

## RPG-игра

### 1. Условия.

Задача кейса – написать простую фэнтези RPG-инди-игру от третьего лица под виртуальную реальность. Цель игрока — поднять социальный ранг и уровень персонажа, заработать золотые монеты и выжить в подземельях.

### 2. Техническое задание.

Должна быть реализована игра, где Вашей задачей является выбор действий главного героя, влияющих на развитие сюжета. В процессе игры игроку рассказывают интерактивную историю. Игрок имеет выбор действий, ограниченный набором из не менее трех действий в каждой ситуации сюжета. Действия игрока должны быть разнообразны в каждой точке сюжета: от выбора необходимости и способа атаки до перечня покупок у неигровых персонажей (чем больше вариативность, тем лучше). История заканчивается, когда игрок попадает в предопределённый «тупик» своей ветки истории, то есть у текущей ситуации нет вариантов развития событий (достигнут предписанный конец игры), либо, когда одна из характеристик главного героя достигает критического значения (например, обнулится показатель здоровья). При выборе варианта развития сюжета у персонажа могут меняться его характеристики: уровень персонажа (L), количество золота (G), репутация (R), количество очков здоровья (H). Наилучшим решением будет являться игра с большим количеством направлений развития истории (более 10) и различными концовками (более 5), демонстрируемых в конце игры в виде графа.

Успешной реализацией кейса будет считаться игра с графическим интерфейсом, трёхмерными графикой и моделями персонажей и работой с очками виртуальной реальности. Также должны присутствовать записи диалогов, лежащие отдельными файлами, а не их текстовая визуализация. Алгоритм логики игры должен быть реализован на парадигме ООП.

**Московская предпрофессиональная олимпиада школьников**  
**Заключительный этап**  
**Кейсовые задачи.**

**Инженерно-конструкторское направление. Профиль Информационные технологии**

---

**3. Регламент испытаний:**

- Участники демонстрируют жюри функционал выбора игровой последовательности повествования и действия в игре.
- Участники демонстрируют один из заинтересовавших жюри вариантов прохождения игры после прохождения игры.
- Участники демонстрируют жюри граф прохождения игры со всеми уровнями, выборами, изменениями характеристик и победным и проигрышным выборами.

**4. Примерный перечень средств и инструментов для выполнения задания.**

- <https://developer.android.com/studio>
- <https://unity.com/ru>
- <https://developer.apple.com/xcode/>
- <https://www.babylonjs.com/>
- <https://aframe.io/>
- <https://java.com/ru/download/>
- <https://docs.microsoft.com/ru-ru/dotnet/csharp/>
- <https://www.apple.com/ru/swift/>
- <https://www.typescriptlang.org/>

**«Электронный журнал»**

**Условие:**

Задача кейса – разработать прототип электронного журнал для школы.

**Техническое задание:**

В кейсе рассматривается среднестатистическая общеобразовательная школа, в которой происходит обучение детей по стандартной программе среднего общего образования (11 классов). Необходимо учесть, что параллель одного года обучения может в себе содержать не один класс. В школе учатся ученики разных классов, а также работают учителя, которые преподают по несколько дисциплин, причем некоторые имеют дополнительную нагрузку в виде классного руководства либо факультативных предметов, кружков. Помимо преподавателей, в школе есть прочий персонал: директор, завуч, охранники, повара, уборщики. У каждого ученика, учителя и сотрудника школы есть пропуск, который имеет собственный уникальный номер, однако школу посещают и люди, у которых его нет, например, родители пришедшие на родительское собрание.

Успешно выполненной реализацией кейса будет считаться продукт, включающий в себя следующий функционал:

- проведение внутреннего хранения информации обо всех учениках с их родителями, учителей и сотрудников школы;
- хранение в программе всех классов с их учениками и классными руководителями, а также всех группы кружков и факультативов;
- контроль успеваемости всех учеников на их уроках, кружках и факультативах;
- разделение ролей при аутентификации пользователя в программу через удаленный сервер;

**Московская предпрофессиональная олимпиада школьников**

**Заключительный этап**

**Кейсовые задачи.**

**Инженерно-конструкторское направление. Профиль Информационные технологии**

---

- возможность администрировать и отслеживать работу пользователей в системе на сервере;
- приложение должно быть кроссплатформенным.

Все данные вы имеете право создавать сами, однако хранить их нужно в любом удаленном СУБД. Нет необходимости совпадения ваших данных с реальными данными ваших учебных заведений.

**Регламент испытаний:**

- Участники демонстрируют жюри возможность получения всех полей данных по человеку из базы по его идентификационному номеру пропуска в своем приложении.

- Участники демонстрируют жюри возможность получения полного графика обучения отдельно взятого из системы ученика.

- Участники демонстрируют жюри возможность вывода как общей успеваемости класса по отдельно взятому предмету, так и полную успеваемость отдельно взятого ученика по всем программам обучения.

- Участники демонстрируют жюри возможность вывода полного списка класса с возможностью просмотра как всей информации ученика, так и его родителей.

- Участники демонстрируют возможность ведения журнала посещаемости и успеваемости класса по предмету или группы по кружку/факультативу только аутентифицированному преподавателю.

**Примерный перечень средств и инструментов для выполнения задания.**

- <https://developer.android.com/studio>
- <https://developer.apple.com/xcode/>
- <https://dotnet.microsoft.com/apps/xamarin>
- <https://ru.vuejs.org/index.html>
- <https://angular.io/>
- <https://dotnet.microsoft.com/apps/aspnet/web-apps/blazor>
- <https://metanit.com/sharp/wpf/>
- <https://metanit.com/sharp/aspnet5/>
- <https://nodejs.org/en/>

## Распознавание объектов

### Условия:

Задача кейса – разработать приложение с использованием моделей машинного обучения для распознавания объектов реального мира (например, стол, стул, кружка, тарелка, человек, дерево, телевизор, кошка и т. д.) с камеры и далее преобразовывать название класса распознанного объекта в аудиосообщение с последующим его воспроизведением.

### Техническое задание.

Должно быть реализовано приложение для стационарного компьютера или мобильного устройства. Программа получает изображение с камеры и с помощью библиотек Keras, PyTorch или других распознаёт изображение. В данных библиотеках существуют обученные модели для распознавания изображения. Далее с помощью библиотек RNVoice, eSpeak NG, онлайн-сервисов от Google, Yandex, или другими способами (например, предзаписанный текст) происходит озвучка класса распознанного объекта.

Успешной реализацией проекта будет считаться законченное приложение с минимальным интерфейсом. Проект имеет важную социальную значимость. Слабовидящие и незрячие люди смогут пользоваться этим приложением в повседневной жизни.

### Регламент испытаний:

- Участники демонстрируют жюри работающее приложение.
- Участники демонстрируют жюри процесс распознавания объектов, точность распознавания.
- Участники демонстрируют жюри возможности озвучивания названия классов.

**Московская предпрофессиональная олимпиада школьников**

**Заключительный этап**

**Кейсовые задачи.**

**Инженерно-конструкторское направление. Профиль Информационные технологии**

---

- Участники демонстрируют жюри интерфейс программы и дополнительные возможности.

**Примерный перечень средств и инструментов для выполнения задания.**

- <https://developer.android.com/studio>
- <https://pytorch.org/>
- <https://keras.io/>
- <https://github.com/Olga-Yakovleva/RHVoice>
- <https://github.com/espeak-ng/espeak-ng>
- <https://cloud.google.com/speech-to-text/>
- <https://cloud.yandex.ru/services/speechkit>

### AR-модель города

*9 – 11 классы:*

#### **Условия:**

Задача кейса – разработать приложение, использующее технологию дополненной реальности (Augmented reality) для создания, редактирования и интерактивного взаимодействия с моделью города (частью города).

#### **Техническое задание.**

Должно быть реализовано приложение для различного рода платформ (по выбору разработчика, хотя бы одна платформа): стационарные компьютеры, мобильные устройства, очки виртуальной и/или дополненной реальности. Используя это приложение и различного рода физические маркеры, пользователь может интерактивно создавать модель города или района. Маркер – это специально подготовленные объекты реального мира, легко распознаваемые простыми алгоритмами, карточки с простыми геометрическими изображениями и др. Должны быть представлены различные маркеры для: жилых зданий, общественных зданий, больниц, школ, торговых центров, спортивных сооружений, парков и т.п. 3D-модели зданий можно сгенерировать заранее в доступных 3D-редакторах. Размещая эти маркеры на плоскости, пользователь через разработанное приложение видит не маркеры, а изображение этих объектов. При этом, например, дорожная сеть достраивается автоматически между всеми объектами. Пользователь может в интерактивном режиме передвигать объекты, добавлять или убирать их со сцены. Программа должна автоматически генерировать связи между объектами (дорожную сеть). Оптимизации дорожной сети не требуется, достаточно просто нарисовать кратчайшие пути между объектами. При применении программы допустимо использовать готовые библиотеки для 3D-визуализации, например, Unity.

**Регламент испытаний:**

- Участники демонстрируют жюри работающее приложение.
- Участники демонстрируют жюри процесс распознавания отдельных маркеров.
- Участники демонстрируют жюри возможность интерактивного взаимодействия с моделью, перемещать маркеры, убирать их или добавлять на сцену. При этом программа должна самостоятельно, без ввода дополнительной информации генерировать изображения 3D-объектов в соответствии с расположением маркеров.
- Участники демонстрируют жюри дополнительные возможности программы.

**Примерный перечень средств и инструментов для выполнения задания:**

- <https://developer.android.com/studio>
- <https://unity.com/>
- <https://sparkar.facebook.com/>
- <https://www.viewar.com/>

## Сбор и обработка данных температуры

### Условие:

Реализовать программный модуль для сбора, хранения и обработки данных с удалённых температурных датчиков.

### Техническое задание:

Имеется специализированный сервис, расположенный по адресу [http://dt.miet.ru/ppo\\_it](http://dt.miet.ru/ppo_it).

Интерфейс прикладного программирования (API) позволяет получать данные от датчиков температуры воздуха на улице и в квартирах жилых домов, расположенных в различных районах в десяти городах.

Взаимодействие с сервисом осуществляется посредством HTTP-запросов.

Получение данных от сервиса осуществляется с помощью GET-запросов.

Необходимо на протяжении не менее суток реального времени собрать данные об уличной температуре во всех городах и температуре с не менее чем десяти датчиков, расположенных в квартирах в не менее чем пяти районах городов.

48 часов реального времени соответствуют календарному году, моделируемого сервисом.

Обращение к сервису должно происходить не реже, чем один раз в 10 минут реального времени, но не чаще, чем один раз в минуту.

Рекомендуется использовать СУБД для хранения собранных данных, также предоставить ER-модель.

По запросу в пользовательском интерфейсе программный модуль должен отображать в виде графика/диаграммы информацию о температуре за период сбора информации как на улице, так и в квартирах. Пользовательский интерфейс должен быть кроссплатформенным, удобным и понятным пользователю.

Проект рекомендуется вести с помощью системы контроля версий git.

**Московская предпрофессиональная олимпиада школьников  
Заключительный этап**

**Кейсовые задачи.**

**Инженерно-конструкторское направление. Профиль Информационные технологии**

---

**Регламент испытаний:**

- Участники демонстрируют получение данных температуры в реальном времени в определённой квартире по запросу жюри.
- Участники демонстрируют жюри график, отображающий изменение показаний уличной температуры на протяжении суток реального времени (полугодие моделируемого времени) в одном из городов.
- Участники демонстрируют жюри график, отображающий изменение средней температуры в квартирах в одном из городов на протяжении суток реального времени (полугодие моделируемого времени).
- Участники демонстрируют жюри график, отображающий изменение температуры, полученной в одной квартире в каждом из городов.
- Участники демонстрируют жюри диаграмму, отображающую максимальную температуру, полученную в квартирах в каждом из районов (не менее пяти) одного города.

**Примерный перечень средств и инструментов для выполнения задания:**

- <https://www.python.org/>
- <https://nodejs.org/>
- <https://sqlite.org/>
- <https://www.tarantool.io>
- <https://pypi.org/project/requests/>
- <https://reactjs.org/>
- <https://vuejs.org/>
- <https://matplotlib.org/>
- <https://poiskvps.ru/>

## Охранный комплекс

### Условия:

Разработать и реализовать охранный комплекс с возможностью детектирования в человека и хранением данных на удалённом хранилище.

### Техническое задание:

Комплекс во включённом состоянии должен обеспечивать контроль доступа в помещение, с помощью видеокамеры. В случае обнаружения движения человека в кадре, производится серия снимков (15 шт. 1 раз/сек.) до обнаружения и, до тех пор, пока есть движение в кадре, снимки должны отправляться на удалённое хранилище. Возможен вариант формирования видеофрагментов с частотой кадров не менее 1 кадр/сек.

Должно быть предусмотрено включение/отключение охранного режима.

Доступ к удалённому хранилищу рекомендуется осуществлять по логину/паролю, предусмотреть добавление пользователей с разграничением доступа: чтения и чтения/записи. Должен иметься интерфейс пользователя и администратора для взаимодействия с удалённым хранилищем и созданием/удалением пользователей.

Удалённое хранилище рекомендуется размещать в глобальном доступе.

Использование дополнительных датчиков, помимо видеокамеры, не предусмотрено.

Использование камер видеонаблюдения с предустановленным ПО, не предусмотрено.

Использование сторонних сервисов, позволяющих осуществлять распознавание образов не рекомендуется.

Использование готовых решений сервисов облачного хранения не предусмотрено.

**Московская предпрофессиональная олимпиада школьников**  
**Заключительный этап**  
**Кейсовые задачи.**

**Инженерно-конструкторское направление. Профиль Информационные технологии**

---

Проект рекомендуется вести с помощью системы контроля версий git.

**Регламент испытаний:**

- Комплекс должен быть включён, по умолчанию при включении охранный режим активен.
- Осуществляется движение объекта (не человека) в кадре, и проверяется отсутствие кадров/видеофрагментов на удалённом хранилище.
- Осуществляется движение человека в кадре и проверяется наличие кадров/видеофрагментов на удалённом хранилище без аутентификации.
- Доступ к хранилищу осуществляется по логину/паролю пользователя по умолчанию с правами чтения/записи.
- Создаётся новый пользователь с возможностью только чтения данных на удалённом хранилище.

**Примерный перечень средств и инструментов для выполнения задания:**

- <https://www.python.org/>
- <https://mariadb.org/>
- <https://www.raspberrypi.org/>
- <https://opencv.org/>
- <https://owncloud.org/>
- <https://www.raspberrypi.org/documentation/usage/camera/>

## Электронная цифровая подпись

### Условие:

В компании внедрён электронный документооборот, значительно ускоряющий согласование документов на всех уровнях.

Каждому сотруднику компании создана электронная подпись в виде flash-накопителя с записанным на неё определённым индивидуальным 5-значным шифром. Метод шифрования выбирается на усмотрение команды.

Подтверждение (визирование) документов происходит посредством проверки совпадения шифра с flash-накопителя с шифром, взятым из базы данных сотрудников.

Задача кейса – разработать аналог электронной подписи сотрудника компании (файл на flash-накопителе) для проведения согласования документооборота и разработать базу данных сотрудников организации.

### Техническое задание:

Необходимо разработать базу данных сотрудников компании, содержащую такие характеристики, как:

- ФИО сотрудника;
- дату рождения сотрудника;
- отдел, в котором сотрудник числится;
- должность сотрудника;
- образец электронной подписи;
- уровень подписи сотрудника (генеральный директор, зам, начальник отдела, рядовой сотрудник) для соблюдения иерархии подписания документов.

База данных должна содержать 3 таблицы:

- сотрудники;
- отдел;

**Московская предпрофессиональная олимпиада школьников**

**Заключительный этап**

**Кейсовые задачи.**

**Инженерно-конструкторское направление. Профиль Информационные технологии**

---

– реестр подписанных/согласованных договоров.

Разработанная система проверки подлинности должна осуществлять сравнение данных с flash-накопителя и базой данных в момент согласования документа.

Система проверки имеет интерфейс с полями выбора из базы данных (выпадающий список, автоматический поиск по фамилии, мастер-копи подписи для проверки) подписантов документа, кнопки функционала – «Проверка», «Отмена», «Выход».

**Регламент испытаний:**

Подписанты по очереди вставляют flash-накопители для подтверждения электронной подписи, и в случае совпадения документ считается подписанным (сообщение системы «Подписи сторон подтверждены. Договор подписан»). В случае несовпадения электронной подписи система выдаёт окно ошибки с указанием, чья электронная подпись не подтверждена.

В ходе тестирования используются flash-накопители с записанным шифром: корректный и некорректный.

## **Мобильное приложение с использованием краудсорсинга**

### **Условия:**

Краудсорсинг – это совместное решение задач с привлечением большого количества людей. На данный момент краудсорсинг применяется почти во всех сферах, начиная от бизнеса, государственных и муниципальных задач и заканчивая искусством. Всеобщее распространение мобильных устройств с доступом в сеть Интернет позволило внедрить идею краудсорсинга в множество важных процессов.

Например, общественный контроль состояния дорог, незаконных свалок мусора, нарушений правил дорожного движения (парковка в неподобающем месте), составление карты достопримечательностей родного города и т. д.

Участникам олимпиады предлагается создать мобильное приложение для решения какой-либо социально значимой проблемы с привлечением механизма краудсорсинга. Конкретная область применения приложения выбирается участником самостоятельно.

### **Техническое задание:**

Приложение должно быть реализовано для платформы Android на языках программирования Java или Kotlin.

#### *Описание функционала приложения*

1. Мобильное приложение должно иметь функционал для регистрации и аутентификации пользователей.
2. Каждый пользователь приложения должен иметь возможность вносить свой вклад в краудсорсинговый проект, который далее мы будем называть Сообщением.

Пользователь должен иметь возможность создавать новое сообщение, просматривать список своих и чужих сообщений, просматривать подробную информацию о конкретном сообщении, время и дату его создания.

3. Формат Сообщения:

- Название Сообщения.
- Содержание Сообщения (описание вклада в краудсорсинговый проект).
- Описательные поля Сообщения, по которым будет полезно проводить фильтрацию списка Сообщений для пользователей. Поля разрабатываются участником самостоятельно, с учётом специфики сферы, в которой будет применяться приложение. Обязательные описательные поля:

- геолокация места, которое относится к Сообщению. В качестве геолокации может использоваться текущее местоположение или выбрано место на карте;

- пользователь должен иметь возможность прикрепить одно или несколько изображений к Сообщению. Изображение может быть сделано с помощью фотокамеры мобильного устройства или добавлено из файловой системы мобильного устройства.

4. Список сообщений, созданных пользователем, должен иметь возможность:

- сортировки и фильтрации списка по статусу сообщения, полям и дате создания сообщения;

- Поиска Сообщений по названию Сообщения.

5. Сообщение должно иметь статус. Статус Сообщения может быть одним из: «отправлено», «на рассмотрении», «отклонено», «выполняется», «закрито». Вы можете добавить свои собственные статусы Сообщений в зависимости от специфики сферы, в которой будет использоваться приложение. Статус Сообщения должен быть виден в списке Сообщений,

Московская предпрофессиональная олимпиада школьников

Заключительный этап

Кейсовые задачи.

**Инженерно-конструкторское направление. Профиль Информационные технологии**

---

созданных пользователем, а также указываться при просмотре подробной информации о Сообщении.

При изменении статуса сообщения в системе краудсорсинга, ответственное лицо, которое изменило статус сообщения, должно оставить комментарий, который должен быть виден пользователю при просмотре подробной информации о сообщении.

При изменении статуса Сообщения в мобильном устройстве должно появляться уведомление об изменении статуса. На уведомлении должна открываться активность с информацией о сообщении.

б. Серверная часть приложения, а также приложение, в котором работают люди, ответственные за обработку Сообщений (изменение статуса, комментарии к сообщению), могут быть реализованы на любом языке программирования и с применением любых доступных технологий. Хранение данных должно быть организовано с использованием базы данных. Серверная часть должна работать в режиме 24 x 7, чтобы обеспечить работоспособность мобильного приложения в период проведения олимпиады. Серверная часть приложения **не является** объектом оценивания.

*В качестве ответа сдаются:*

1. Ссылка на исходный код и исполняемый файл (apk) приложения в системе контроля версий.
2. Ссылка на видео, демонстрирующее работу приложения продолжительностью до 2 минут.
3. Презентация с следующими минимальными требованиями:
  - Заголовок: наименование проекта, ФИО участников.
  - Введение: идея приложения, для решения каких задач оно создано.
  - Описание реализации: структура и особенности приложения, схема классов.

**Московская предпрофессиональная олимпиада школьников**  
**Заключительный этап**  
**Кейсовые задачи.**

**Инженерно-конструкторское направление. Профиль Информационные технологии**

---

- **Заключение:** выводы по работе, возможности для доработки и развития.

Видео и презентация о приложении призваны доступно и кратко донести до членов жюри преимущества приложения.

**Регламент испытаний:**

1. Установка из apk файла минимум на два мобильных устройства: смартфон и планшет.
2. Регистрация и аутентификация нескольких пользователей.
3. Создание нескольких сообщений в приложении.
4. Просмотр, фильтрация и сортировка списка Сообщений.
5. Поиск и просмотр подробной информации о Сообщении.
6. Ожидается изменение статуса Сообщения в системе. Статус Сообщения изменяется авторами приложения. Проверяется уведомление об изменении статуса Сообщения и переход к подробной информации о Сообщении при нажатии на уведомление.

Пункты 2 – 6 выполняются для различных пользователей на различных устройствах.

В процессе испытания может происходить параллельное взаимодействие пользователя с другими приложениями, изменение ориентации устройства и т. д.

**Примерный перечень средств и инструментов для выполнения задания:**

- Открытый учебник «IT ШКОЛА SAMSUNG» <https://opencourse.myitschool.ru/>
- Среда разработки Android Studio <https://developer.android.com/studio>
- Документация для разработчиков Android-приложений <https://developer.android.com/>

**Московская предпрофессиональная олимпиада школьников**

**Заключительный этап**

**Кейсовые задачи.**

**Инженерно-конструкторское направление. Профиль Информационные технологии**

---

- Документация по языку Kotlin <https://kotlinlang.org/docs/reference/>
- Документация по языку Java <https://docs.oracle.com/>

-