность программного кода.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.506900 81 01-1				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

3. Назначение разработки

3.1. Функциональное назначение

К функциональным возможностям программы относятся: просмотр списка актуальных акций, составление и редактирование списков покупок, регистрация и вход в аккаунт пользователей, возможность совместного использования списков покупок между пользователями. К функциональным возможностям серверной части относится сбор актуальных акций магазинов посредством crawling'a (см. терминологию) сайтов торговых сетей и отправки запросов на сервер приложения.

3.2. Эксплуатационное назначение

Программа предназначена для запуска на мобильных устройствах операционной системы Андроид. Продукт является приложением “на стыке онлайн и оффлайна”, позволяющим планировать покупки в магазине с учетом актуальных скидок. Программа даёт пользователю возможность иметь несколько списков одновременно и добавлять в них любые товары; искать и подбирать интересующие его товары по сниженной цене, и, таким образом, экономить средства.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.506900 81 01-1				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

4. Технические характеристики

4.1. Постановка задачи на разработку программы

Программа должна соответствовать требованиям, представленным в Техническом Задании.

Задачи работы (Андроид-клиент):

1. Реализовать возможность просмотра списка доступных магазинов с акционными товарами
2. Реализовать представление текущих акций для конкретного магазина:
 - (а) В виде общего списка
 - (б) По категориям
3. Реализовать постепенную загрузку товаров магазинов (по страницам) для экономии трафика и меньшей нагрузкой на мобильное устройство
4. Реализовать регистрацию через мобильное приложение
5. Реализовать вход в аккаунт через мобильное приложение
6. Реализовать возможность смены аккаунта
7. Реализовать возможность создания списков покупок с разными названиями
8. Реализовать возможность удаления списка покупок
9. Реализовать возможность добавления товара в список покупок
10. Реализовать возможность удаления товара из списка покупок
11. Реализовать возможность добавления в список покупок пользовательских товаров, которых нет в магазине (см. терминологию)
12. Реализовать возможность просмотра подобранных программой товаров согласно запросу пользователя
13. Реализовать добавление подобранных товаров в список покупок
14. Реализовать предварительное отображение элементов каждого списка покупок до их открытия
15. Реализовать отображение всплывающих подсказок при долгом нажатии на элементы управления button (кнопка)
16. Реализовать перенаправление в настройки сети для последующего включения интернета при отсутствии интернет-соединения
17. Реализовать отображение индикатора процесса загрузки данных с сервера
18. Реализовать обучающий фрагмент в разделе help (см. терминологию), содержащий руководство пользователя по управлению программой

Задачи работы (серверная часть):

1. Реализовать crawling веб-страниц для сбора актуальной информации об акционных товарах
2. Реализовать добавление товаров в базу данных посредством отправки запросов REST API (REST API и база данных реализованы напарником)

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.506900 81 01-1				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

3. Реализовать запись акционных товаров со всех магазинах в JSON файл
4. Реализовать email уведомления администраторам сервиса об ошибках и неполадках в работе сервера

4.2. Описание алгоритма и функционирования программы

4.2.1. Описание построения Андроид приложения

В андроид приложении задумано 4 активности (см. терминологию) и 2 фрагмента (см. терминологию)

1. **LoginActivity** Рис 1. Активность, формирующая интерфейс входа пользователя в систему. Является самой первой активностью, которая загружается всегда при запуске приложения, но если пользователь до этого момента уже входил в свой аккаунт, его данные сохранены в shared preferences (см. терминологию) и эта активность опускается. В этой активности 2 поля для ввода текста (логина и пароля) и 3 кнопки (войти, зарегистрироваться и продолжить использование без входа в аккаунт);
2. **RegisterActivity** Рис 2. Активность для регистрации пользователя. Имеет 2 поля для ввода текста (логина и пароля), и 2 кнопки: “Зарегистрироваться” (для новых пользователей) и “Войти” (для пользователей, уже имеющих аккаунт). Пользователь также может вернуться в активность для входа в систему посредством нажатия системной кнопки “назад”;

Рис. 1: Активность для входа в аккаунт пользователя

Рис. 2: Активность для регистрации аккаунта пользователя

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.506900 81 01-1				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

3. **MainActivity** Главная активность, содержащая 2 дочерних фрагмента, toolbar и BottomNavigationView (см. терминологию).

Фрагменты:

- (a) **HomeFragment** Рис 3. Фрагмент, содержащий список всех товаров. Список может быть отфильтрован по магазинам и по категориям. Содержит toolbar, в котором есть выпадающий список для выбора магазина, кнопка для выхода/входа из/в аккаунт(а), и меню опций;
- (b) **ShopListsPreviewFragment** Рис 4. Фрагмент, содержащий список всех списков покупок. Списки покупок упорядочены в виде карточек с предпросмотром в них добавленных товаров магазина и пользовательских товаров. Нажатие по одному из них открывает активность ShopListActivity, содержащую подробную информацию об элементах списка покупок. Долгое нажатие по одному из списков вызовет диалоговое окно, в котором спросится подтверждение намерения удалить данный список покупок. В элементе toolbar в этом фрагменте есть кнопка для входа/выхода в/из аккаунт(а), и кнопка для добавления нового списка покупок. Использование фрагмента не доступно неавторизованным пользователям;

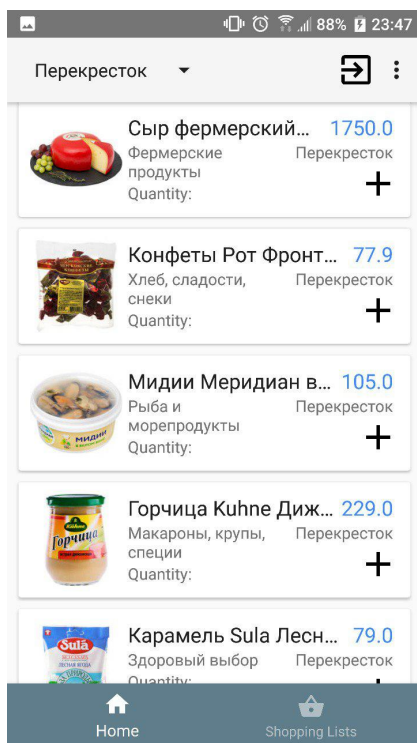


Рис. 3: MainActivity. Фрагмент со списком товаров магазинов

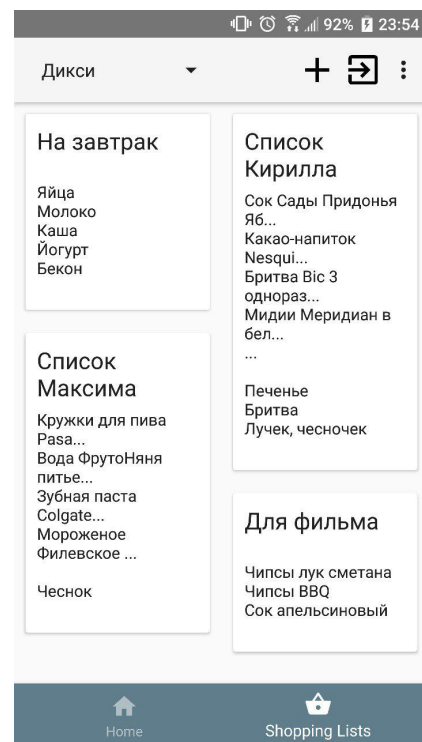


Рис. 4: MainActivity. Фрагмент со всеми списками покупок

4. **ShopListActivity** Рис 5, рис 6. Активность, содержащая подробную информацию о конкретном списке покупок. В элементе toolbar отображается кнопка "назад", название списка покупок, и итоговая сумма по всем товарам, в стандартной для товаров валюте. В центре активности расположен список товаров магазина, а под ним - список пользовательских товаров. В свою очередь, каждый пользовательский товар содержит список подобранных товаров под данный из магазинов. Использование активности не доступно неавторизованным пользователям

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.506900 81 01-1				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

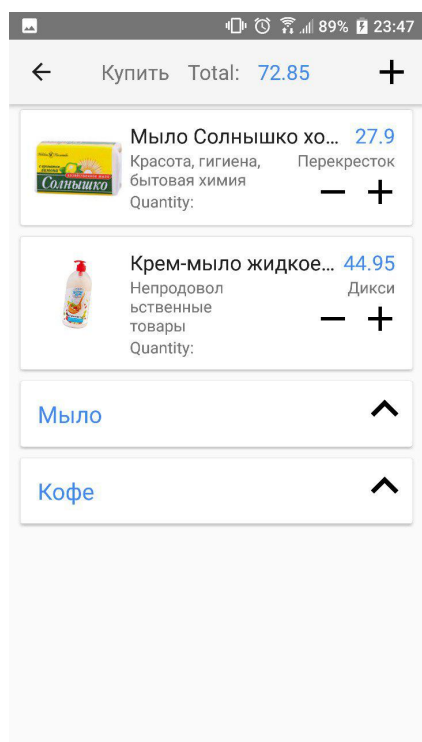


Рис. 5: ShopListActivity

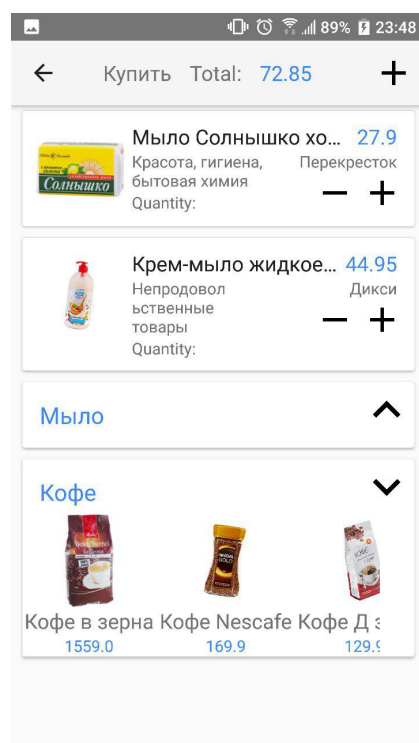


Рис. 6: ShopListActivity. Показаны соответствующие пользовательскому товару товары их магазинов

4.2.2. Описание построения crawler'a

В crawler'e веб-страниц магазинов предусмотрены следующие модули: spiders/dixy_spider, spiders/perekrestok_spider, text_processor, dixy_selectors, perekrestok_selectors, settings, config, notify и pipelines.

1. **dixy_spider** описывает класс DixySpider, унаследованный от scrapy.Spider. Отвечает за получение исходного кода страницы магазина "Дикси", его парсинга при помощи "модульных" css селекторов (см. терминологию), перехода по страницам на сайте. Формирует объекты класса DixyItem и передаёт далее модулю pipelines на обработку;
2. **perekrestok_spider** описывает класс PerekrestokSpider, функционал идентичен DixySpider, но действует на сайте магазина "Перекресток";
3. **text_processor** содержит функции обработки текста при помощи регулярных выражений и базовых операций со строками;
4. **dixy/perekrestok selectors** содержат соответствующие xpath селекторы для веб-страниц магазинов. Выделение этой логики в отдельный модуль - корневая причина модульности приложения. В случае изменения дизайна сайта, кроулинг которого происходит, администратору придётся изменить лишь одну строчку для одного элемента на странице;
5. **settings, config** содержат настройки и конфигурацию кроулера;
6. **pipelines** содержит методы для обработки объектов DixyItem и PerekrestokItem. Мо-

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.506900 81 01-1				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

дальше так же совершает POST запрос на сервер приложения для добавления товара в базу данных, и сохраняет все объекты в JSON файл

4.3. Ежедневное обновление акций

Процесс ежедневного обновления акций необходимо проводить, чтобы акции в приложении всегда были актуальными.

На сервере запущена cron job (см. терминологию), которая каждый день в 2:00 ночи в 6:00 утра запускает процесс кроулинга акций во всех магазинах:

```
00 2,6 * * * cd /home/hes/Projects/crawler/easysales && ./crawl_all.sh > /home/hes/crawler.log 2>&1
```

Скрипт `crawl_all.sh` отвечает за поочередный запуск всех spider'ов. Далее, по цепочке, спайдеры собирают данные с сайтов при помощи xpath селекторов, используют модуль `text_processor` для очистки собранных данных от non-breaking space символов, длинных пробелов и других символов. Формируются `<Shop>Items`, которые затем отправляются в модуль `pipelines`. Классы `<Shop>Items`, процесс добавления Item'а в базу и модель с селекторами для магазина "Перекресток" описаны ниже.

Классы `DixyItem` и `PerekrestokItem` унаследованы от `EasySalesItem`, базового товара приложения. Его поля всегда заполнены, то есть не могут иметь значения `null`.

```
class EasySalesItem(scrapy.Item):
    name = scrapy.Field()
    newPrice = scrapy.Field()
    imageUrl = scrapy.Field()
    shopId = scrapy.Field()
    crawlDate = scrapy.Field()
```

```
class DixyItem(EasySalesItem):
    category = scrapy.Field()
    oldPrice = scrapy.Field()
    discount = scrapy.Field()
    dateIn = scrapy.Field()
    dateOut = scrapy.Field()
    condition = scrapy.Field()
```

```
class PerekrestokItem(EasySalesItem):
    category = scrapy.Field()
    oldPrice = scrapy.Field()
```

Отправка объекта класса `Item` в базу данных происходит при помощи API запроса на сервер:

```
class DataBaseWriterPipeLine(object):

    def process_item(self, item, spider):
        requests.post(const['ADD_ITEM_API'],
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.506900 81 01-1				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

```

data=json.dumps(dict(item)),
headers=const['REQUEST_HEADERS'])
return item

```

Xpath селекторы для магазина “Перекресток”. Модульность и простота использования написанного фрагмента кода достигается тем, что при изменении дизайна сайта магазина, возникает необходимость исправить всего несколько строчек в приведённом фрагменте. Таким образом, частично решается проблема всех кроулеров - поломка, в случае изменения дизайна сайта. А поскольку большинство кроулеров не используют такой модульный подход, времени и сил на починку уходит гораздо больше, так как приходится править множество файлов во многих местах.

```

# Start urls.
#
URLS = ['https://www.perekrestok.ru/catalog']
URL_CORE = 'https://perekrestok.ru'

# Categories.
#
CATEGORIES = '//a[@class="xf-catalog-categories__link"]/@href'
CATEGORIES_TEXT = '//span[@class="xf-catalog-categories__text"]/text()'
POST_CATEGORY = '//span[@class="xf-breadcrumbs__current"]/text()'
CURRENT_CATEGORY = '//h1[@class="xf-caption__title"]/text()'

ROOT_NODE = '//ul[@id="catalogItems"]'
ITEM = 'li/div[contains(@class, "xf-product")]'

# Item attributes.
#
NAME = 'div[@class="xf-product__title xf-product-title"]/a[@class="xf-product-\
title__link js-product__title"]/text()'
IMG = 'figure/a/img/@data-src'
NEW_PRICE = 'div/div[contains(@class, "xf-product-cost__current")]/@data-cost'
OLD_PRICE = 'div/div[contains(@class, "xf-price xf-product-cost__prev")]/@data-cost'

# Pagination selector.
#
# next_page = '//li[@class="next"]/a/@href'
NEXT_PAGE = '//a[contains(@class, "xf-paginator__item js-paginator__next")]/@href'

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.506900 81 01-1				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

5. Техничко-экономические показатели

5.1. Предполагаемая потребность

«Игра — Эскейп Квест с Использованием Очков Виртуальной Реальности» может быть использована в игровой сфере. Программу могут использовать люди, нуждающиеся как в отдыхе в мире виртуальной реальности, так и желающие интеллектуально развиваться в игровой форме.

5.2. Экономические преимущества разработки

На момент принятия решения о написании данного продукта во Всемирной сети Интернет был лишь 1 аналог: игра «Lost in the Kismet». Преимущества данной разработки перед конкурентом:

1. Совместимость со всеми девайсами Cardboard
2. VR Mode и Normal Mode
3. Оригинальный сюжет

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.506900 81 01-1				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

6. Источники, используемые при разработке

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.506900 81 01-1				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

7. Приложение 1. Терминология

7.1. Терминология

VR - Virtual Reality (Виртуальная Реальность). Созданный программными и техническими средствами мир, передаваемый человеку при помощи взаимодействия специальных внешних устройств с его органами чувств.

Google Cardboard - Эксперимент компании Google в области виртуальной реальности, в основе которого лежит шлем, который, по замыслу разработчиков, можно собрать из подручных материалов. Состоит из картона, оптических линз, липучек-застёжек. Так же необходимо наличие смартфона с поддержкой технологии VR и установленным VR приложением. Он закрепляется непосредственно в шлеме, а шлем крепится к голове пользователя, что передает программе движения головы.

VR Mode - Режим отображения картинки на экране мобильного устройства, при котором экран разделен на 2 части, на которые выводятся изображения для левого и правого глаза. Система линз в Google Cardboard (Cardboard) корректирует геометрию изображения, доставляя пользователю полное ощущение присутствия в виртуальном 3D мире.

Normal Mode - Режим отображения картинки на экране мобильного устройства, при котором экран не разделяется на 2 части. Не требует наличия Google Cardboard.

Reticle - Указатель (прицел) в центре экрана, меняющий свою позицию синхронно с главной камерой. Позволяет более точно прицеливаться для взаимодействия с объектами.

Draw Calls - Вызовы отрисовки. То, сколько объектов отрисовываются на экране за один кадр (frame).

Escape the room - (рус. Выйти из комнаты; покинуть комнату) — жанр компьютерных игр, поджанр квестов, основная цель которого - найти выход из запертого помещения, используя любые подручные средства.

Rigidbody - Rigidbody дает игровым объектам физические свойства, такие как масса, влияние гравитации и constraints (ограничения, задержки) по всем осям для задержания объекта на месте.

Collider - Компонент коллайдер определяет фигуру объекта с целью установки физических столкновений.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.506900 81 01-1				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

8. Приложение 2. Описание классов

При программировании на Unity, каждый скрипт представляет собой отдельный класс, в котором, как правило, есть 2 метода: Start и Update. Start вызывается 1 раз при запуске скрипта, а Update вызывается 1 раз за фрейм.

8.1. WalkByLook

Поля:

```
public float angle1 = 30F, angle2 = 80F;  
public float speed = 5F;  
private bool moveForward;  
private CharacterController cc;  
public Transform vrCam;  
private bool lookingToObject = false;  
private bool mustLookToTheObject = false;  
public AudioSource[] footSteps;  
private bool shouldMove = false, playing = false;
```

Класс, созданный для передвижения игрока при помощи угла наклона головы по оси X.

8.2. BellBehaviour

Поля:

```
public Material bellHoverMaterial;  
public Material defaultBellMaterial;  
private AudioSource audioSrc;
```

Отвечает за поведение колокольчика при наведении на него reticle и нажатии (загрузки нижнего индикатора).

8.3. BellPuzzleLauncher

Поля:

```
public Transform handMountingPosition;  
private bool hasBell = false;  
private float angle;  
public Transform vrCam;  
public GameObject bellPuzzle;  
public Transform firstToComplete, secondToComplete;  
private bool firstPut = false, secondPut = false;
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.506900 81 01-1				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

Прикрепляется к невидимому кубу с коллайдером, который при столкновении с персонажем и наведении им указателя на место головоломки вешает колокольчик из руки, и, если все 5 на месте, запускает головоломку.

8.4. KeyDoorOpener

Поля:

```
public GameObject door, final;  
private static bool opened = false;  
private float timeToOpen = 1F;  
public Transform handMountingPosition;
```

Класс, отвечающий за открытие двери когда в руке у игрока находится ключ.

8.5. LastPuzzleLogic

Поля:

```
char[] numbers;  
bool solved;  
float step;  
string key = "4118956";  
public GameObject Key;
```

Класс, отвечающий за поведение последней головоломки с ключем на книжном шкафу.

8.6. MainMenu

Поля:

```
public GameObject character, story;
```

Класс главного меню, прикрепляется к двум лэйблам «PLAY» и «EXIT».

8.7. ModChanger

Поля:

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.506900 81 01-1				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

```
public GameObject modeChooser;  
public static bool vrModeEnabled = false;  
private bool checked_ = false;
```

Класс, отвечающий за выбор режима (VR Mode или Normal Mode)

8.8. NumberChanger

Поля:

```
int n;  
public float timeDelay = 1F;  
float currentTime = 0F;  
bool gazedAt = false;
```

Класс, отвечающий за изменение каждой цифры в последней головоломке.

8.9. OpenDoorAndLoadScene

Поля:

```
public GameObject door;  
public static bool opened = false;  
public float timeToOpen = 0.6F;
```

Класс, отвечающий за открытие двери с TransitScene для входа в дом.

8.10. PickUpObject

Поля:

```
public Vector3 handPosition;  
public Vector3 handRotation;  
Vector3 oldScale;  
public float angle1 = 275F, angle2 = 303F;  
public Transform handMountingPosition;  
public Transform vrCam;  
private bool tilted;  
bool pickedUp = false;
```

Класс, отвечающий за перемещение объекта с места, где он лежал, в руку игроку.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.506900 81 01-1				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

8.11. StoryLogic

Поля:

```
public GameObject character, vrCam, guide, tryIt, glhf, stone, constraint;  
private bool freezeMove = true, thrown = false, shown = false;
```

Класс, отвечающий за последовательность показа приветственных текстов и обучающего фрагмента вначале.

8.12. Unscrew

Поля:

```
bool unscrewed = false, hasScrewDriver = false;  
public Transform handMountingPosition;
```

Класс, отвечающий за откручивание колокольчика со стены при помощи отвертки в руке игрока.

8.13. VRSlider

Поля:

```
public float fillTime = 2f;  
public int Scene = 1;  
private Slider mySlider;  
private float timer;  
private bool gazedAt;  
private Coroutine fillBarRoutine;
```

Класс, отвечающий за анимацию слайдера - индикатора внизу экрана.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.506900 81 01-1				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

Лист регистрации изменений

Изм.	Номера листов (страниц)				Всего листов (страниц) в докум.	№ докум.	Входящий № сопроводительного докум. и дата	Подпись	Дата
	измененных	замененных	новых	аннулированных					

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.506900 81 01-1				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата